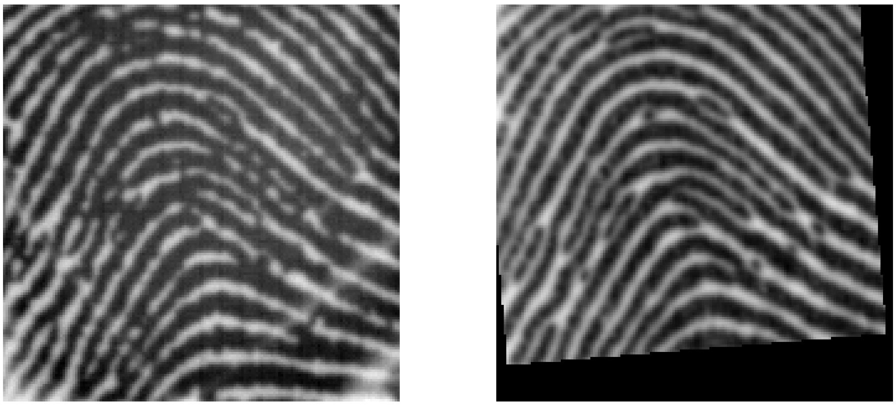
CHALMERS
UNIVERSITY OF TECHNOLOGY



Feature-based Quality Assessment of Spoof Fingerprint Images

Master's thesis in Biomedical Engineering

JENNY NILSSON

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis 2017:01

Feature-based Quality Assessment of Spoof Fingerprint Images

JENNY NILSSON

Department of Signals and Systems
Signal Processing and Biomedical Engineering
Computer Vision and Medical Image Analysis
Chalmers University of Technology
Gothenburg, Sweden 2017

Feature-based Quality Assessment of Spoof Fingerprint Images
JENNY NILSSON

Supervisor: Jennifer Alvén, Department of Signals and Systems
Supervisor: Olof Enqvist, Department of Signals and Systems
Supervisor: Kenneth Jonsson, Fingerprint Cards AB
Examiner: Fredrik Kahl, Department of Signals and Systems

Master's Thesis 2017:01
Department of Signals and Systems
Signal Processing and Biomedical Engineering
Computer Vision and Medical Image Analysis
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Example fingerprint images. The left one is a fingerprint image created by placing a synthetic fingerprint on the fingerprint sensor and the right one is created by placing the live finger on the fingerprint sensor. The two fingerprint images originate from the same fingerprint area.

Gothenburg, Sweden 2017

Abstract

Fingerprint recognition has over the last decade become a natural component in modern identity management systems. As the commercial use of fingerprint recognition systems increases, the benefits from attacking such systems become greater. The security of a biometric system is seriously compromised if the system is unable to differentiate between a real and a counterfeit fingerprint. From this security threat, a need for methods to prevent or detect such spoofing attacks has emerged.

This thesis is concerned with so called liveness detection, that is the process of determining whether a captured fingerprint is fake or not. More precisely, the thesis explores different ways to assess how difficult it is to correctly classify a set of fake fingerprint images. Differences in image characteristics between the two classes are also explored. The purpose of the thesis is to design a quality assessment tool for fake fingerprint images used in the liveness algorithm development at Fingerprint Cards. The quality assessment tool aims to give an indication of how difficult a set of such 'spoof' images are to classify based on the evaluated liveness characteristics.

In the first part of the thesis, features which differ between images of genuine and fake fingerprints are designed. Based on these designed liveness features, a support vector machine classifier is created by identifying the hyperplane model which best separates the images of living and spoof fingerprints. The quality of a spoof image data set is defined as the number of spoof images that this hyperplane model manages to classify correctly. Further, the quality of each individual spoof image is defined as the liveness probability assigned by the hyperplane model.

Promising results were obtained from the quality assessment tool developed in the first part of the thesis. The spoof images that were assigned a low quality by the hyperplane model were images which easily could be differentiated from their live equivalents in a manual inspection. Conversely, the spoof images that were assigned a high quality were images in which the fingerprint patterns could not be differentiated from live fingerprint patterns. Hence, these results indicate a successfully designed spoof quality assessment. Further, it shows that manually designed liveness features can be used to estimate the spoof image quality.

In the second part of the thesis, a deep fine-tuned convolutional neural network is evaluated for quality assessment of spoof images. The utilized network has recently obtained state-of-the-art results in fingerprint liveness detection. If the deep neural network cannot differentiate between the live and fake images in a set, the images are considered very hard to classify. Conversely, if a shallow network easily differentiates spoof images from live images, these images are considered easy to classify.

The liveness classification results obtained in the second part of the thesis were far better than expected. The fine-tuned convolutional neural network demonstrated fantastic liveness classification results by classifying all images in the test set correctly. These results imply that all the images in the set are possible to classify properly. However, the fact that the network managed to classify even the most realistic spoof images correctly with a high degree of certainty makes this network architecture unsuitable for spoof quality assessment. Differentiation between the images in the set could however possibly be obtained with a more shallow network.

Keywords: secure identity management, biometric recognition, fingerprints, spoofing attacks, liveness detection, spoof fingerprint image quality, liveness feature extraction, convolutional neural networks.

Acknowledgements

I would first like to acknowledge Kenneth Jonsson for giving me the opportunity to work with this master thesis project. Further, I would like to thank the entire Algorithm group at Fingerprint Cards, Gothenburg, for introducing me to the fields of fingerprint recognition and liveness detection and for all the help and valuable discussions along the way. I would also like to thank all other staff at Fingerprint Cards, Gothenburg, for the warm welcome and for showing interest in my thesis work. Moreover, I would like to sincerely thank my academic supervisors, Jennifer Alvén and Olof Enqvist, for their expertise, guidance and support throughout the project. It has been a pleasure working with all of you. Lastly, I would like to thank my friends and family, especially Filip, for your unconditional support during my studies at Chalmers. Without you I would not have come this far.

Jenny Nilsson, Gothenburg, January 2017

Contents

1
Introduction

Identity management is an essential element in a large number of societal applications today. As concerns about the security in many of these applications increase, so does the need for systems which reliably assess the identity of its users. Examples of applications in which the demand for secure identity establishment is particularly high are regulation of international border crossings, access control to important facilities or privileged information as well as electronic access control and transactions. Since traditional knowledge and token-based credentials such as passwords, keys, cards and passports can be shared, guessed, misplaced or stolen, they cannot be fully trusted to establish identities. Thus, these traditional identity management systems often fail to meet high demands on performance and security [1].

Biometric recognition has over the last decades been increasingly deployed as an alternative or supplement to traditional systems and has now become a natural component in authorization and identification systems. Since biometric recognition systems are based on the premise that all individuals possess distinctive anatomical and behavioral characteristics from which they uniquely can be associated with an identity, these systems are generally agreed to be a reliable and powerful tool in modern identity management. The biometric characteristics most commonly used today are fingerprints, faces and irises and since biometric identifiers intrinsically represent the bodily identity of an individual they can neither be misplaced, shared nor stolen. Consequently, person recognition systems utilizing identifiers that intrinsically are linked to the user are considered to be superior to traditional knowledge or token-based methods, both in terms of security and user convenience [1, 2].

Even though biometric recognition systems are based on identifiers unique for all users, this fact does not make them immune to fraudulent attacks. In general, there are two ways to circumvent a biometric system, either by direct or indirect attacks. Indirect attacks refer to the type of attack performed inside the system, such as manipulation of the feature extraction or feature matching modules and modification of the database containing enrolled feature sets. Since indirect attacks are performed within the digital limits of the system they can be prevented by digital protection mechanisms like anti-virus software, encryption, firewalls and intrusion detection. The type of attacks performed outside the digital limits of the system is referred to as direct attacks. In such attacks the impostor either modifies its biometric identifier to evade identification or poses as a valid user by presenting a forged biometric identifier to the sensor in order to illegitimately be granted access to the system. The latter of these, the presentation of a counterfeit biometric identifier to the sensor, is commonly known as spoofing or presentation attacks. The counterfeit biometric identifiers used in these attacks are commonly known as spoofs. As spoofing neither require any advanced programming skills nor any remarkable alterations of the intruders own biometric identifiers, it is considered to be the most potent and damaging type of system attack [3].

As the commercial use of biometric systems increases, the benefits from attacking such systems become greater, hence the frequency and intensity of spoofing attacks are expected to increase in the coming years. The security of a biometric system is of course seriously compromised if the system is unable to differentiate between a real and a counterfeit biometric identifier. From this security threat a need for methods to prevent or detect spoofing attacks has emerged. Our facial images and voices are constantly captured by cameras and audio recorders and our fingerprints and DNA are left wherever we touch, hence our biometric identifiers can in no way be claimed as secret. Consequently, the security of biometric systems cannot rely on the inaccessibility of our biometric identifiers to potential attackers, but must instead take the liveness of the presented sample into account. Thus, most biometric recognition systems now couple their identification or verification process with a spoofing countermeasure module which evaluates the liveness of the presented sample and classifies it either as a real, living sample or as a non-live sample [1, 3].

In fingerprint recognition systems, there are both hardware-based and software-based techniques to evaluate the liveness of a sample. In the hardware-based approaches, additional devices are added to the system to detect different properties of the sample that is associated with either live or non-live traits. There are a lot of ways to measure vital signs with hardware-based techniques, a few of many examples are measurement of blood flow, oxygenation of the blood, electrical properties or signals, skin perspiration, spectral characteristics or biochemical assays of human tissues as well as changes in skin tone when the finger is pressed against the sensor surface. It is also possible to detect odors and properties associated with different spoofing materials. While hardware-based approaches operate directly on the finger, software-based approaches operate on the fingerprint images already obtained from the sensor and make use of differences in features extracted from live samples and spoofs. Hardware-based techniques generally have a higher performance compared to software-based approaches and the best classification performance would probably be obtained by a combination of the two. The additional sensors required in hardware-based approaches, however, brings considerable and undesirable additional costs and size to the systems. Thus, software-based approaches are preferable in applications where low cost and small size are significant factors [3].

1.1 Problem description

Fingerprint Cards is a company developing both software and hardware solutions for biometric recognition systems based on fingerprint characteristics. One of these software solutions is an algorithm which evaluates the liveness of fingers presented to the sensor [4]. Algorithms for liveness detection are often based on feature extraction from the fingerprint image and use different machine learning techniques to classify the presented finger as either live or spoof based on the resulting feature vector. The performance of such a classification algorithm is heavily dependent on the amount and nature of training data as well as on the characteristics of the data to be classified [2, 5]. Since the algorithm performance is data dependent, it differs for different data sets due to variations in the data. In order to obtain comparable algorithm performance measurements for data sets from different sources, a quantitative quality assessment of the spoof images in the data sets is needed.

1.2 Contribution

This thesis explores different ways to assess how difficult it is to correctly classify a set of fake fingerprint images. The purpose of the thesis is to design a quality assessment tool for the fake fingerprint images used in algorithm training and evaluation at Fingerprint Cards. The quality assessment tool aims to give an indication of how difficult a set of spoof fingerprint images are to classify correctly.

1.3 Scope

In this thesis, two different ways to assess how difficult it is to correctly classify a set of fake fingerprint images are explored. The thesis is thus divided into two parts. In the first part of the thesis, features which differ between live and spoof fingerprint images are designed. These differences will be measured quantitatively and used to estimate the quality of spoof fingerprint images. This part has been limited to only considering images acquired from one type of sensor, and it has also been limited to only considering images of spoofs made of wood glue that are fabricated from a two-dimensional fingerprint capture. In the second part of the thesis, convolutional neural networks are considered as a possible way of evaluating the spoof fingerprint image quality. A liveness classifier based on a deep convolutional neural network will be implemented and the spoof image quality in a given data set will be estimated from the obtained network classification results.

1.4 Related work

The Fingerprint Liveness Detection Competition is a competition which is held every other year and which compares fingerprint liveness classification methodologies and establishes the current state-of-the-art in liveness classification. The data sets used in these competitions are publicly available and are used as benchmark data sets in the liveness classification research community [3]. State-of-the-art results on these data sets using the software-based liveness detection approach has been reported in [6–9]. The fingerprint liveness classification algorithm presented in [6] is a learning-based classifier which uses convolutional neural networks for fingerprint liveness detection. This algorithm won the last Fingerprint Liveness Detection Competition which took place in 2015. The algorithm which placed second in the last Fingerprint Liveness Detection Competition was submitted from the research group behind [7, 8]. In these articles, a liveness detection method based on local descriptors and support vector machine classification is proposed. The novel local descriptor proposed in [7] is based on image contrast and phase information extracted locally from the spatial and frequency domains of the image. One of the descriptors investigated in [8], is the same as the local descriptor proposed for fingerprint liveness classification in [9]. This descriptor encodes the local fingerprint texture to a feature vector by using automatically learned filters. The classification algorithm in [9] is a support vector machine classifier based on this feature vector. These four methodologies are however designed for optical fingerprint images, which differ much in characteristics from the capacitive fingerprint images assessed at Fingerprint Cards.

Various fingerprint-specific image quality features have also been proposed for fingerprint liveness classification purposes [10–15]. The fingerprint image quality may be assessed by measuring the pattern clarity, the pattern continuity, and the pattern strength or directionality [10]. The pattern strength has been assessed both by measuring the fingerprint orientation certainty level [11] and the energy concentration in the power spectrum [12]. The discriminative power of these liveness features has been found to be high [10]. The pattern clarity has been assessed by measuring the mean and standard deviation of the fingerprint image intensities [13], by estimating a local clarity score [14] and by analyzing the amplitude and variance of the sine waves which form the fingerprint pattern [15]. The pattern continuity has been assessed both by measuring the local fingerprint orientation quality [14] and the continuity of the fingerprint orientation field [11]. Both the pattern clarity and the pattern continuity have been found to have medium discriminative power [10] in the quality assessment for liveness classification.

2

Theory

2.1 Fingerprints

The pattern of the fingertip epidermis is referred to as a fingerprint. The most obvious structural characteristic of a fingerprint is the pattern of interleaved ridges and valleys, see Figure 2.1. In most fingerprint representations, the ridges are dark and the valleys are bright. The fingerprints of an individual have fully formed already at birth. Except for temporary alterations such as bruises and cuts, the fingerprint ridge configuration does not change throughout the life of an individual. In addition, it is generally believed that all fingerprints are unique, even though this is an empirical observation rather than a scientifically established fact. These properties make fingerprints very attractive as a biometric identifier and identification based on fingerprint recognition has been formally accepted since the early twentieth century [2].

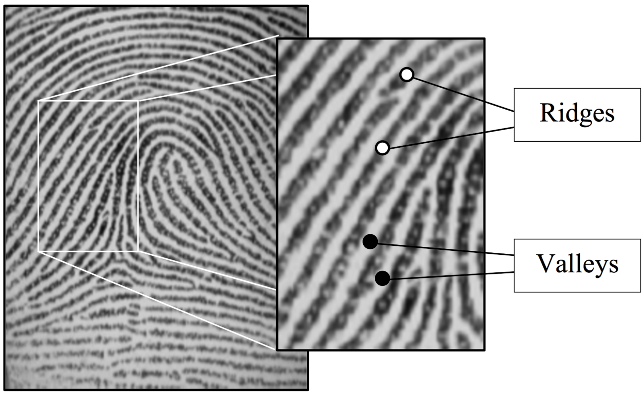


Figure 2.1: Example of a fingerprint image. In the magnified image, the fingerprint ridges and valleys are emphasized [2].

2.1.1 Fingerprint formation

Fingerprint patterns are fully formed already when a fetus is seven months old. The epidermal fingerprint pattern emerges when the skin on the fingertip starts to differentiate. The non-uniform growth of the basal cell layer in the epidermis causes a buckling instability in the basal layer which results in the creation of epidermal ridges. The creation of epidermal ridges is also affected by environmental changes such as changes in the position of the fetus or the flow of amniotic fluids surrounding it. These changes determine the finer details of the fingerprints and even minor differences in the micro-environment affect the fingerprint formation since the changes are amplified by the differentiation process of the cells. Also, since the micro-environment slightly differs from finger to finger, the cells will grow differently on all fingers and thus the fingerprints of an individual will become unique even though the genetic material influencing the differentiation process is identical. The huge variety in environmental changes and genetic information makes it virtually impossible for two fingers to get the exact same fingerprint pattern [2, 16].

2.1.2 Fingerprint representation

Fingerprint patterns are generally described at three different feature levels, the global level, the local level and the fine level. At the global level, fingerprints are represented by its ridge flow and its ridge frequency. The exact dimensions and locations of the ridges are however ignored. The global fingerprint pattern is mostly composed of smooth and parallel lines, but the pattern also contains regions in which the ridge orientation abruptly changes and the pattern assumes a distinctive shape. Such regions are termed singularities or singular points and they are broadly classified into deltas, loops and whorls, see Figure 2.2 [1, 2].

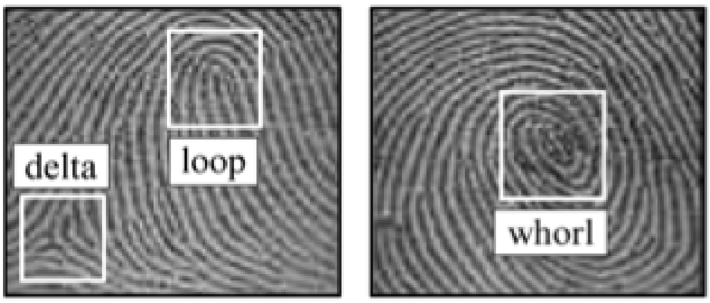


Figure 2.2: Fingerprint singularities. In the left image a delta and a loop are emphasized and in the right image a whorl singularity is emphasized [17].

At the local level, fingerprints are represented with their ridge skeletons. A ridge skeleton is created by converting the fingerprint ridges to one-pixel-wide lines using an iterative ridge width reduction method. By representing fingerprints with their skeletons, the ridge location information is conserved. The geometrical and dimensional details of the ridges are however still ignored. Another important feature at the local level are the locations in which the skeleton or the ridges are discontinuous. These discontinuities are termed as minutiae points. The two minutiae types used to describe the local ridge pattern are ridge endings and ridge bifurcations, i.e. when a ridge is abruptly ended or is divided into two ridges. Other examples of minutiae are lakes, dots, spikes and crossovers. In Figure 2.3, the seven most common minutiae types are displayed. Minutiae-based fingerprint representations are used extensively in automatic fingerprint matching since most of the discriminative information or individuality in a fingerprint is captured by its set of minutiae points. Algorithms extracting minutiae are though heavily dependent on the fingerprint image quality. In an ideal fingerprint image, the ridge flow is locally constant and there is a distinct alternation between ridges and valleys. Minutiae points are quite easily extracted from ideal fingerprints, but if the image quality for some reason is degraded the extraction of minutiae becomes more problematic. Degraded fingerprint images may, for example, contain poorly separated ridges, ridge discontinuities or ridge breaks and fingerprint imperfections like cuts, creases, and bruises. Thus, when extracting minutiae points from a degraded image, spurious minutiae are often created while genuine minutiae might be ignored. Factors affecting fingerprint image quality negatively are incorrect finger pressure, inherently low-quality fingerprints, sensor noise and skin conditions, like too dry or too wet fingers [1, 2, 15].

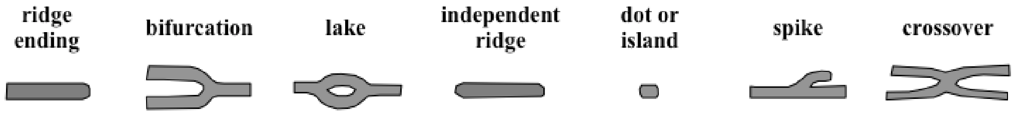


Figure 2.3: The seven most common fingerprint minutiae types. The first two, ridge endings and bifurcations, are used to represent local fingerprint patterns [2].

At the even finer level, detailed information embedded in the ridges are observed. Such information includes the contours, shape and width of the ridges as well as ridge breaks, creases and sweat pores. Also included at the fine level are incipient ridges and ridge dots. Incipient ridges are thinner ridges which due to their immaturity do not contain any sweat pores and a dot is a very short ridge. However, a reliable capture of these fine level features requires both high sensor resolution and good image quality. To date, most fingerprint sensors are equipped with 500 dpi resolution while a reliable capture of most fine level features requires a sensor resolution of 1000 dpi [1, 2]. In Figure 2.4, the features at the three different levels are visualized.

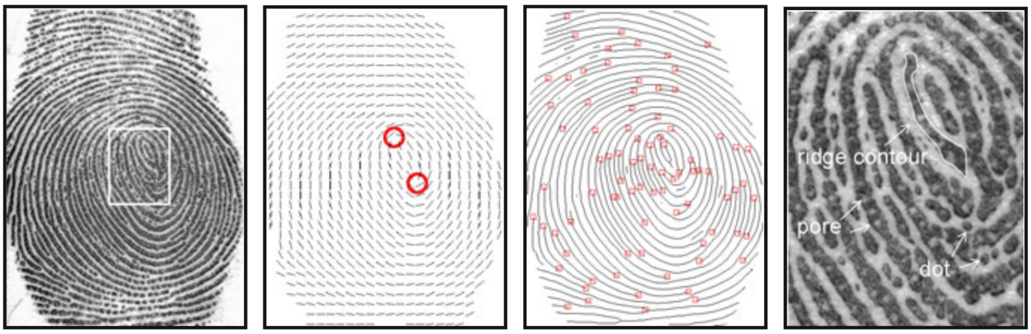


Figure 2.4: Fingerprint feature levels. The first image shows the original fingerprint and the consecutive images visualizes the different feature levels. The second image shows the global level representation as the ridge flow and singularities, the third image shows the local level representation as the ridge skeleton and minutiae points and the fourth image shows some of the fine level fingerprint features [1].

2.1.3 Fingerprint sensing

In automated fingerprint identification systems, fingerprints are acquired by sensors capable of digitizing the prints on contact. There are different types of fingerprint sensing techniques, some examples are the optical, capacitive, thermal and ultrasound sensing techniques. Optical and capacitive sensors are the most commonly used and while optical sensors have the longest history and highest resolution, capacitive sensors are cheaper, more compact in size and easier to embed in consumer products. In addition to the sensor technology used, fingerprint images are also characterized by the resolution, sensor area, contrast and by geometric effects [2].

The large variability in impressions of a single finger makes reliable fingerprint recognition a difficult task. The fingerprint area acquired by the sensor is, of course, dependent on the location and rotation of the finger with respect to the sensor. Even the smallest finger displacement results in a noticeable translation of the acquired fingerprint area. For example, a displacement of 2 mm would be imperceptible to the user but results in a translation of around 40 pixels if the resolution of the fingerprint sensor is 500 dpi. Due to the plasticity of the skin, the finger is also deformed when placed on a surface. The deformation of the finger is dependent on the way it is applied to the sensor. If the contact force contains components non-orthogonal to the sensor surface, non-linear distortions of the finger such as compression and stretching are produced. Hence, impression variations most often arise from the way the finger is presented to the sensor. Other factors responsible for the large variations in fingerprint impressions of a single finger are changes in skin condition and variable finger pressure against the sensor. A fingerprint is accurately captured by the sensor if there is a uniform contact between the finger and the sensor. A uniform contact is however highly unlikely since it would require a uniform finger pressure in combination with a perfect skin condition. The non-uniform contact introduces noise to the acquired fingerprint images and since the contact between the finger and the sensor varies in successive acquisitions of the same finger, so do the amount of noise in the fingerprint images. Noise may also be introduced if there are dust or residues from the previous fingerprint capture on the sensor surface [2].

2.1.4 Fingerprint spoofing

Due to recent advances in fingerprint spoofing, the security of a fingerprint recognition system relies on an accurate liveness evaluation of the finger presented to the fingerprint sensor. Thus, in modern fingerprint recognition systems the identification or verification processes is coupled with a spoofing countermeasure module which evaluates the liveness of the presented finger and classifies the obtained image either as a live or spoof fingerprint in order to prevent system circumvention. However, if a spoof is similar enough to its live equivalent, the system will interpret the spoof as a genuine fingerprint, hence accepting the intruder as a valid user [1].

Fingerprint spoofs are thin pieces of gelatin, latex, silicone, wood glue or similar material carrying a fingerprint. There are two general methods to create spoofs. In the first method, a negative impression of the finger is created by placing the finger into dental silicone or a similar plasticine-like material. The spoof is then created by covering the three-dimensional negative impression with a thin layer of some spoof material. This method is referred to as the cooperative method since it typically requires a cooperative subject. In the second method, which is referred to as the non-cooperative method, the fingerprint is lifted from a surface such as a mobile phone display or a glass. The lifted, two-dimensional fingerprint is then used to create a spoof either by printing the fingerprint image with conductive silver ink or by etching the fingerprint image onto a printed circuit board and cover it with a thin layer of some spoof material [5]. The characteristics of a spoof fingerprint image is dependent on the quality of the spoof, the spoof material and also on the fingerprint sensor since the material properties of the different spoof materials are not equally compatible with the different sensor technologies.

2.2 Fingerprint image pre-processing

In the following sections, some background and theory related to the feature design part of the thesis are presented. Subjects included are fingerprint image segmentation, fingerprint pattern orientation, fingerprint image binarization, morphological operations used to analyze the fingerprint patterns as well as minutiae extraction.

2.2.1 Segmentation

If the finger is not in contact with the entire sensor area when the fingerprint image is captured, the resulting image will also contain some background. For sensors which create images in which the ridges are dark and the valleys are bright, the image background will also be bright since no signal is generated in the non-contact area. The process in which a fingerprint image is separated into foreground and background is referred to as fingerprint segmentation. The fingerprint area of the image, which is characterized by the striped and oriented ridge and valley pattern, is denoted foreground and the bright area caused by deficient finger presentation is denoted background. By separating the foreground from the background, the analysis of the fingerprint image can be constrained to the relevant area of the fingerprint image. A segmentation algorithm can be used to assign each pixel in an image to either the foreground or the background. The output of such an algorithm is a binary map of the same size as the image in which the pixels belonging to the foreground are ones and the pixels belonging to the background are zeros [2].

The segmentation algorithm implemented in this thesis is based on three pixel features; local gradient coherence, local intensity mean and local intensity variance. Due to lack of signal in the background, the local mean intensity is generally higher in the background than in the foreground. The intensity variance is high in the foreground due to the ridge and valley pattern, whereas only a small variance due to noise is seen in the background. There can however be some darker clusters in the background due to dust or grease on the fingerprint sensor, hence an additional feature is needed to make the segmentation algorithm robust to noise. Gradient coherence is a pixel feature which can discriminate the oriented pattern in the foreground from the isotropic pattern in the background. It measures to which degree the squared gradient vectors in a neighborhood share orientation. The gradient coherence is one if all the squared gradient vectors in a neighborhood are parallel and the gradient coherence is zero if the squared gradient vectors are equally distributed over all directions. Since a fingerprint consists of parallel line structures, the squared gradients in a neighborhood is more likely to point in the same direction in the foreground. The squared gradients in the background should to a larger extent be equally distributed due to noise and lack of parallel line structures [2, 18].

2.2.2 Fingerprint orientation

The tangential direction of the ridge or valley lines passing through a pixel is referred to as the ridge or fingerprint orientation at the given pixel. The orientation of a fingerprint image describes the coarse structure of the fingerprint pattern. The orientation map, which also is referred to as the directional field, of a fingerprint image is a matrix of the same size as the fingerprint image in which each matrix element contains the orientation of the corresponding pixel in the fingerprint image. In principle, the directional field in a fingerprint image is perpendicular to the angle of the local gradients in the image. Thus, the directional field is often estimated by averaging the gradients in a local neighborhood. However, since the angles of the gradients in an image range between 0 and 2π and the orientation only range between 0 and π, a traditional averaging of the gradients in a neighborhood is not applicable. This is because gradient vectors pointing in opposite directions cancel each other out in a conventional averaging operation, although these opposite pointing gradient vectors imply the same fingerprint orientation. This problem is often solved by doubling the angles of the gradient vectors before performing the averaging. The cyclic properties of the angles result in orientation angles in the range 0 and π when the doubled angles later are converted back. The length of the gradient vector is often also considered when estimating the orientation in an image by allowing the angles of the steeper gradients in a neighborhood to affect the orientation estimation more than the angles of the less steep gradients [1, 19].

2.2.3 Binarization

Binarization is the process in which an intensity image is converted to a binary image by grouping all the pixel values in an image into two modes. The two modes are separated by assigning the pixels above a certain intensity threshold to the image foreground and by assigning the pixels below the threshold to the image background. The intensity threshold is either determined on a global or a local scale and while a global threshold is constant and applicable to the entire image, a local threshold may differ in different parts of the image. A global threshold is often determined using an approach referred to as Otsu's method. Otsu's method calculates the optimum global threshold of an image by maximizing the inter-class variance and minimizing the intra-class variance with respect to the intensity values in the two modes. A local threshold is instead based on properties of a neighborhood, for example, the average pixel intensity in the neighborhood [20].

The success of an intensity-based thresholding is directly related to the intensity values in the image or in the local neighborhood. If the intensity histogram of the area of interest contains two well separated peaks, the chance of a successful separation of the modes in this area is high. However, if the histogram peaks are poorly separated or if the peaks are too wide, the selection of a threshold which would result in a successful separation of the modes becomes more problematic [20].

2.2.4 Morphological operations

Mathematical morphology is a technique based on set theory, lattice theory and integral geometry which is used in the analysis and processing of spatial structures. By applying morphological operations to an image, structures in the image may be extracted or modified. The most fundamental morphological operators are dilation and erosion. Other important concepts in morphology are opening, closing and the extraction of connected components and image skeletons. When processing an image with a morphological operator, the image is probed with a structuring element which is a small set of pixels. The structuring element is shifted over the entire image and in each pixel, the properties of interest in the morphological operation is evaluated. Properties of interest might for example be the union or the intersection of the structuring element and the underlying image pixels [20, 21].

In a morphological dilation, the foreground in the resulting image is an expanded version of the foreground in the original image since the dilation process assigns a pixel to the foreground if there is any overlap between the foreground of the original image and the foreground of the structuring element when it is centered around said pixel. A morphological erosion, on the other hand, shrinks the foreground in the original image since the erosion process only assigns a pixel to the foreground if the foreground of the original image and the foreground of the structuring element overlap completely [20]. In Figure 2.5, an image is visualized together with the results from a morphological erosion and a morphological dilation of the same image. A diamond-shaped structuring element was used in this example.

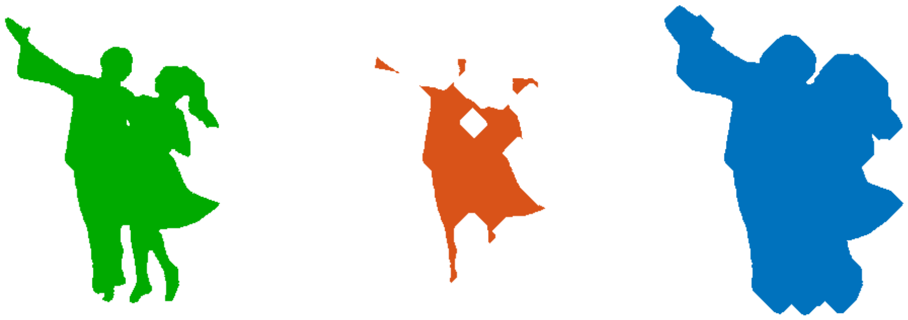


Figure 2.5: Morphological erosion and dilation. The middle image visualizes an erosion of the left image and the right image visualizes a dilation of the left image.

The morphological opening and closing operations both smooths the contours of the objects in an image. An opening operation is an erosion followed by a dilation and a closing operation is a dilation followed by an erosion. Morphological opening makes the contours in the image smoother by breaking narrow isthmuses and by removing thin protrusions, while closing makes the contours smoother by fusing narrow breaks. An opening operation also removes small foreground objects in the image background and a closing operation fills small holes in the image foreground [20].

Another morphological technique used in image analysis applications is the extraction of connected components in an image. Two adjacent foreground pixels are considered connected if the foreground of the structuring element, when centered at the first pixel, overlaps with the second pixel. The foreground of the structuring element is thus used to define the neighborhood in which the connectivity for a pixel is evaluated. For a two-dimensional image, the structuring element is either a 4-connected neighborhood or an 8-connected neighborhood, see Figure 2.6. If the structuring element is defined by a 4-connected neighborhood, the four pixels connected vertically and horizontally to the center pixel are considered as possible connected pixels. If the structuring element is defined by an 8-connected neighborhood, all the pixels surrounding the center pixel are considered as possible connected pixels. Non-adjacent pixels in an image are considered to be connected if there exists an N-connected path between them which entirely consists of connected foreground pixels. The pixels connected by such a path compose a connected set [20].

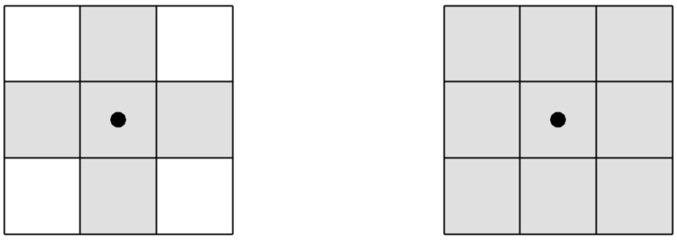


Figure 2.6: Structuring elements. The left image displays a structuring element based on 4-connectivity and the right image displays a structuring element based on 8-connectivity. The foreground pixels of the structuring elements are colored in gray and the background pixels of the structuring elements are colored in white. The black dots indicate the center pixels of the structuring elements.

An important technique for the representation of structural shapes in an image is to reduce its foreground objects to thin structures called skeletons. This skeletonization procedure iteratively removes the boundaries of the foreground objects in the image by morphological erosion until convergence. The erosions are however not allowed to remove end points or to break 8-connected components [20]. In Figure 2.7, an example of a skeleton is visualized together with the image of the original object.

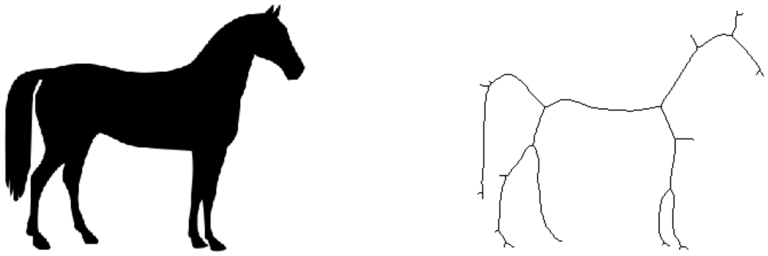


Figure 2.7: Morphological skeleton. The right image visualizes the morphological skeleton of the object in the left image. Note that the skeleton contains short spurs which are produced due to small irregularities in the boundary of the object.

2.2.5 Minutiae extraction

Once a fingerprint skeleton is obtained, the coordinates of the minutiae points in an image are easily extracted by calculating the crossing number for each skeleton pixel. The crossing number of a skeleton pixel is defined as half the sum of the pixel value differences between each pair of adjacent pixels in the skeleton pixel's 8-neighborhood. The definition of the crossing number is presented in Equation 2.1 and the indexing of the pixels in the neighborhood is clarified in Figure 2.8 [2].

$$\text{crossing number} \; = \; 0.5 \sum_{i=0}^{7} \mid P_{i+1 \bmod 8} - P_i \mid \tag{2.1}$$

P_0	P_7	P_6
P_1		P_5
P_2	P_3	P_4

Figure 2.8: Crossing number neighborhood. This is a schematic view of the neighborhood indexing around a skeleton pixel when its crossing number is calculated.

The crossing number is zero for isolated dots in the skeleton, one for skeleton end points, two for intermediate skeleton points, three for skeleton bifurcations and four for skeleton crossovers [2]. In Figure 2.9, examples of skeleton neighborhoods for three of these minutiae types are visualized. The crossing number of the center pixels in these examples are calculated from the pixels inside the marked areas.

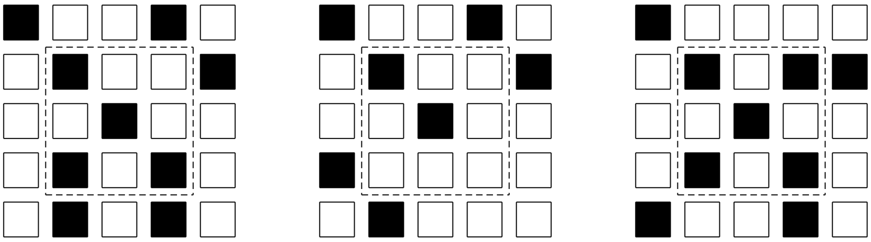


Figure 2.9: Examples of minutiae points neighborhoods. The center pixel in the left neighborhood is a skeleton bifurcation, the center pixel in the middle neighborhood is a skeleton end point and the center pixel in the right neighborhood is a skeleton crossover. The skeleton foreground pixels are the pixels in black.

2.3 Learning-based feature extraction

This section presents some background to learning-based tools for feature extraction, that is, neural networks and more specifically convolutional neural networks.

In recent years, deep learning and neural networks has come to provide powerful solutions to complex computer vision tasks such as recognition and classification. Deep learning refers to a machine learning technique in which deep network structures automatically learn appropriate internal representations, i.e. features, of the observed data and neural networks are such learnable network structures which are inspired by the human brain and its neural pathways. Convolutional neural networks are a specialized form of neural networks in which at least one of the network layers utilizes convolutional operations (i.e. filtering) [22].

A typical convolutional neural network consists of three types of layers; convolutional layers, pooling layers and fully connected layers. In a convolutional layer, convolutional operations are performed on the input to produce a set of linear activations, or feature maps, which are processed by non-linear activation functions such as the rectified linear unit function. Since each convolutional layer consists of several filters which extract different features, several feature maps are produced in each convolutional layer. While the first convolutional layers in the network typically identify edges, corners and extended contours, the higher level convolutional layers in the network structure encode more abstract features. Pooling layers in the network modifies the output from the previous convolutional layer by aggregating the activation information of neighboring pixels. The pooling operations often reduce the size of the feature maps by replacing each of the sets of neighboring pixels in the input feature map with a single pixel value representing the neighborhood statistics. The max pooling operation, which passes the maximum activation value in each neighborhood, is the most commonly used pooling operation. The pooling operations in the network structure produce feature maps which are less sensitive to local translations in the input image, thus enabling extraction of features which are more invariant of local translations in the original image higher up in the hierarchical model. The last few layers in a convolutional network are fully connected layers, which means that all its neurons are directly connected to all the activations in the previous layer. Such layers are computationally expensive but are needed at the end of the network to combine the final feature maps and to reach a final classification decision. The last fully connected layer is often followed by a softmax function in order to convert the network output to posterior probabilities [22].

Convolutional neural networks are often trained by combining an optimization algorithm such as stochastic gradient descent with the back-propagation algorithm. A set of labelled images is needed to train the network and the general practice is to divide this set into a training set, a validation set and a test set. The training and validation sets are both used when training the network. The training set is used to calculate the gradients needed for updating the network parameters in the back-propagation, while the validation set is only used to monitor the training process. The training progress is evaluated after an epoch is completed, i.e. when the entire set of training images have been processed. During each epoch of training, the images in the training set are further divided into smaller image sets called batches. The images in each batch are randomly fed through the network and by comparing the actual network outputs with the desired network outputs the learning algorithm updates the network parameters such that the discrepancies in the current batch are minimized. Once all the batches are processed, the next epoch is initiated [23, 24].

3
Methods

In this chapter, the methods used in the thesis are presented. The chapter is divided into two sections. The first section presents the methods used in the feature design part of the thesis and the second section presents the methods used in the learning-based liveness classification part of the thesis. The software used when implementing the algorithms was MATLAB R2016b. In the second part of the thesis, the publicly available MATLAB toolbox MatConvNet [25] was also used.

3.1 Feature design for fingerprint liveness classification

In the feature design part of the thesis, a data set provided by Fingerprint Cards was used. The data set contained around 9000 images of live fingerprints and around 12000 images of wood glue spoofs. The sensor used to acquire these images is the fingerprint touch sensor FPC1025, which is a capacitive sensor generating normalized images of size 160×160 at a resolution of 508 dpi. The fingerprints in the live images originate from 240 unique fingers and the spoof images originate from the fake counterparts of these 240 fingers. Since the sensor used is significantly smaller than a fingerprint, the data set contains several images of each finger as an attempt to capture the entire fingerprint in the data set. The data set also contained information about pairs of images of the same fingerprints in which an overlap of the two prints has been established. The affine transformations needed to align the images in the pairs were also provided by Fingerprint Cards. There were 186 image pairs in which a spoof image was found to overlap with a live image and 1266 pairs of images in which two live images were found to overlap. The images in the first set of image pairs were used to identify and detect differences between live and spoof images and the images in the second set of image pairs were used as a reference.

The first step in the liveness feature design part of the thesis was to identify features which frequently differed between live and spoof images in the data set provided by Fingerprint Cards. By registering, or aligning, the images in each image pair containing one live image and one spoof image, the comparison was facilitated. During the visual inspection of the spoof images, the fingerprint valleys were found to be more discontinuous in the spoof images than in the live images. Such discontinuities result in spurious minutiae in the form of skeleton breaks, hence these discontinuities could be detected by analyzing the skeleton (see Section 3.1.6.1 for this analysis). Another difference found was that the contours of the ridges and valleys were somewhat more irregular in the spoof images than in the live images. Since irregular contours result in spurious minutiae in the form of spikes as well as in a more jagged skeleton, these can be captured by detecting spikes in the skeleton (Section 3.1.6.2) and by assessing the fingerprint skeleton curvature (Section 3.1.6.3). During the visual inspection of the image pairs, some tendencies related to the image intensity was also found. Thus, the intensities in the images were studied (Section 3.1.7.1). Included in these tendencies were that the ridges in the spoof fingerprint images quite often were found to be more gray than the ridges in their live equivalents. The valley intensities in the spoof images did also often differ from the valleys in their live equivalents, but in an inconsistent manner, i.e. the valley intensities in the spoof images were both brighter and darker than the valley intensities in the live fingerprint images. The spoof fabrication process is also believed to introduce a sharp shift between the ridges and valleys, hence the steepness of the intensity profile perpendicular to the ridge and valley pattern was assessed (Section 3.1.7.2). The width of the ridges and valleys were also found to differ between the live images and the spoof images. The width variations were investigated both by applying a distance transform to the binarized fingerprint images (Section 3.1.8.1) and by measuring the relative width of the ridges and valleys in the image pairs (Section 3.1.8.2). Since the frequencies present in the image relates to the ridge and valley width, the frequency content of the images was also studied (Section 3.1.8.3). The different parts of the analysis are explained in detail in the sections below.

3.1.1 Registration

The paired images were spatially aligned by transforming the coordinate system of one of the images in the pair into the coordinate system of the other image in the pair. Apart from differences in translation and rotation, two images of the same fingerprint may also be somewhat scaled due to differences in pressure between the finger and the sensor. Hence, an affine transformation is needed to align the images. The affine transformations needed to align the images in the data set were extracted from the matching algorithm used at Fingerprint Cards. In the image pairs containing one spoof and one live image, the spoof images were used as targets and the live images were used as source images. In the image pairs used as references, both the source and target images were images of live fingerprints. The images in the pairs were aligned by transforming, i.e. warping, the source images according to the affine transformations. The registration process is visualized in Figure 3.1.

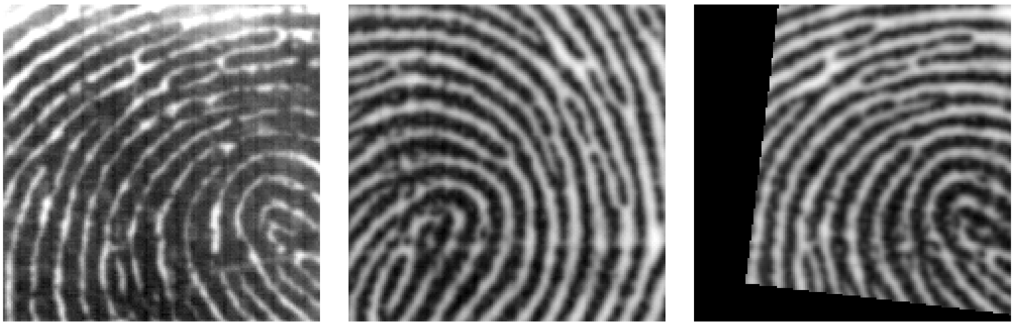


Figure 3.1: Visualization of the registration process. The left image is a spoof image used as target, the middle image is a live image used as source and the right image is the live image transformed to the coordinate system of the spoof image.

3.1.2 Visual interpretation

To identify differences within the aligned image pairs, the two images in each pair were compared by placing them in two separate color channels of an RGB-image. For each image pair, a color image was built by placing the grayscale live fingerprint image in the blue channel and the grayscale spoof fingerprint image in the green channel. Since the red channel is kept empty, the colors in the resulting RGB-image are restricted to different combinations of black, blue and green. The pixels in which both of the overlapped images have low intensities are displayed in black, while the pixels in which both of the overlapped images have high intensities are displayed in turquoise. The pixels in which the intensities in the spoof image differ from the intensities in the live image will either be more green or more blue. The pixels in which the live image have higher intensities than the spoof image will be more blue and the pixels in which the spoof image have higher intensities than the live image will be more green. Hence, by detecting more green and more blue pixel clusters, differences between the images were identified. In Figure 3.2, an example overlap image is shown together with the two grayscale images used to create it. The part of the image in which the live image lack information due to the affine transformation will be limited to combinations of green and black.

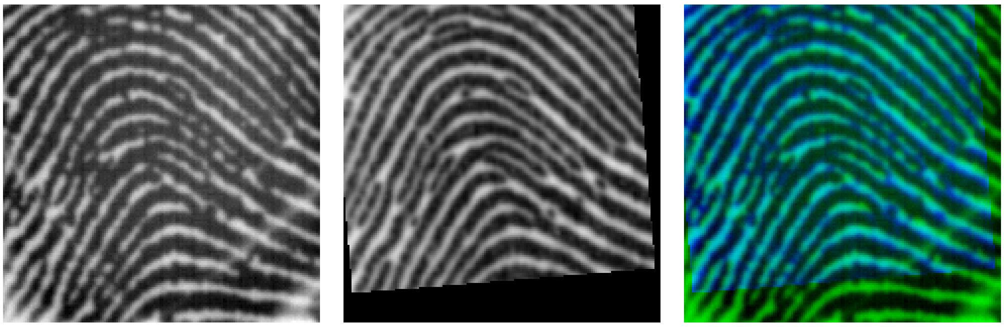


Figure 3.2: Example of an overlap image. The right image is created by placing the left and middle image in the green and blue channels of an RGB-image.

3.1.3 Segmentation

Segmentation of the fingerprint images was used to separate the foreground in the images from the background (see Section 2.2.1). This step was needed both to constrain the analysis to the fingerprint foreground and since the background pixel intensities otherwise would have affected the fingerprint binarization process. The fingerprint images were segmented using the method described in [18]. This method is based on the pixel features local gradient coherence, local intensity mean and local intensity variance. A linear classifier is used to assign the pixels to either foreground or background based on the feature vector calculated for each pixel. The pixel features were calculated by using a sliding neighborhood with block size 3×3. To avoid edge effects when calculating the features, the images were mirror-reflected across the image borders. By multiplying the calculated feature values in each pixel with the weights of the linear classifier, a segmentation probability map was created. The output from the linear classifier was then post-processed by applying morphological opening and closing on the segmentation probability map to remove small clusters and holes in the map. The structuring elements used in the opening and closing were disc-shaped elements. The diameter of the structuring element used in the opening operation was 3 pixels, while the diameter of the structuring element used in the closing operation was 5 pixels. In an additional post-processing step, all background clusters not connected to an image edge were discarded.

Segmentation masks were created for all the images analyzed in the liveness feature design part of the thesis. Since one of the images in each pair was transformed into the coordinate system of the other image in the pair during the analysis, the corresponding segmentation masks were transformed in the same manner. A joint segmentation mask was then created for each image pair by calculating the intersection of the aligned segmentation masks for each image pair. In order to avoid edge effects in the analysis of the images, reduced versions of the segmentation masks were also created by filtering the joint masks with a minimum filter of size 7×7. Since zero-padding was used during filtering, this operation removed a three-pixel wide segment around the entire foreground of each mask.

The segmentation of the fingerprint images was evaluated both by manual inspection and by calculating the Dice similarity coefficient for fingerprint images segmented with the implemented algorithm and the gold standard, which in this case are fingerprint images that are segmented manually. The Dice coefficient is a commonly used measure to evaluate the similarity between image sets and it is defined by Equation 3.1. This coefficient ranges between zero and one and a Dice coefficient of zero indicates that there is no similarity between the two images while a Dice coefficient of one indicates that the two images are identical. In Equation 3.1, X and Y corresponds to automatic and manual (i.e. gold standard) segmentations.

$$\text{Dice coefficient} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{3.1}$$

The gold standard used when calculating the Dice coefficient was obtained by manually labelling 30 fingerprint images in the data set. All 30 images contained reasonably large background areas and none of them belonged to any of the image pairs used in the feature design analysis.

3.1.4 Fingerprint orientation

The orientation in the fingerprint images (see Section 2.2.2) was estimated by implementing the method proposed in [19]. In this method, the directional field is estimated by calculating the dominant gradient direction in the neighborhood surrounding each pixel. Since the directional field is perpendicular to the dominant gradient direction, the conversion between the two is trivial. The implemented algorithm followed the general concepts of this method, but some alterations were made to the proposed filtering steps. First, the gradients in each image were obtained by filtering the fingerprint images with a Gaussian derivative filter of size 7×7 with standard deviation 1. The principal gradient directions were then estimated by applying Principal Component Analysis (PCA) to the covariance matrix of the gradient components as proposed in [19]. The traditional averaging proposed for this step was however replaced with a smoothing operation using a Gaussian filter of size 19×19 with standard deviation 3. The components of the directional vector field were then calculated and after an additional smoothing of the vector field components, again using a Gaussian filter of size 19×19 with standard deviation 3, the dominant gradient directions was calculated from the smoothed vector components. Finally, the fingerprint orientation was calculated by adding $\pi/2$ to the dominant gradient direction estimations in each pixel since the orientation and the dominant gradient direction are orthogonal.

3.1.5 Fingerprint binarization

The fingerprint images were converted to binary images by replacing all pixel values above an image specific threshold with ones and by replacing all the other pixel values with zeros. The thresholds were calculated using Otsu's method (see Section 2.2.3) in order to maximize the inter-class variations between the ridges and valleys in each image. Only the values of the pixels assigned to the image foreground in the segmentation process were used to calculate the thresholds. A global threshold was preferred over a local threshold since local thresholds may introduce artifacts to the binary images. Since the fingerprint valleys are the brighter pixels in the fingerprint pattern, the foreground of the resulting binary image mostly contained fingerprint valleys. The complement of each image was also produced in order to obtain images in which the foreground contained the ridges of the fingerprint patterns.

3.1.6 Fingerprint skeletons

Fingerprint skeletons were produced by performing a morphological skeletonization of the binary fingerprint images (see Section 2.2.4). Ridge skeletons were produced from the binary images in which the foreground consisted of ridges and valley skeletons were produced from the binary images in which the foreground consisted of valleys. In order to constrain the analysis of the images in an image pair to the same part of the fingerprint, the skeleton foreground pixels which coincided with background pixels of their joint segmentation mask were converted to the skeleton image background.

3.1.6.1 Detection of skeleton breaks

Breaks in the fingerprint skeletons were detected by identifying end points in the skeleton and by finding pairs of end points in close proximity to each other. The number of skeleton breaks in the ridge skeletons and the valley skeletons were calculated for both images in each image pair. The end points in an image were identified by finding the pixels with crossing number one. Isolated islands in the skeletons were also considered as end points, hence the pixels for which the crossing number was zero were also considered as end points. Any end points outside the foreground of the corresponding joint and reduced segmentation mask were however ignored in order to avoid detection of false skeleton end points. All pairs of end points which were closer than 20 pixels apart were considered as possible skeleton breaks. The number of possible breaks in an image was then narrowed down by removing all pairs of end points which were connected by the skeleton. In order to prevent improper break detections due to incorrect fingerprint orientation estimations like the ones shown in Figure 3.3, an assessment of the skeleton image patch in between the two end points in each pair was also necessary. If there were any skeleton pixels in the skeleton image patch between the two end points in a pair the possible break was discarded. Since each identified end point only was allowed to be part of one skeleton break, the remaining point pairs were assessed to ensure that all end points in the final pairs were unique. In order to grade the possible breaks, the absolute difference between the angle of the vector connecting the two points in each pair and the estimated fingerprint orientation in each neighborhood was calculated. The possible breaks were sorted in ascending order based on how well the angle of the point vector and the estimated fingerprint orientation agreed. The remaining point pairs were assessed one by one and a point pair was only accepted as a skeleton break if the points in it did not belong to a pair in which the angle of the point vector and the estimated fingerprint orientation better coincided. All point pairs in which the difference between the angle of the point vector and the estimated orientation were larger than $\pi/5$ were discarded. Since the size of the analyzed area in the different image pairs varies widely due to differences in the overlap between the two original images, the number of detected skeleton breaks found in an image had to be normalized with respect to the analyzed area in order to obtain comparable results. Hence, the normalization was performed by dividing the number of breaks found in an image with its total number of skeleton pixels in the analyzed area.

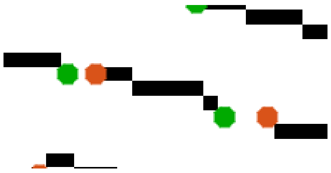


Figure 3.3: Incorrect skeleton breaks. This image is an example showing two skeleton breaks which are improperly detected due to incorrect fingerprint orientation estimations. The points in one of the detected skeleton breaks are marked in green and the points in the other detected skeleton break are marked in orange.

3.1.6.2 Detection of skeleton spikes

Spikes in the fingerprint skeletons were detected by identifying bifurcations and end points in the skeleton and by finding pairs of the two types of points in close proximity to each other. The number of skeleton breaks in the ridge skeletons and the valley skeletons were calculated for both images in each image pair. Bifurcations and end points in an image were identified by finding the pixels with crossing number three and one respectively. The end points in an image already counted in a skeleton break were not allowed to be part of a skeleton spike, hence these were removed from the list of identified end points. Bifurcations and end points outside the foreground of the corresponding reduced joint segmentation mask were also ignored in order to avoid detection of false minutiae points. All pairs of bifurcations and end points which were closer than 10 pixels apart were considered as possible skeleton spikes. By removing all point pairs not connected by the skeleton, the number of possible spikes in an image were narrowed down. Since each bifurcation and end point only were allowed to be part of a single skeleton spike, the remaining point pairs were assessed to ensure that all points in the final spikes were unique. If a bifurcation was connected to more than one end point, the pair with the shortest Euclidean distance between the points was considered as most probable to be a skeleton spike. Just as when grading the possible skeleton breaks, the possible spikes were graded by calculating the absolute difference between the angle of the vector connecting the two points in each pair and the estimated fingerprint orientation in each neighborhood. However, since the angle between the bifurcation and the end point defining a skeleton spike should be somewhat orthogonal, the possible spikes were sorted in ascending order based on how close the difference between vector angle and estimated neighborhood orientation came to $\pi/2$. The remaining point pairs were then assessed one by one and a point pair was only accepted as a skeleton spike if the points in it were not already part of a pair in which the angle of the point vector and the estimated orientation better coincided with $\pi/2$. All point pairs in which the difference between the angle of the point vector and the estimated fingerprint orientation were further away than $\pi/5$ from $\pi/2$ were discarded. Since the number of skeleton bifurcations also was found to be a feature which differed between live skeletons and spoof skeletons, the number of detected bifurcations in a fingerprint skeleton that were not already part of a detected spike was also saved for each image. Both the number of detected spikes and the number of detected bifurcations in an image were normalized by dividing with its total number of skeleton pixels.

3.1.6.3 Skeleton curvature

The curvature of a two-dimensional curve is a measure of how sharply the curve bends. It is defined as the magnitude of the rate of change of the unit tangent vector with respect to the curve length. The curvature at a point on a parametric curve given by $(x(s), y(s))$ is often calculated from the first and second derivatives of the curve at the point with respect to the parameter s using Equation 3.2 [26].

$$\kappa = \frac{|\, x'y'' - y'x'' \,|}{(x'^{\,2} + y'^{\,2}\,)^{\,2/3}} \tag{3.2}$$

If a curve is discrete, as in the segments of a fingerprint skeleton, a slight modification of the curvature definition is needed to compensate for sampling differences of the discrete points on the curve. In order to calculate the curvature at a point on a discrete curve, the relative positions of its predecessor and successor are needed. The derivatives in Equation 3.2 are first calculated by fitting the three sequential points to two polynomials and then the curvature at the points is estimated [26, 27].

The jaggedness of a fingerprint skeleton was evaluated by estimating the mean curvature for all its curve segments. The skeleton curvature was calculated for both images in each image pair. As in the previous sections, the analysis of the fingerprint skeletons of an image pair was constrained to the pixels corresponding to the foreground area of the joint segmentation mask. Since the curvature at skeleton bifurcations and skeleton crossovers would shadow the smaller curvature variations originating from skeleton jaggedness, the skeleton pixels in a patch of size 3 × 3 around each bifurcation and crossover were removed from the skeleton before the curve segments were extracted from the skeleton image. Groups of connected pixels were extracted by identifying all 8-connections in the skeleton. The curvature was calculated for each curve segment of connected skeleton pixels containing at least three connected pixels. The absolute curvature values in all the points along each curve segment were summed up and divided by the number of estimated curvature values in the same segment. At last, a measure of the curvature in an entire fingerprint skeleton was obtained by taking the mean of the curvature values of all its skeleton curve segments.

3.1.7 Fingerprint intensities

The intensity of the fingerprint images was studied in three different ways. The intensity distributions of the fingerprint images were studied both by evaluating all intensity values in the ridges and valleys and by evaluating the image intensity values in the pixels along the fingerprint skeletons. Since the spoof fabrication process is believed to introduce a sharper shift between the ridges and valleys, the steepness of the intensity profile perpendicular to the ridge and valley pattern was also assessed.

3.1.7.1 Intensity distributions

The intensity distribution analysis was performed both by calculating the mean of the intensities along the ridges and valleys in each fingerprint image and by calculating the mean of the intensities along the ridge and valley skeletons of each fingerprint image. The mean intensity values were calculated for both images in each image pair and the analysis was constrained to the pixel values corresponding to the foreground area of the joint segmentation mask.

3.1.7.2 Intensity profiles

The intensity steepness in the transition between ridges and valleys in the fingerprint images was evaluated by studying the intensity along a line perpendicular to the ridges and valleys in the fingerprint pattern. The intensity profile analysis was performed on both images in each pair and for each image both the ridge-valley-ridge intensity profiles and the valley-ridge-valley intensity profiles was evaluated. In order to evaluate the intensity profiles in an image, observation lines perpendicular to the ridges and valleys had to be created. When the ridge-valley-ridge intensity profiles in an image were evaluated, the intensity profiles were extracted by creating lines which passed through each foreground pixel in the valley skeleton. In the same manner, when the valley-ridge-valley intensity profiles in an image were evaluated, the intensity profiles were extracted by creating lines which passed through each foreground pixel in the ridge skeleton. Each observation line was perpendicular to the fingerprint orientation estimated in the skeleton pixel it intersected. If the obtained line was too sparsely sampled, it was discarded since such lines would result in too imprecise measurements. The pixels in an intensity profile were defined as the segment of the line between the mid points of the two adjacent ridges or valleys of the skeleton pixel. Such a line segment is visualized in Figure 3.4. These two end points were identified by studying the binary version of the fingerprint image. The pixel values along the line in the binary image were grouped into connected elements and from this information the mid pixels of the adjacent ridges or valleys were identified. If any of the ridges and valleys which were transversed by the line segment were wider than a certain threshold, the ridges or valleys were however assumed to be accreted. Such measurements were also discarded. The width of a fingerprint ridge is seldom larger than 300 μm [2], but the threshold was set to 425 μm since the observed width of the ridges and valleys in the fingerprint images also is affected by the pressure between the finger and the sensor during capture. In a fingerprint image obtained from a fingerprint sensor with 508 dpi resolution, 425 μm correspond to 8.5 pixels. The intensities along the line segment between the two end points were then extracted from the original fingerprint image and from these intensity values the intensity gradient along the line segment was calculated. The gradient values were normalized with respect to the Euclidean distance between the points on the line. The steepness of the intensity transition was evaluated by integrating the gradients, i.e. squaring the gradient values along the line segment and dividing the sum of the squared values with the amount of points in the line segment. The mean value of all the integrated intensity profile gradients was then calculated for each fingerprint image.

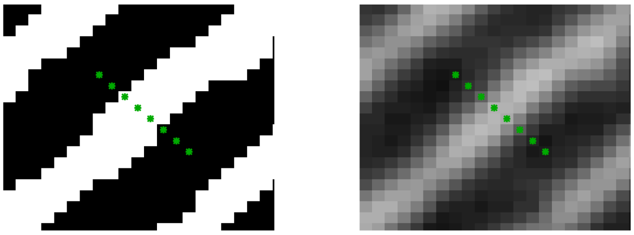


Figure 3.4: Intensity profile example. This figure visualizes the points along the line segment from which an example intensity profile is calculated. The left image visualizes the line segment in green together with the binary version of the intensity profile neighborhood, while the right image visualizes the line segment in green together with the original version of the neighborhood.

3.1.8 Ridge and valley width

The valleys in the spoof images were often found to be thinner than the valleys in the live images in the initial visual inspection. Equivalently, the ridges in the spoof images were often found to be wider than the ridges in the live images. The variations in ridge and valley width were investigated in three different ways. First, the average ridge and valley distance from all the valley and ridge pixels in the images were calculated with a distance transform. The average ridge and valley distances provide information about the width since they for an ideal fingerprint pattern correspond to one fourth of the ridge and valley width respectively. Second, a relative comparison of the ridge and valley width in the two images in each pair was made. Third, the magnitude of the frequency spectrum of each image was studied since the frequencies present in the image relates to the appearance of the fingerprint ridges and valleys. The methods used to evaluate the ridge and valley width are described in more detail below.

3.1.8.1 Distance transforms

By applying a distance transform to the binary versions of the fingerprint images the Euclidean distance to the closest ridge or valley in each pixel was obtained. The distance transform was applied on both images in each pair and again, the analysis of the images in each image pair was constrained to their foreground area of the joint segmentation mask. A distance transform of a binary image returns a distance map in which the value of each pixel corresponds to the distance to the closest foreground pixel in the image. Hence, the distance to the closest ridge for each valley pixel in an image was evaluated by applying the distance transform to the binary image in which the ridges belong to the foreground. Conversely, the distance to the closest valley for each ridge pixel in an image was evaluated by applying the distance transform to the binary image in which the valleys belong to the foreground. The average distance to the closest ridge in all the valley pixels in each image and the average distance to the closest valley in all the ridge pixels in an image was then calculated. Distances exceeding a certain threshold, which was set slightly higher than half the largest acceptable ridge and valley width (4.25 pixels), were however discarded before the averaging since such measurements originated from areas in the fingerprint images where the ridges or valleys in the binary images were not represented correctly.

3.1.8.2 Relative ridge and valley width

The relative width of the ridges and valleys in an image pair was estimated by pairwise comparisons of the ridge and valley widths. For each image pair, the overlapping foreground pixels in their skeletons determined the locations for which the width was assessed in. The ridge skeletons were used to find joint skeleton pixels to assess the ridge width in and the valley skeletons were used to find joint skeleton pixels to assess the valley width in. In order to evaluate the ridge or valley width in these joint pixels, lines which passed through each joint skeleton pixel with an angle perpendicular to the fingerprint orientation estimated in the same pixels were created. By extracting the values of the line pixels from the corresponding binary images and by grouping these values into connected components, the first and last line pixel within the current ridge or valley was identified. The Euclidean distance between these two pixels on the line was calculated and then the ratio between the distances calculated for the same skeleton pixel in the two images of an image pair was determined. If an estimated ridge or valley width was wider than a certain threshold (set to 8.5 pixels), the measurement was however discarded since the ridge or valley then was assumed to be accreted with adjacent ridges or valleys. Other constraints put on the measurements were that the angles of the two lines intersecting the same joint skeleton pixel in an image pair were not allowed to differ more than $\pi/12$ and that the lines were not allowed to be too sparsely sampled. The relative width for each joint ridge skeleton pixel was calculated as the ratio between the estimated ridge width in the image used as target and the estimated ridge width in the warped image. The relative width for each joint valley skeleton pixel was calculated as the ratio between the estimated valley width in the warped image and the estimated valley width in the image used as target.

3.1.8.3 Frequency spectrum analysis

The frequency content of the fingerprint images was evaluated by studying the magnitude of their frequency spectra. The edge effects in the frequency spectra were minimized by mirror reflecting the fingerprint images before converting them to the frequency domain. In order to compare the frequency spectra of the live and spoof images, the two-dimensional magnitudes of the frequency spectra were converted to a one-dimensional signal representing the radial average of the frequency magnitudes. Since the one-dimensional signal is created by computing the frequency content average along the radius, the frequency content in the corners of the frequency spectra magnitudes is ignored. The corners of the frequency spectra magnitudes for the fingerprint images do however only contain noise, thus no important information will be lost in this conversion. The radial averages of the frequency spectra magnitudes were calculated for all source and target images in the image pairs. An average one-dimensional frequency spectrum magnitude signal was then calculated for all the source and target fingerprint images for both the image pairs containing one live and one spoof image and the image pairs in the reference data.

3.1.9 Statistical significance

The statistical significance of the detected features was evaluated by performing statistical t-tests on the feature value distributions obtained for the image pairs which contained both live and spoof fingerprint images. Paired-sample t-tests were used for the skeleton-based features, the intensity distributions and the distributions obtained from the distance transform analyses, while a two-sample t-test was used for the relative ridge and valley width analyses. All distributions were assumed to be normal distributions, but no assumptions regarding the variances were made. The significance level used was 1% for all the features except for the relative valley width where a significance level of 2.5% was used instead. The higher significance level was used since the relative valley width result was not significant at the 1% significance level. One sided t-tests were used in most cases, but since the obtained results for the skeleton curvature and the intensity profile analyses were opposite to the expected, the significance of the difference between the live and spoof distributions from these analyses was evaluated with a two-sided t-test. The null hypothesis in the t-tests was that the mean of the spoof distribution equals the mean of the live distribution. The alternative hypothesis in the one-sided t-tests was that the mean of the spoof distribution was greater than the mean of the live distribution. Theses hypotheses are specified in Equation 3.3. The difference between the distributions was only considered significant if the null hypothesis could be rejected.

$$\begin{aligned} H_0 &: \mu_{spoof} = \mu_{live} \\ H_1 &: \mu_{spoof} > \mu_{live} \end{aligned} \tag{3.3}$$

3.1.10 Spoof image quality

The quality of the spoof images was estimated using a binary support vector machine model in which ten of the designed liveness features were used as predictors. These predictors were chosen by manually identifying the ten features which best separated the investigated live and spoof images. Features which did not show a statistically significant difference between the live and spoof images were not considered to be relevant to include in the quality assessment model. The features chosen as predictors in the model are given in the list below. The fact that some of these features were correlated was ignored when they were selected as model predictors.

- breaks in the valley skeleton
- spikes in the ridge skeleton

- spikes in the valley skeleton
- bifurcations in the ridge skeleton
- bifurcations in the valley skeleton
- ridge skeleton curvature
- ridge skeleton intensity
- valley-ridge-valley intensity profiles
- ridge-valley-ridge intensity profiles
- distance to the closest valley

The binary support vector machine model aims to find a hyperplane that successfully separates the ten-dimensional predictor vectors in an optimal sense. The hyperplane equation is given by Equation 3.4. In this equation, $f(\mathbf{x})$ is the classification score, $\mathbf{x}$ is the feature vector consisting of the chosen predictors, $\boldsymbol{\beta}$ is a vector containing the model weights and b is a bias term. The separating hyperplane corresponds to the feature vector which satisfies $f(\mathbf{x}) = 0$. A feature vector is classified as belonging to one of the two classes depending on the sign of its classification score, i.e. $f(\mathbf{x}) > 0$ implies a live classification and $f(\mathbf{x}) < 0$ implies a spoof classification.

$$f(\mathbf{x}) \ = \ \mathbf{x}'\boldsymbol{\beta} + b \tag{3.4}$$

The support vector machine model was trained on the standardized feature vectors obtained for all the images used for feature design (i.e. the pairs containing both live and spoof images) in order to obtain the separating hyperplane which best separated the live and spoof images based on the ten predictors. When the model was estimated, the spoof images in the set were classified with the model. The relative quality of the spoof images was then estimated from their classification scores. The support vector machine classification score is defined as the signed distance from the decision boundary of the separating hyperplane. These scores were mapped into posterior probabilities by an optimal score-to-posterior-probability transformation function. For inseparable classes, as in this case, this transformation is a sigmoid function. The quality of a spoof image is regarded as high if the posterior probability of the live class is high and the quality of a spoof image is regarded as low if the posterior probability of the spoof class is high. The quality of an entire set of spoof images may be defined as the percentage of these images that the separating hyperplane model manages to classify correctly. The separating hyperplane model could of course also estimate the quality of the live images in a similar manner, but such an evaluation is outside the scope of this thesis.

3.2 Learning-based fingerprint liveness classification

In the second part of the thesis, convolutional neural networks for fingerprint liveness detection was implemented since the spoof quality in a data set also can be estimated from how well a state-of-the-art fingerprint liveness classifier performs on the spoof images in the set. A set of fingerprint images is considered to be very hard to classify if the state-of-the-art network cannot differentiate between the live and fake images in the set. Conversely, if a shallow network easily differentiates spoof images from live images, these images are considered easy to classify.

The first implemented network was an imitation of the network which won the last Fingerprint Liveness Detection Competition in 2015 [6, 28]. However, since the images in the data sets used to train this model differed a lot from the images in the data set provided by Fingerprint Cards due to differences in sensor technology, the state-of-the-art network did not perform well on the images which were assessed in the feature design part of the thesis. A second network was thus trained and evaluated on the data provided by Fingerprint Cards. The idea was to estimate the spoof quality in a data set by evaluating the number of correctly classified spoof images in the set and to estimate the quality of the individual images in the set by evaluating the degree of liveness assigned to the spoof images by the neural network.

The data sets which were used to train the network in [6] were provided by the Liveness Detection Competition. These data sets were the training and testing sets used in the competitions in 2009 [29], 2011 [30] and 2013 [31]. In total, these data sets contained more than 50000 live and spoof fingerprint images obtained from eight different optical fingerprint sensors. The spoof fingerprint images were obtained from spoofs of eight different spoof materials. Of these images, around 40% belonged to the training set and the remaining 60% belonged to the test set. In both the training set and the test set, there were an equal amount of live and spoof fingerprint images. The sensors used came from Biometrika, Crossmatch, Digital, Identix, Italdata, Sagem and Swipe and the different spoof materials were body double, ecoflex, gelatin, latex, modasil, Play-Doh, silicone and wood glue. The spoofs used to create the spoof fingerprint images in the data sets were fabricated using both the cooperative and the non-cooperative method.

The data set which was used to train the second network was the same data set as the one used in the liveness feature design part of the thesis. This data set was provided by Fingerprint Cards and contained around 9000 images of live fingerprints and around 12000 images of wood glue spoofs. The spoofs used to create the images in the data set were fabricated using the non-cooperative method. Around 3000 randomly chosen spoof images were removed from the data set in order to obtain a balanced training set. The images in the data set which belonged to any of the live-spoof image pairs analyzed in the first part of the thesis were assigned to the test set and the remaining images were assigned to the training set. Among the image pairs, there were 162 unique live images and 172 unique spoof images, hence the test set for the second network contained 334 fingerprint images.

The networks were trained by fine-tuning the 19-layer deep pre-trained convolutional neural network developed by [32] for image classification. Their model, denoted as VGG, achieved second place in the classification task of the ImageNet Large Scale Visual Recognition Competition in 2014. The network was trained on 1.3 million images of 1000 different classes. Even though none of these classes were related to fingerprints or liveness detection, a fine-tuning of this pre-trained network on live and spoof fingerprint images has been proven to be an efficient approach to obtain a liveness detection classifier [6]. Of the 19 layers in the pre-trained network, the first 16 of them were 3×3 convolutional layers and the last three were fully connected layers. Each convolutional layer was followed by a rectified linear unit (ReLU). There were also five 2×2 pooling layers in between some of the convolutional layers. The pre-trained network was trained for 1000 different classes, hence the last layer of the network had to be converted from a 1000-unit softmax layer to a 2-unit softmax layer. The number of classes available in the network was consequently reduced to two, one for live fingerprint images and one for spoof fingerprint images. The softmax layer was during training converted to a softmax loss layer, which is a combination of a loss function and a softmax. These were the only changes made to the pre-trained network before fine-tuning it with fingerprint images. A schematic view of the network architecture is presented in Figure 3.5.

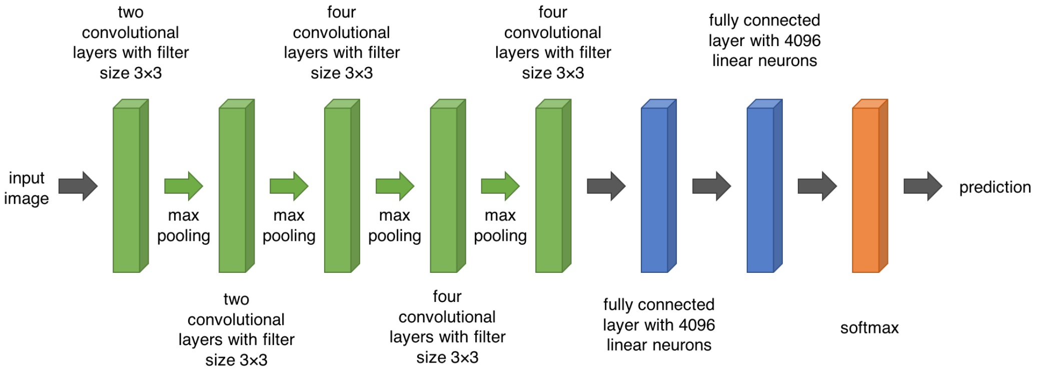

two convolutional layers with filter size 3×3
four convolutional layers with filter size 3×3
four convolutional layers with filter size 3×3
fully connected layer with 4096 linear neurons
input image
max pooling
max pooling
max pooling
max pooling
prediction
two convolutional layers with filter size 3×3
four convolutional layers with filter size 3×3
fully connected layer with 4096 linear neurons
softmax

Figure 3.5: Network architecture. In this image, the architecture of the utilized convolutional neural network is presented. The input to the network is the image to be classified and the output of the network is its prediction. The green building blocks correspond to the convolutional layers, while the green arrows in between these blocks represent the pooling operations. Each green building block contains at least two convolutional layers. The blue building blocks correspond to the fully connected layers and the orange building block corresponds to the softmax layer.

The fingerprint images in the data sets were all grayscale images and their size ranged between 160 × 160 to 700 × 800 pixels depending on the used sensor. Since the input to the pre-trained model was required to be of size 224 × 224 × 3, the first two dimensions of the images in the data sets were resized accordingly and the third dimension requirement was fulfilled by placing copies of the resized grayscale image in all three RGB-channels. The intensity mean of all the images in the training set was removed from the images in the set before the training was initialized. The intensity mean of the training set was also removed from the test set before testing. The data set augmentation proposed by [6] could however not be implemented since the training and testing variables used as input to the network, in that case, exceeded the allowed variable size in MATLAB (8 GB).

During the training of the convolutional neural network, stochastic gradient descent with momentum was used to optimize the network parameters using the backpropagation algorithm. As specified in [6], the learning rate and momentum used was 10^{-6} and 0.9 and the batch size used was 5. The training of the network was continued until the network ceased to learn, i.e. when the training error stopped decreasing. The network was however saved after each completed epoch, so when the training was done the final network could be chosen such that its validation error was low at the same time as the number of epochs was kept to a minimum. The training of the first network was ran for 35 epochs and the final network was the network obtained after 30 epochs. The training of the second network was ran for 15 epochs and the final network was the network obtained after 11 epochs. When the final network was identified, the network performance was evaluated on the test set. The classification performance of the obtained networks was evaluated by calculating the spoof false positive rate *(SFPR)*, the spoof false negative rate *(SFNR)* and the average classification error *(ACE)* of the test sets. The spoof false positive rate is defined as the percentage of misclassified live images, the spoof false negative rate is defined as the percentage of misclassified spoof images and the average classification error is defined as the average of the two.

4

Experimental results and analysis

4.1 Feature design for fingerprint liveness classification

In the following sections, the experimental results of the liveness feature design part of the thesis are presented and analyzed. In the first section below, a visual motivation of the performed analyses is presented. This section is followed by a section in which the segmentation results are presented and by the sections in which the results from the designed fingerprint liveness features are presented.

4.1.1 Visual interpretation

During the visual inspection of the pairs of live and spoof fingerprint images, several detectable features were identified. Some of these features can be seen in the three overlap images presented in Figure 4.1. The most frequent feature found was valley discontinuities in the spoof images. This feature can be seen as blue interruptions of the turquoise valleys in all three images in Figure 4.1. By studying these images, it can also be seen that the ridge and valley edges in the spoof images are somewhat more irregular than the ridge and valley edges in the live images.

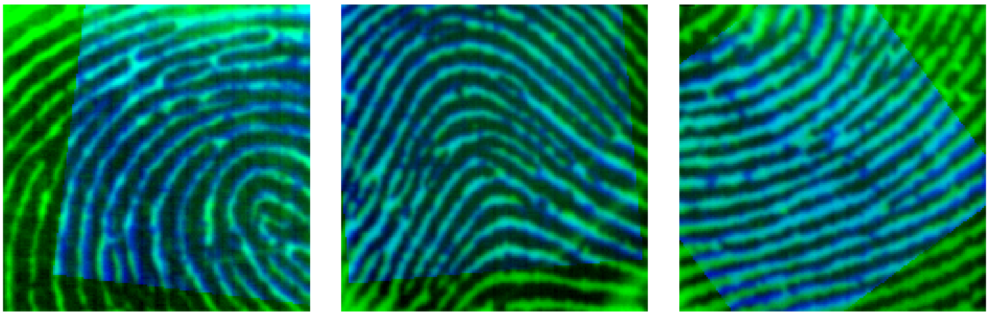


Figure 4.1: Visualization result examples. The overlap images from three image pairs are presented in order to visualize some of the features detected when comparing images obtained from live fingers and images obtained from spoofs. Recall that blue and green colors originate from the live and spoof images respectively.

Some tendencies regarding the fingerprint image intensities were also noticed. Since the ridge color in the overlap images often was a mixture of black and green, the ridges in the spoof fingerprint images were, in general, brighter than the ridges in the live fingerprint images. The color of the valleys in the overlap images was quite often a more blue shade of turquoise which implied that the valley intensity in the spoof images often was lower than the valley intensity in the live images. The greenish ridge color can be seen in all three images in Figure 4.1, while the bluer shade of turquoise is especially apparent in the right image.

Thinner valleys, or equivalently wider ridges, in the spoof images were another quite frequently observed feature in the image pairs. At first glance, the blue appearance of the valleys in the left image in Figure 4.1 might seem only to be due to a significant difference in valley intensities between the live and the spoof image. However, from a closer inspection of Figure 4.2, in which this overlap image is displayed again together with the live and spoof images used to create it, it becomes clear that the blue appearance of the valleys in the overlap image is mostly due to the significant difference in valley width between the live image and the spoof image, even though there are intensity differences which also affect the appearance.

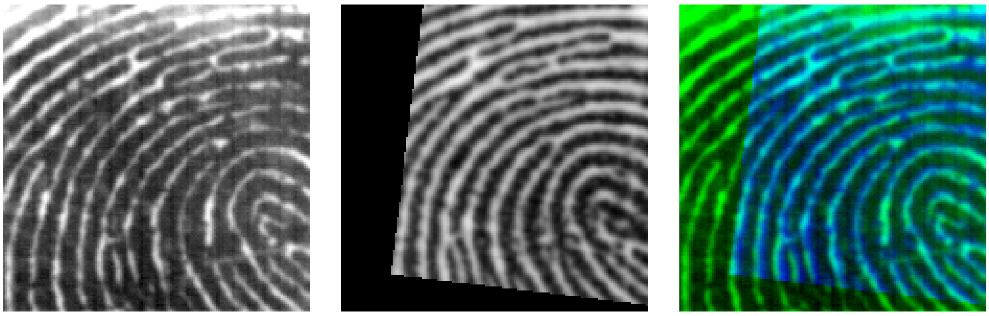


Figure 4.2: Visualization of an image pair in which there is a significant difference in ridge and valley width between the spoof image and the live image. The left image is the spoof image, the middle image is the live image and the right image is the overlap image created from the two.

4.1.2 Segmentation

This section presents the results of the fingerprint segmentation. In Figure 4.3, three example segmentation results are presented by displaying the fingerprint images together with the foreground borders in the corresponding segmentation masks. The Dice coefficient calculated for the test set containing 30 images ranged between 0.9763 and 0.9995 and the mean Dice coefficient for the set was 0.9921.

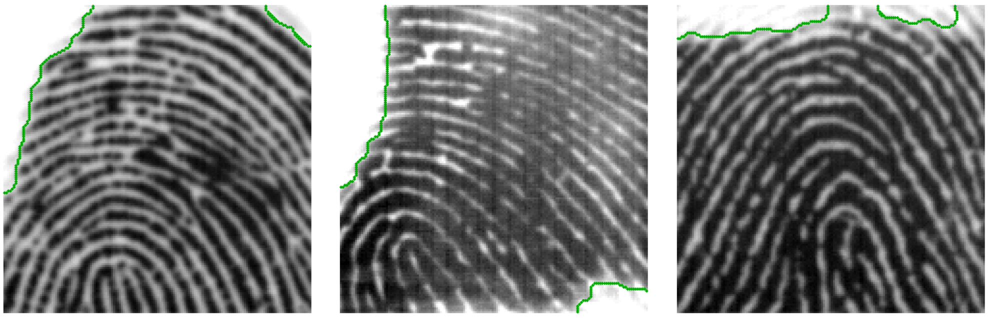


Figure 4.3: Segmentation result examples. Three fingerprint images are displayed together with green lines representing the borders of their segmentation masks.

From the results presented in Figure 4.3 and from the high Dice coefficients obtained in the evaluation of the test set, it is clear that the results provided by the implemented algorithm are most satisfactory. Since the algorithm extracts the foreground of the fingerprint images with an accuracy comparable to manual segmentation of the fingerprint images, the analysis is reliably constrained to the foreground area.

4.1.3 Skeleton-based features

In the following sections, the results from the detection of skeleton breaks, skeleton spikes and skeleton bifurcations are presented together with the results from the skeleton curvature analysis. In each section, the results are accompanied by an analysis of the obtained results. The skeletons of the fingerprint images turned out to contain several features which differed for live and spoof images, the most obvious differences are the amount of ridge skeleton bifurcations and valley skeleton breaks. The skeletons in Figure 4.4 and 4.5 clearly illustrates these significant differences.

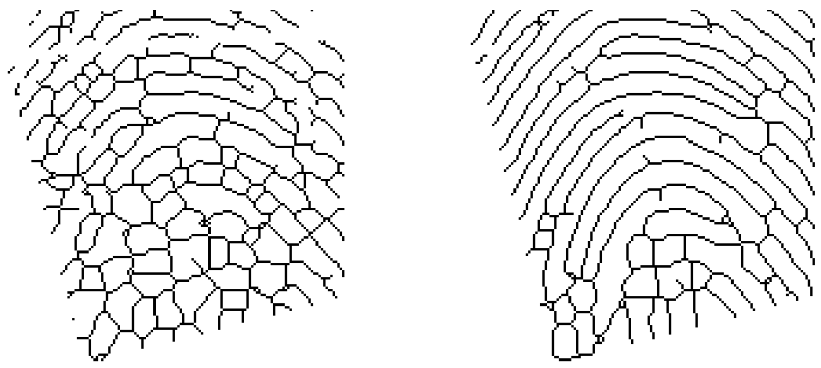


Figure 4.4: Ridge skeleton examples. The left image is a typical example of a ridge skeleton obtained from a spoof image and the right image is its live equivalent.

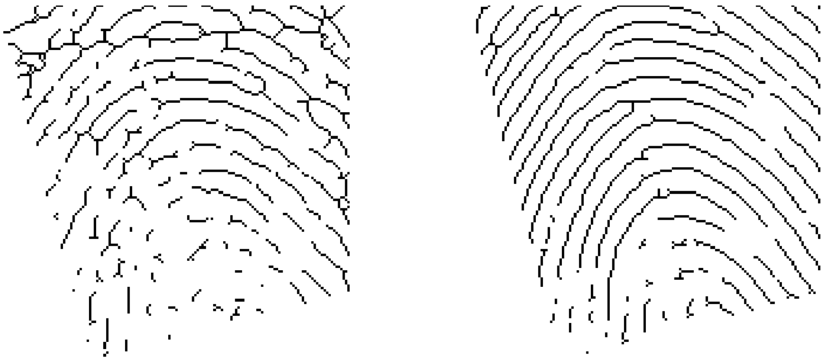


Figure 4.5: Valley skeleton examples. The left image is a typical example of a valley skeleton obtained from a spoof image and the right image is its live equivalent.

The analyses presented in the sections below were performed both on the set of image pairs containing one live and one spoof image and also on the set of image pairs containing two live images. The first set of image pairs were analyzed in order to study the differences between live and spoof fingerprint images, while the second set of image pairs were only used as reference. For each of the analyses presented below, the results for the two image sets are presented in the same figure but in separate histograms. Hence, each histogram figure contains two histograms and the left histogram visualizes the results obtained from the first set of image pairs, while the right histogram visualizes the results obtained from the second set of image pairs. In the left histograms, the distribution in green originates from the spoof images and the distributions in blue originates from the live images. Both distributions in the right histograms originate from live fingerprint images, the pink distributions originate from the live images used as targets in the registration process while the gray distributions originate from the warped images.

4.1.3.1 Skeleton break detection

The results from the skeleton break detection are presented in Figure 4.6 and 4.7. As previously stated, the left histogram visualizes the results obtained from the image pairs containing both live and spoof images, while the right histogram visualizes the results obtained from the image pairs containing two live images.

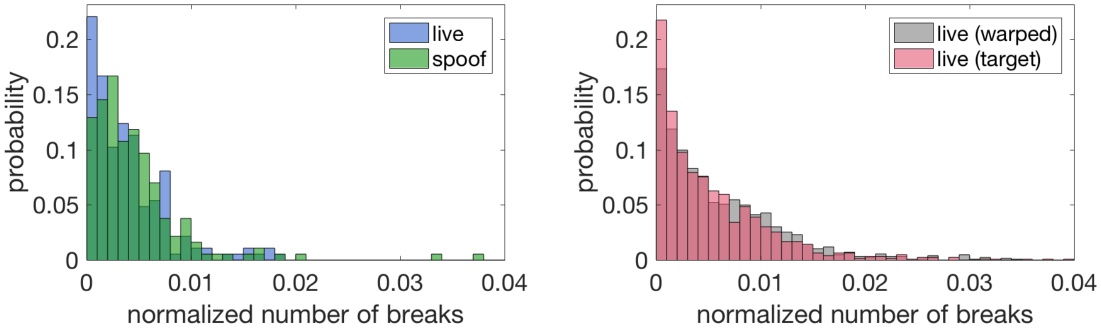


Figure 4.6: Histograms of detected ridge skeleton breaks. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

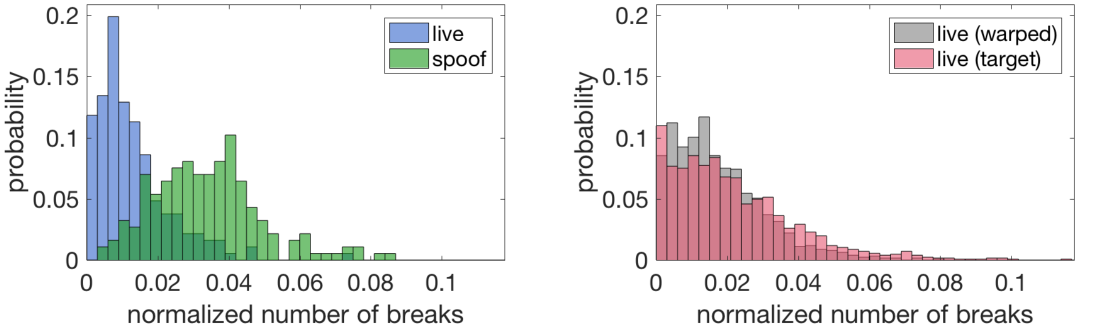


Figure 4.7: Histograms of detected valley skeleton breaks. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

In Figure 4.8 and Figure 4.9, the example skeletons previously presented in Figure

4.4 and 4.5 are displayed together with the detected breaks in each skeleton. From a closer inspection of these images, it can be noted that the break detection is not completely flawless. Even though the detected breaks are correct in most cases, there are a few examples of incorrect end point matches. These are probably a result of minor deficiencies in the estimated fingerprint orientation. Since the incorrect matches make up such a small proportion of the total number of detected breaks in an image, their effect on the final result is however considered to be small.

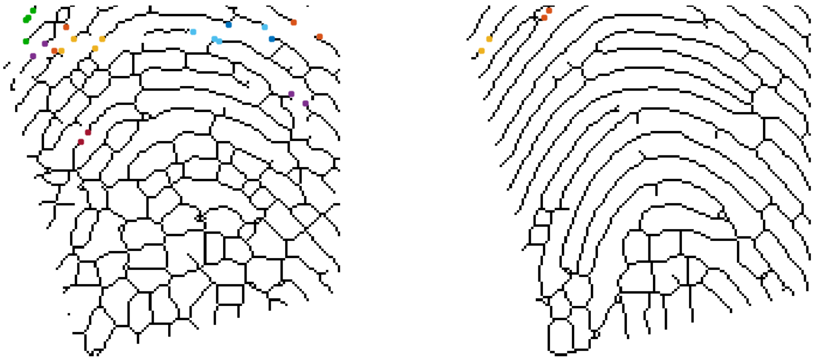


Figure 4.8: Detected ridge skeleton breaks. The left image visualizes the breaks detected in the skeleton obtained from a spoof image and the right image visualizes the breaks detected in the skeleton obtained from the corresponding live image. The two end points in each break are emphasized with markers of the same color.

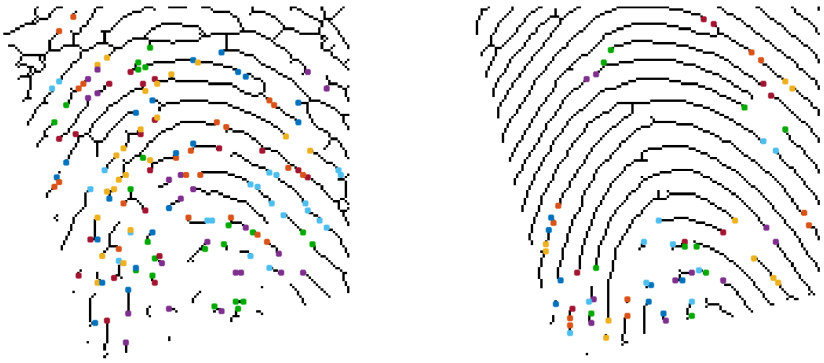


Figure 4.9: Detected valley skeleton breaks. The left image visualizes the breaks detected in the skeleton obtained from a spoof image and the right image visualizes the breaks detected in the skeleton obtained from the corresponding live image. The two end points in each break are emphasized with markers of the same color.

As anticipated, it is evident from the result presented in Figure 4.6 and 4.7 that there is a significant difference in the number of detected valley breaks between the live and spoof images but no significant difference in the number of detected ridge breaks. The difference in distribution seen in the left image in Figure 4.7 is significant at the $p = 0.01$ level. Hence, the number of valley breaks in an image is considered to be a valuable feature in the quality assessment of fingerprint images obtained from spoofs. The small differences in distribution seen in the reference data are most probably due to the natural sources of variation in fingerprint images such as the pressure against the sensor and the moisture level of the finger.

4.1.3.2 Skeleton spike detection

The results from the skeleton spike detection are presented in Figure 4.10 and 4.11. Recall that the results presented in the left histograms originate from the first set of image pairs, while the results presented in the right histograms originate from the second set of image pairs. In Figure 4.12 and 4.13, the example skeletons previously presented are displayed again together with the detected spikes in each skeleton.

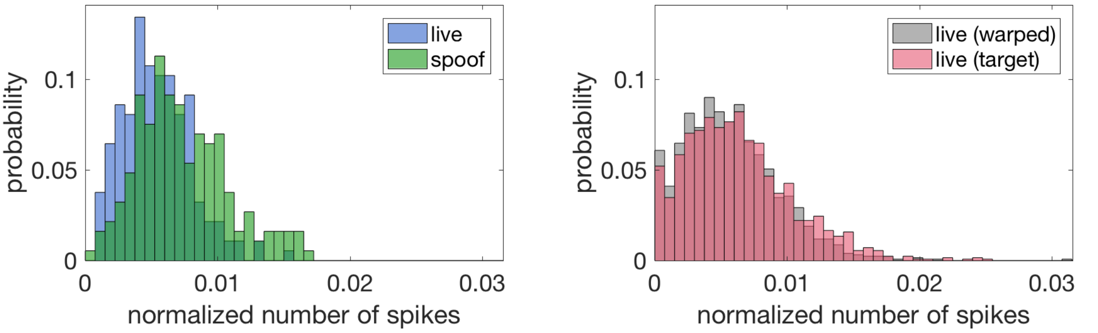


Figure 4.10: Histograms of detected ridge skeleton spikes. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

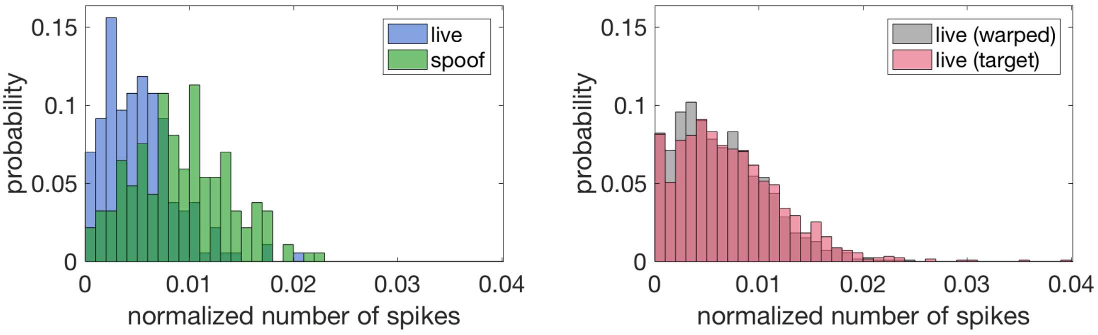


Figure 4.11: Histograms of detected valley skeleton spikes. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

From Figure 4.10 and 4.11, it is clear that there is a difference in the amount of skeleton spikes detected in the live and spoof skeletons. The differences in distribution seen in both left histograms are significant at the $p = 0.01$ level. The number of spikes in both fingerprint skeletons are thus considered as features which can contribute to the assessment of fingerprint images obtained from spoofs. The similar shift between the live and spoof distributions seen in these images is expected since irregularities in the ridge and valley interfaces in fingerprint images reasonably would generate spikes in both the ridge and valley skeletons. The correlation between the number of spikes in the spoof ridge and valley skeletons are however only 0.52, thus these features are both considered as valuable features in the spoof quality assessment.

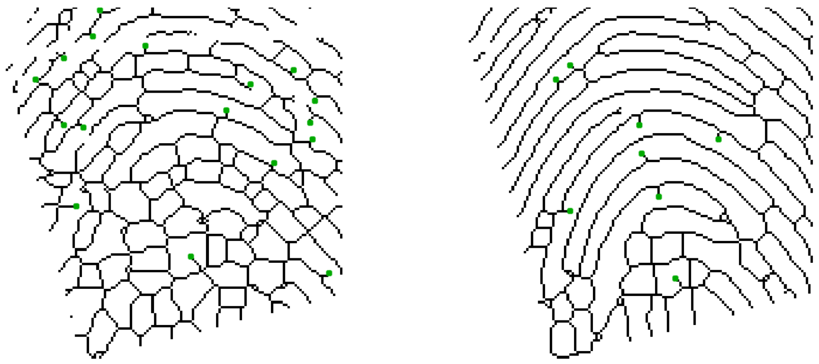


Figure 4.12: Detected ridge skeleton spikes. The left image visualizes the spikes detected in the skeleton obtained from a spoof image and the right image visualizes the spikes detected in the skeleton obtained from the corresponding live image.

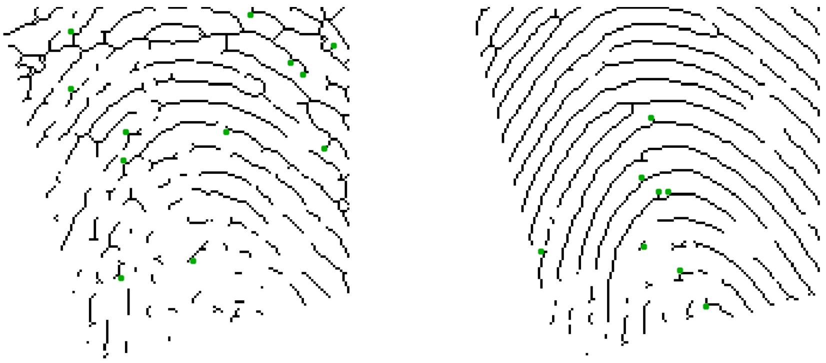


Figure 4.13: Detected valley skeleton spikes. The left image visualizes the spikes detected in the skeleton obtained from a spoof image and the right image visualizes the spikes detected in the skeleton obtained from the corresponding live image.

4.1.3.3 Skeleton bifurcations

In Figure 4.14 and 4.15 the histograms of the number of skeleton bifurcations not already part of a skeleton spike are presented. Recall that the results presented in the left histograms originate from the first set of image pairs, while the results presented in the right histograms originate from the second set of image pairs. By studying Figure 4.14 and 4.15, it is evident that there is an increase in the number of bifurcations in the spoof skeletons compared to the live skeletons. The differences in distribution seen in the left image are significant at the $p = 0.01$ level. Hence, the number of bifurcations in both skeletons are also considered as valuable features in the quality assessment of fingerprint images obtained from spoofs.

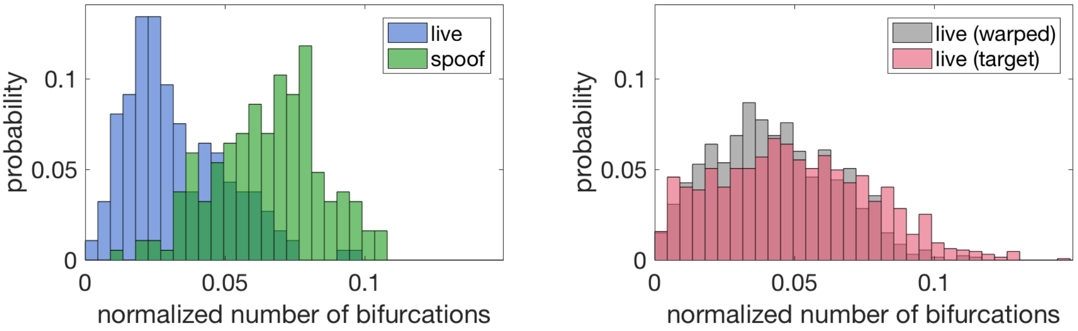


Figure 4.14: Histograms of the ridge skeleton bifurcations. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

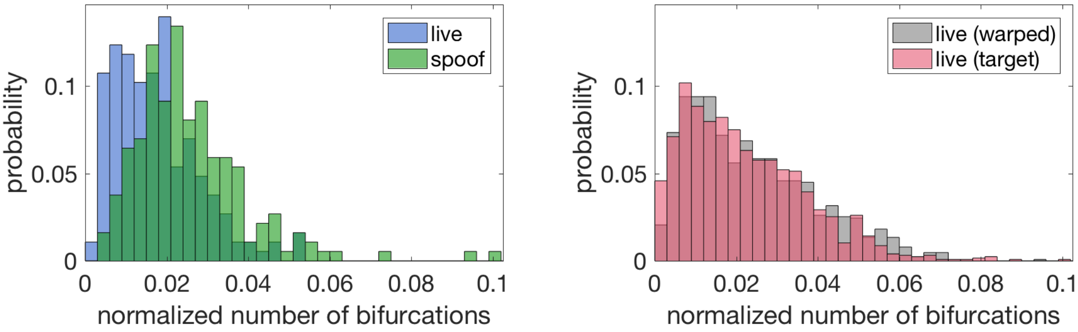


Figure 4.15: Histograms of the valley skeleton bifurcations. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

The considerable shift in distribution seen in the left histogram in Figure 4.14 is much expected since an increased number of breaks in the valley skeleton implies that there is an increased number of bridges between adjacent ridges. Each bridge between adjacent ridges result in two additional bifurcations in the ridge skeleton, hence an increased number of breaks in the valley skeleton is analogous to an increased number of bifurcations in the ridge skeleton. The correlation between the number of breaks in the spoof valley skeletons and the number of bifurcations in the spoof ridge skeleton is 0.86, thus a combination of these two features in the quality assessment of the fingerprint images will not yield an additive effect in the evaluation even though both features are considered valuable. The shift in distribution seen in the left histogram in Figure 4.15 is however not as expected. A possible explanation for this might be that the skeletonization process tends to split ridge and valley endings into two spikes. Since such spikes do not originate from irregular ridge and valley edges, they should not be detected as spikes and hence the bifurcation in a split ending is included in the bifurcation statistics. The correlation between the number of bifurcations in the valley skeleton and the number of end points in the valley skeleton is however only 0.14, which suggests that the increased number of bifurcations in the spoof valley skeletons is caused by other factors too.

4.1.3.4 Skeleton curvature

The results from the skeleton curvature analysis are presented in Figure 4.16 and 4.17. Again, the results presented in the left histograms originate from the first set of image pairs, while the results presented in the right histograms originate from the second set of image pairs.

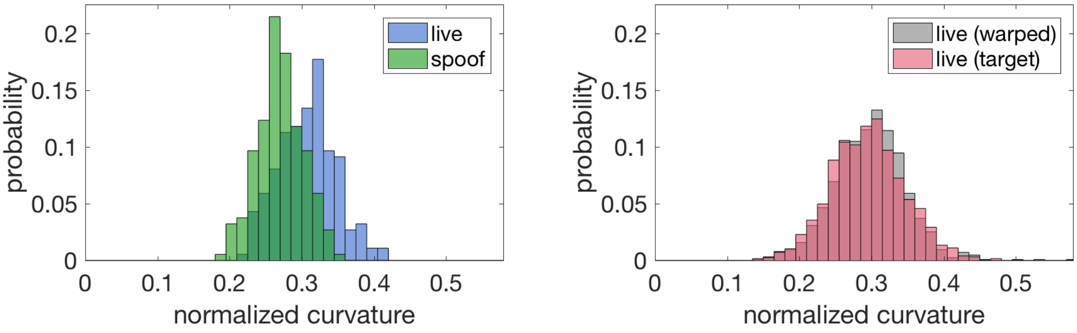


Figure 4.16: Histograms of the ridge skeleton curvature. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

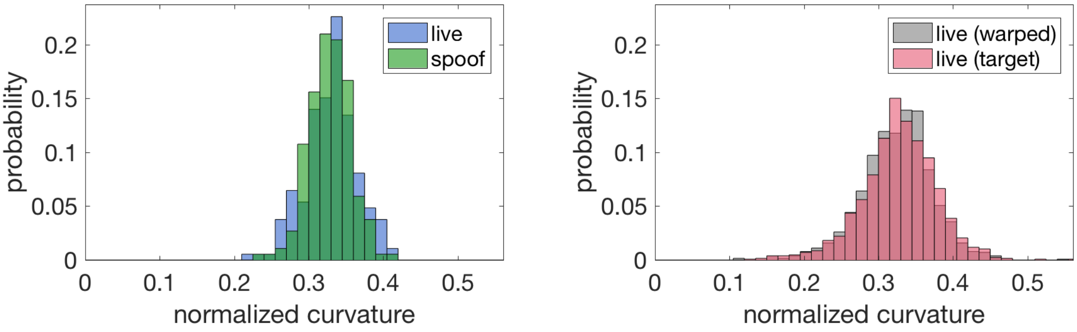


Figure 4.17: Histograms of the valley skeleton curvature. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

The curvature analysis was first performed without removing all the bifurcations and crossovers from the skeletons. The obtained result showed a significant increase in the curvature of the spoof ridge skeletons compared to the live ridge skeletons. This result was however believed to be strongly influenced by the large amount of bifurcations in the ridge skeletons since the skeletons bend sharply in these points. In order to minimize the correlation between the skeleton point detection and the skeleton curvature analysis, all crossovers and bifurcations and their 8-neighbors were removed from the skeletons before the curvature of the skeletons were evaluated. The results in Figure 4.16 and 4.17 are the results obtained after this correction. Interestingly, the obtained result still showed a significant difference in the curvature between live and spoof ridge skeletons, but the shift of the spoof distribution now implied a decrease in curvature in the spoof ridge skeletons compared to the live ridge skeletons. This result implies that the large amount of bifurcations and crossovers removed from the ridge spoof skeletons actually did influence the previous result strongly. A probable explanation for the significant decrease in curvature for the spoof ridge skeletons is that their additional skeleton bifurcations and crossovers often coincide with skeletons pixels in which the larger scale curvature is higher. If this is the case, the curvature analysis might however not measure the jaggedness of the skeletons as intended, but instead measure the larger scale curvature of the ridges and valleys. This explanation however still implies that the results are influenced by the amount of bifurcations in the spoof skeletons.

The difference in curvature between the live and spoof distributions in Figure 4.16 is significant at the $p = 0.01$ level. The correlation between the number of detected bifurcations and the skeleton curvature in the spoof ridge skeletons is -0.54. The influence from the amount of detected skeleton bifurcations is thus considered not to be large enough to exclude this feature from the final quality assessment model. Even though the curvature analysis probably did not manage to measure the jaggedness of the skeletons and instead is believed to measure the larger scale curvature of the ridges and valleys, this feature is still considered valuable in the quality assessment since the larger scale curvature of the images in each pair should be the same.

4.1.4 Intensity distributions

In the following sections, the results from the intensity distribution analyses and the intensity profile analysis are presented and discussed. These analyses were performed both on the set of image pairs containing one live and one spoof image and also on the reference image pairs. For each of the analyses presented below, the results for the two image sets are presented in the same figure but in separate histograms. Hence, each histogram figure contains two histograms and the left histogram visualizes the results obtained from the first set of image pairs, while the right histogram visualizes the results obtained from the second set of image pairs. In the left histograms, the distribution in green originates from the spoof images and the distributions in blue originates from the live images. Both distributions in the right histograms originate from live fingerprint images, the pink distributions originate from the live images used as targets in the registration process while the gray distributions originate from the warped images.

4.1.4.1 Intensity distribution in fingerprint ridges and valleys

The results from the intensity distribution analysis of the ridges and valleys in the fingerprint images are presented in Figure 4.18 and 4.19. As previously stated, the left histogram visualizes the results obtained from the image pairs containing both live and spoof fingerprint images, while the right histogram visualizes the results obtained from the image pairs containing two live fingerprint images.

As can be seen in Figure 4.18 and 4.19, no significant differences in the mean ridge and valley intensity were found. This analysis relies heavily on the separation of the ridges and valleys in the binarization process. But since the binarization of the images is based on global thresholds, the intensity variations in the ridges and valleys are naturally limited by the calculated thresholds. If the intensity in the ridges and valleys varies past this threshold the ridges and valleys are disrupted during the binarization. Thus, the intensity variations seen along the ridges and valleys in the spoof images are not captured by this analysis. In order to capture the intensity variations along the spoof fingerprint ridges and valleys, the ridges and valleys in the binary images need to be intact. There are complex image processing methods capable of repairing the fingerprint ridges and valleys, but an implementation of such methods was not feasible within the time frame of this thesis.

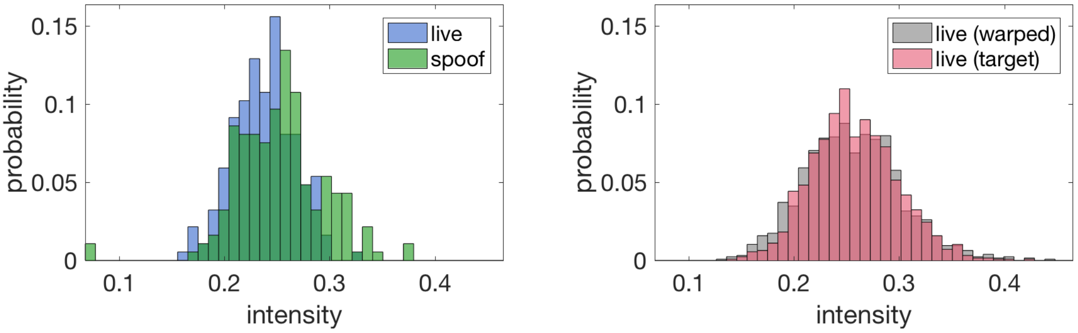


Figure 4.18: Histograms of the average ridge intensity. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

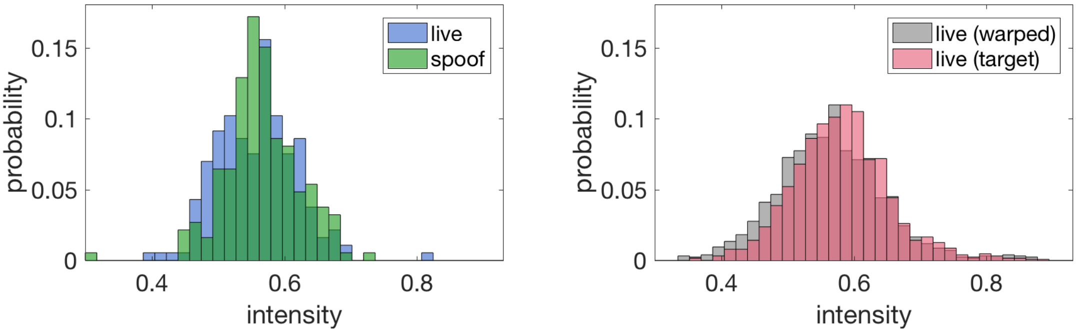


Figure 4.19: Histograms of the average valley intensity. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

4.1.4.2 Intensity distribution along fingerprint skeletons

The results from the intensity distribution analysis along the ridge skeletons and valley skeletons of the fingerprint images are presented in Figure 4.20 and 4.21. The results presented in the left histograms originate from the first set of image pairs, while the results in the right histograms originate from the second set of image pairs.

Just as in the analysis of the intensity distribution of the fingerprint ridges and valleys, the intensity distribution along the valley skeletons are naturally limited by the threshold used in the binarization process. However, by studying the histograms in Figure 4.20 it is clear that there actually is a significant difference in the mean intensity along the ridge skeletons. The difference in mean intensity along the ridge skeletons is significant at the $p = 0.01$ level. This difference is probably partially due to the fact that the ridges in the spoof images often are brighter than the ridges in the live images. However, since a similar difference is not seen in the intensity distribution in the entire spoof fingerprint ridges (see Figure 4.18), additional factors probably also affected the result. For example, the intensity distribution along the spoof ridge skeletons is most likely affected by the large amount of bridges in the spoof ridge skeletons since the intensity along the skeleton bridges is higher than the intensity along the ridges. The correlation between the spoof ridge skeleton intensity mean and the number of valley breaks and ridge bifurcations in the spoof skeletons is however low, 0.16 and 0.18 respectively. Thus, the intensity mean along the ridge skeletons is considered as a feature which can contribute to the assessment of spoof fingerprint images.

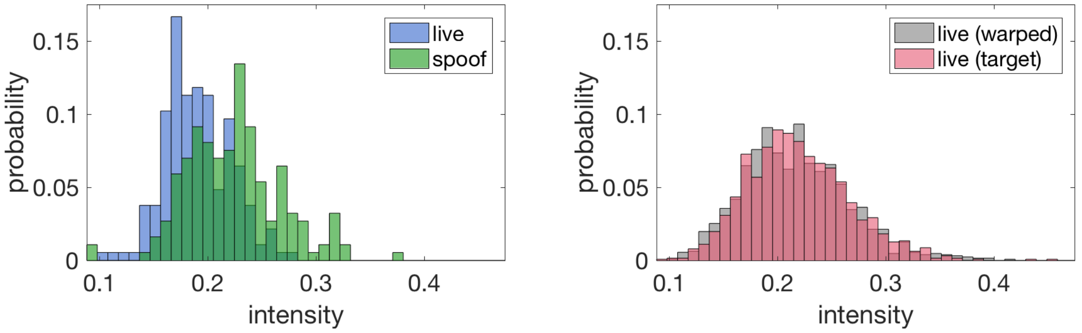


Figure 4.20: Histograms of the average intensity along the ridge skeletons. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

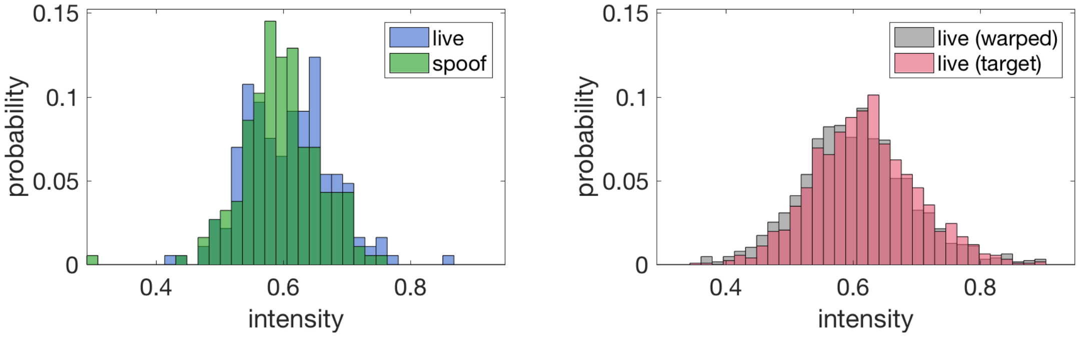


Figure 4.21: Histograms of the average intensity along the valley skeletons. The left histogram visualizes the result from the analysis of the image pairs containing both a live and a spoof image and the right histogram visualizes the result from the analysis of the reference image pairs which instead contained two live images.

4.1.4.3 Intensity profiles

The results from the intensity profile analysis are presented in Figure 4.22 and 4.23. The results presented in the left histograms originate from the first set of image pairs, while the right histograms originate from the second set of image pairs.

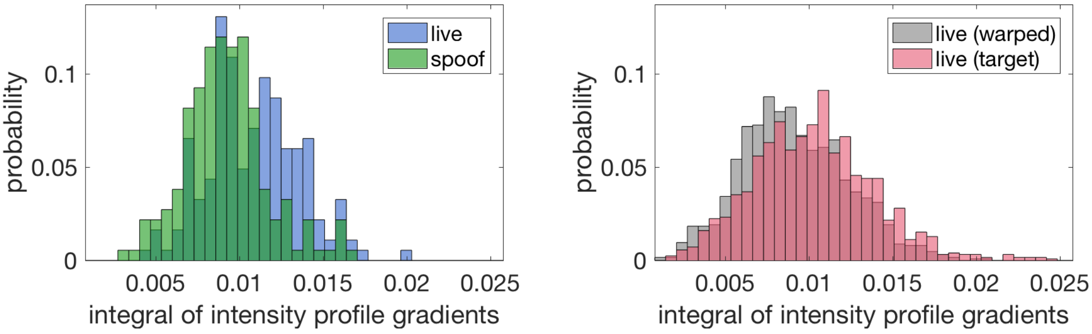


Figure 4.22: Histograms of the average steepness of the valley-ridge-valley intensity profiles. The left histogram visualizes the result from the analysis of the image pairs containing both live and spoof images and the right histogram visualizes the result from the analysis of the reference image pairs.

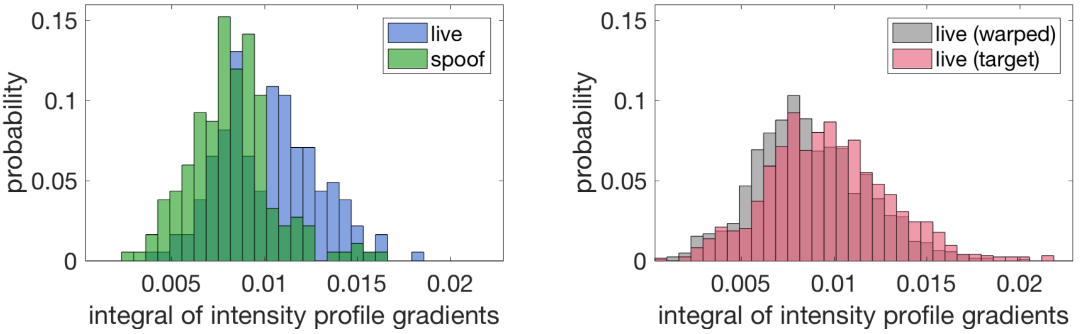


Figure 4.23: Histograms of the average steepness of the ridge-valley-ridge intensity profiles. The left histogram visualizes the result from the analysis of the image pairs containing both live and spoof images and the right histogram visualizes the result from the analysis of the reference image pairs.

The similar shift between the live and spoof distributions seen in Figure 4.22 and 4.23 is most expected since a steep valley-ridge-valley intensity profiles should imply that there is also a steep ridge-valley-ridge intensity profile. The correlation between the valley-ridge-valley intensity profiles and the ridge-valley-ridge intensity profiles is 0.97, hence there is a strong positive correlation between the two. Since the spoof fabrication process is believed to introduce a sharp shift between the ridges and valleys, the steepness of the intensity profile perpendicular to the ridge and valley pattern is expected to be larger in the spoof images than in the live images. However, the obtained results presented in Figure 4.22 and 4.23 suggest the opposite. From these figures, it is clear that the intensity profile is steeper in the live images than in the spoof images. The distribution differences between the live and spoof images seen in these figures are both significant at the $p = 0.01$ level. Since the obtained results were opposite to the expected, two-sided t-tests were used to evaluate the significance. However, this hypothesis was not formed from visual inspection but rather from knowledge about the spoof fabrication process. Actually, a steeper intensity profile in the spoof fingerprint images compared to the live fingerprint images could not be established during the visual inspection. The original hypothesis might thus be incorrect. Another possible explanation for this contradictory result could have been that the steepness of the intensity profile is affected by the slight differences in ridge and valley intensities often seen in the live and spoof images. Since the ridges often are slightly brighter in the spoof fingerprint images compared to the live fingerprints, the steepness of the intensity difference along the intensity profile might have become less steep in the fingerprint images. However, if the intensities values along the profile were normalized before the steepness was calculated, no difference was seen in the result. Thus, this explanation is not probable. Either way, the intensity profile steepness evidently detects some kind of significant difference in the image pairs and is thus considered to be a relevant feature to include in the quality assessment of the fingerprint images. Even though the valley-ridge-valley intensity profiles and the ridge-valley-ridge intensity profiles are highly correlated, both of them are included in the spoof quality assessment.

4.1.5 Ridge and valley width

In the following sections, the results from the analyses regarding the ridge and valley width are presented and discussed. These analyses are the distance transform analysis, the relative width analysis and the frequency spectrum analysis.

4.1.5.1 Distance transforms

The results from the distance transform analysis are presented in Figure 4.24 and 4.25. In both of these figures, the left histograms visualize the results from the analysis of the image pairs containing both live and spoof images while the right histograms visualize the results obtained when performing the same analysis on the reference data. The distributions in green in the left histograms shows the average distance to the closest ridge or valley in each spoof image and the distributions in blue in the left histograms shows the average distance to the closest ridge or valley in each live image. In the right histograms, both distributions are from live images, but the pink distributions originate from the live images used as targets in the registration process while the gray distributions originate from the warped images.

By studying the histograms in Figure 4.24 and 4.25, it is clear that there is a significant difference in the average distance to the closest valley between the live and spoof images. This difference is significant at the $p = 0.01$ level. However, if this difference were due to thinner valleys in the spoof images compared to the live images, an opposite shift of the spoof distribution would have been seen in Figure 4.24 since the ridges in the spoof images then conversely would be thicker. Hence, the large shift in distribution seen in Figure 4.25 is probably not caused by thinner valleys in the spoof images. The large shift is again probably caused by the large amount of valley breaks in the spoof images since the average distance to the closest valley in an image increases if the valley breaks are longer than the average ridge width. The correlation between the average distance to the closest valley and the number of valley breaks is 0.71, thus a combination of these two features in the quality assessment of the fingerprint images will not yield an additive effect in the evaluation even though both features are considered valuable. The smaller shifts in distribution seen in the right images in Figure 4.24 and 4.25 are most probably due to variations in the fingerprint images caused by differences in finger pressure against the sensor between the two sets.

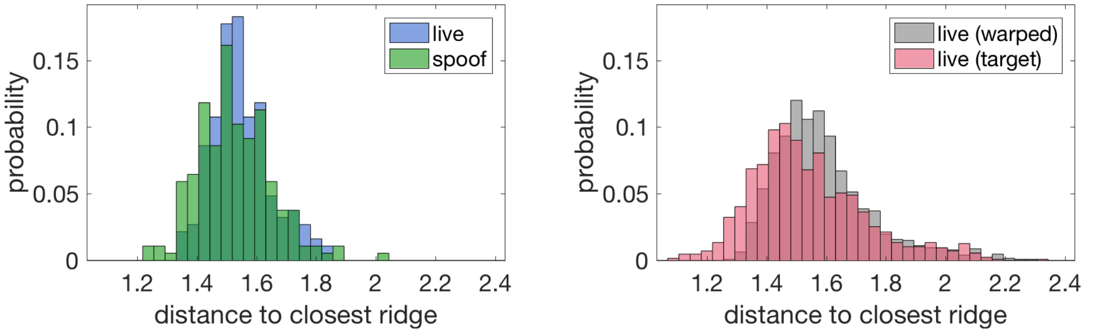


Figure 4.24: Histograms of the average distance from the valley pixels in an image to their closest ridge. The left histogram visualizes the result from the analysis of the image pairs containing both live and spoof images and the right histogram visualizes the result from the analysis of the reference image pairs.

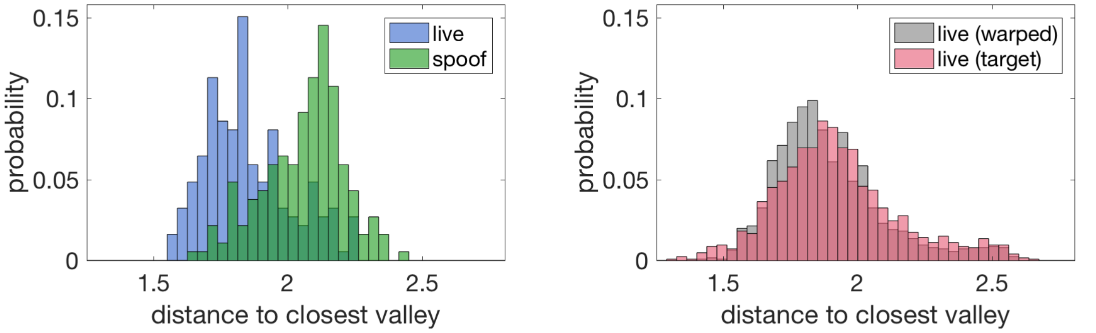


Figure 4.25: Histograms of the average distance from the ridge pixels in an image to their closest valley. The left histogram visualizes the result from the analysis of the image pairs containing both live and spoof images and the right histogram visualizes the result from the analysis of the reference image pairs.

4.1.5.2 Relative ridge and valley width

The results from the relative ridge and valley width analysis are presented in Figure 4.26. In the left histogram, the average relative ridge width for the image pairs is visualized and in the right histogram, the average relative valley width for the image pairs is visualized. The relative ridge width is defined as the ridge width estimated in the image used as target divided by the ridge width estimated in the warped image. The relative valley width is defined as the valley width estimated in the warped image divided with the valley width estimated in the image used as target. The distributions in green show the average relative ridge or valley width in the image pairs containing both live and spoof images while the distributions in blue shows the average relative ridge or valley width in the reference data. If there is a pairwise difference in relative width among the image pairs, the mean of the green distributions will be larger than one, while the mean of the reference data distributions showed in blue should be centered around one.

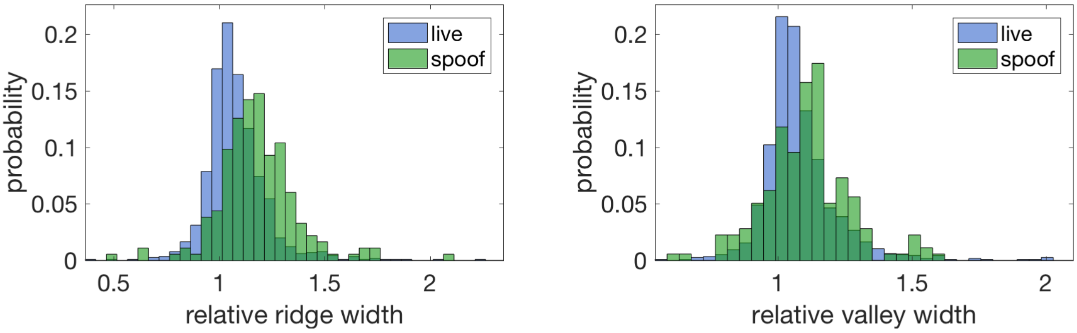


Figure 4.26: Relative ridge and valley width. The left histogram visualizes the relative ridge width in the image pairs and the right histogram visualizes the relative valley width in the image pairs. The green distributions shows the average ridge and valley width ratio in the image pairs containing one live image and one spoof image. The distributions in blue shows the average ridge and valley width ratio in the image pairs containing two live images.

As can be seen in Figure 4.26, there are small differences in relative width between the image pairs containing both live and spoof images and the reference image pairs. The difference in relative ridge width is significant at the $p = 0.01$ level and the difference in relative valley width is significant at the $p = 0.025$ level. Even though the obtained differences are significant, there are a few limitations of this method which are believed to have affected the results. First, the resolution of the investigated fingerprint images is believed to have limited the accuracy of the results obtained in this analysis. A higher image resolution would have resulted in more precise measurements of the ridge and valley widths. Second, noise in the fingerprint pattern such as spikes, holes and incipient ridges might affect the individual measurements considerably. Third, the irregular ridge and valley interfaces seen in the spoof fingerprint images is also believed to limit the accuracy of this analysis. Hence, the reliability of the final statistics might be impaired by these limitations. The relative ridge and valley width was thus not included in the quality assessment model.

Interestingly, it may also be noticed that the mean of the blue distributions originating from the reference image pairs in Figure 4.26 are not centered around one as expected. This result is however in agreement with the small distribution shifts seen in the right images of Figure 4.24 and 4.25 and is most probably due to variations in finger pressure against the sensor between the two sets.

4.1.5.3 Frequency spectrum analysis

In Figure 4.27, the average of the radial frequency signals of the source and target images in the image pairs are visualized. The plot on the left shows the results from the images in the pairs containing one live and one spoof image and the plot on the right shows the results from the reference image pairs. In Figure 4.28, the frequency spectra magnitudes for an example image pair containing a live and a spoof image are displayed and in Figure 4.29 their radial frequency signals are presented. The left image in Figure 4.28 visualizes the frequency spectrum of the spoof image in the pair and the right image visualizes the frequency spectrum of the live image.

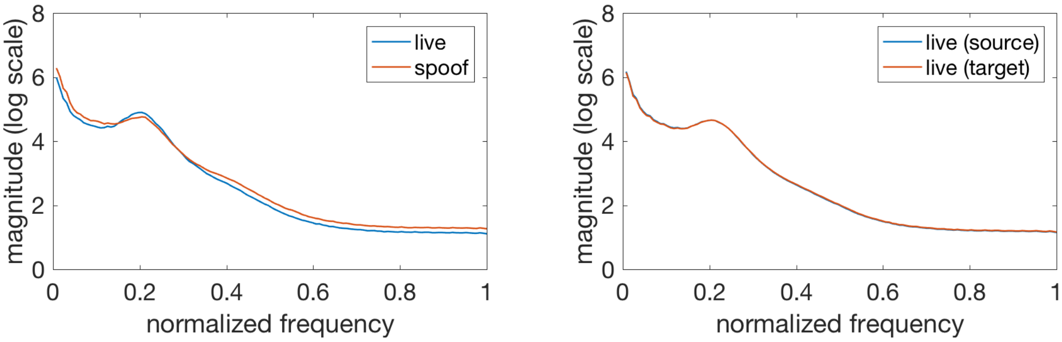


Figure 4.27: Average radial frequency. The left plot visualizes the average radial frequency of the live and spoof images in the data. The right plot visualizes the average radial frequency of the reference data.

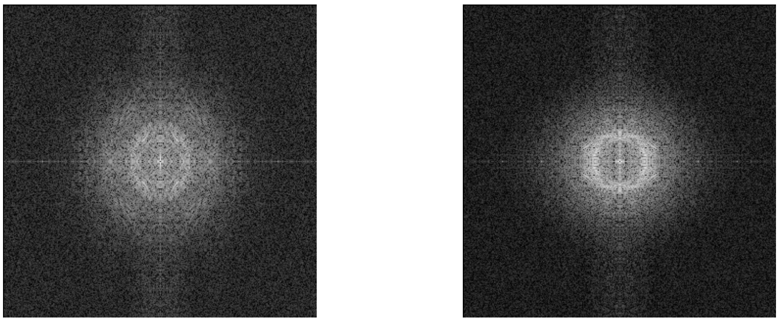


Figure 4.28: Frequency spectra magnitude examples. The left image is a typical example of the frequency spectrum magnitude of a spoof image and the right image is a typical example of the frequency spectrum magnitude of a live image. The magnitude is presented in logarithmic scale in both images.

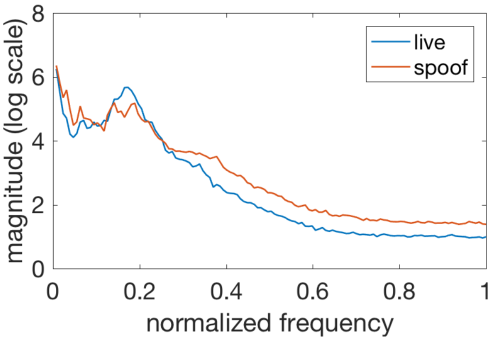


Figure 4.29: Radial frequency example. The plot visualizes an example of the difference in radial average frequency often seen between live and spoof images.

As can be seen in Figure 4.27, there is a difference between the average radial frequency magnitude for the all the live images and the average radial frequency magnitude for the all the spoof images, compare to the right plot in the same figure in which there is a perfect overlap between the average radial frequency magnitude signals for the source and target images. The difference between the live and spoof frequency spectra magnitudes becomes even more apparent when studying each pair separately. The magnitudes of the spoof frequency spectra are in general blurrier than their live equivalents. In addition to the blurriness, the lobe corresponding to the ridge and valley frequencies are more concentrated in the frequency spectra of the live fingerprint images than in the frequency spectra of the spoof images. These trends can clearly be seen in the example image pair in Figure 4.28. Figure 4.29 shows that the difference in the radial frequency obtained from these frequency spectra magnitudes also is more evident than the average of the radial frequency signals for all the image pairs. Due to time constraint, this feature was however not included in the spoof fingerprint image quality assessment.

4.1.6 Spoof image quality

The predictors used in the hyperplane model and their respective weights are presented in Table 4.1. From the magnitude of the weights presented in this table, it is clear that the intensity profiles are the predictors which have the largest impact in the hyperplane model. The number of breaks in the valley skeleton and the number of bifurcations in the ridge skeleton are also given a large impact in the model.

Table 4.1: Hyperplane model predictors and their respective weights. In this table the ten features used as predictors in the separating hyperplane model are presented together with their respective weights. The bias term in the model is -0.1086.

hyperplane model predictors	$\boldsymbol{\beta}$
ridge-valley-ridge intensity profile	1.9123
valley-ridge-valley intensity profile	-1.5150
number of bifurcations in the ridge skeleton	-0.6933
number of breaks in the valley skeleton	-0.6806
ridge skeleton intensity	-0.4656
number of spikes in the ridge skeleton	-0.3147
distance to closest valley	0.2613
number of bifurcations in the valley skeleton	0.1668
number of spikes in the valley skeleton	-0.1631
ridge skeleton curvature	0.1402

In the 186 image pairs containing both live and spoof images, 162 of the live images and 172 of the spoof images are unique. When these unique images are classified with the separating hyperplane model, 66% of the live images and 86% of the spoof images are classified correctly. Hence, the separating hyperplane model classifies 76% of the images in the investigated set correctly. The probability outputs from the separating hyperplane model are used as an indication of the spoof image quality. Low quality images are identified by extracting the spoof images with a high spoof probability while high quality images are identified by extracting the spoof images with a low spoof probability. In Figure 4.30 and 4.31, examples of the spoof images which obtained the highest and lowest spoof probability are presented together with their live equivalents. The spoof probability of the low quality spoofs displayed in

Figure 4.30 are 0.99 while the spoof probability of the high quality spoofs displayed in Figure 4.31 range between 0.06 and 0.16.

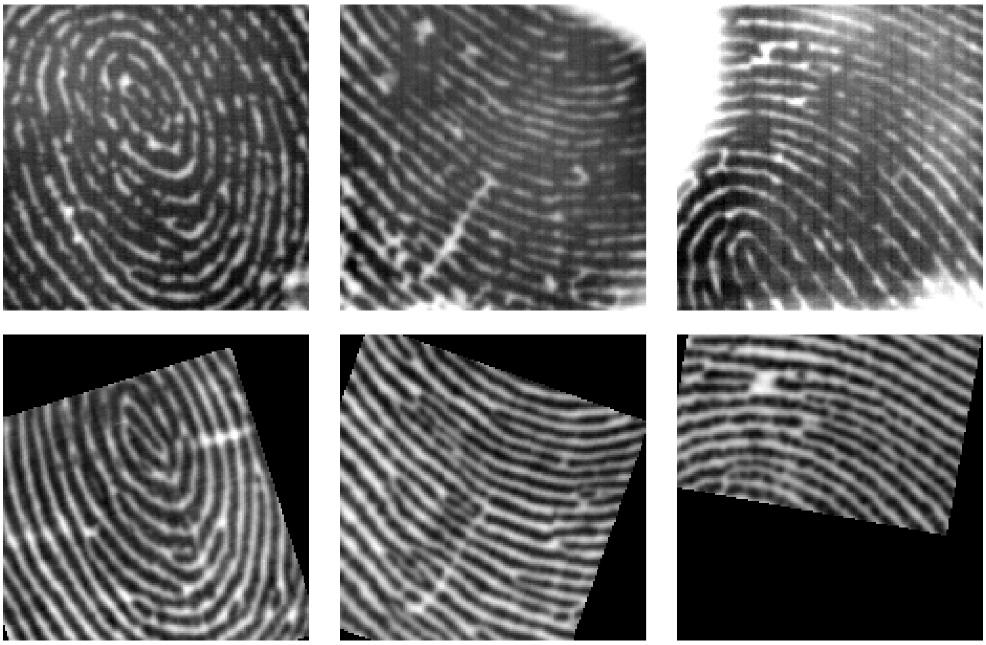


Figure 4.30: Examples of low quality spoof images. In the top row of this figure three of the spoof images with the lowest quality are presented. In the lower row their live equivalents are displayed.

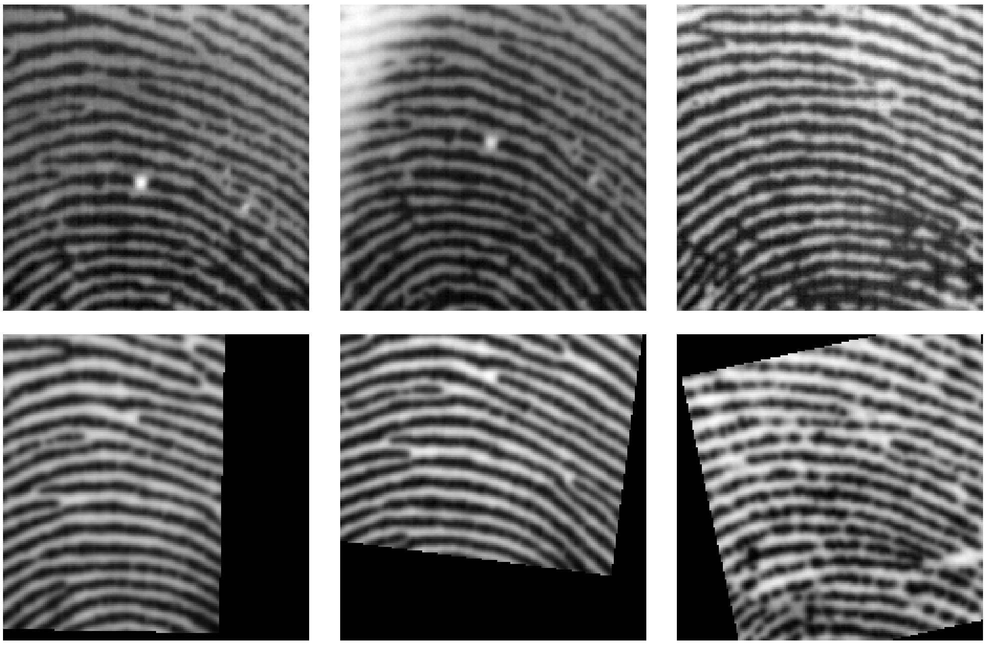


Figure 4.31: Examples of high quality spoof images. In the top row of this figure three of the spoof images with the highest quality are presented. In the lower row their live equivalents are displayed.

The spoof image examples presented in Figure 4.30 and 4.31 clearly demonstrates the success of the quality assessment. The spoof images in Figure 4.30 are typical examples of spoof images which are considered to be easy to differentiate from live images, while the images in Figure 4.31 are typical examples of spoofs images in which the ridge and valley pattern cannot be differentiated from live ridge and valley patterns. Hence, the spoof image quality can successfully be estimated from the features designed in this thesis. Important to mention is that the low quality images presented in Figure 4.30 are not the worst images in the entire data set, since such images did not pass the matching algorithm and thus did not end up in any of the analyzed image pairs. The quality of the spoof images in Figure 4.30 is thus low compared to the spoof images in the 186 image pairs, but is still high enough to pass the used matching algorithm. During a manual inspection of the spoof images in Figure 4.31, the circular shapes with high intensity in the left and middle spoof image should, however, raise a warning flag. Such circular shapes may originate from dirt on the sensor or from a blister on a live finger, but do often originate from an air bubble in the spoof material. However, since a realistic ridge and valley pattern is considered to be the most important feature to assess when estimating the spoof image quality, the quality of the spoof images in Figure 4.31 are still considered as high even though they possess a fairly discriminative feature.

4.2 Learning-based fingerprint liveness classification

In the following sections, the results from the learning-based liveness detection part of the thesis are presented and discussed. Initially, the purpose of the neural network part of the thesis was to implement a state-of-the-art liveness detection method for which the performance on different data sets could be used as an indication of their spoof quality. But since the characteristics of fingerprint images are heavily dependent on sensor technology, the first network, which was trained on images obtained with optical fingerprint sensors, did not manage to generalize to the images obtained with the capacitive fingerprint sensing technology used at Fingerprint Cards. The first network could thus not be used to estimate the quality of the spoof images provided by Fingerprint Cards and a second network had to be trained on capacitive fingerprint images. Hence, the results obtained from the first network is instead used as an assurance that the network is successfully implemented.

4.2.1 Network trained on benchmark data

The classification results for each of the test sets used in the Fingerprint Liveness Detection Competition in 2009, 2011 and 2013 are given in Table 4.2. The spoof false positive rate, the spoof false negative rate and the average classification errors obtained for the test sets after 30 epochs are given for each of the data sets. The average classification errors previously obtained for the same testing data sets using a network which had the exact same architecture and which had been trained on the same data sets are presented within brackets [6] in order to facilitate a comparison between the state-of-the-art results and the results obtained in this thesis.

By studying the average classification errors in Table 4.2, it is clear that the obtained results are not quite as impressive as the results presented in [6]. The obtained results are however good enough to be sure that the network has been implemented successfully. There are many possible reasons to why two networks of a given architecture do not obtain the same performance, even though both are trained and evaluated on the same data sets. First, the data augmentation proposed by [6] is reported to improve their average ACE from 4.2 to 2.9. Hence, the higher average ACE compared to the previously reported results is probably due to the lack of data augmentation in the implementation in this thesis. Other factors which affect the results and that were not specified in [6] are how the training sets were divided into training and validation sets, how the grayscale images in the data sets were converted to three-channel images, what loss function to use for training and the number of epochs the training was run.

Table 4.2: Classification results of the test data sets created for the Liveness Detection Competition in the years of 2009, 2011 and 2013. The obtained spoof false positive rate, spoof false negative rate and average classification rate for each of the test data sets are presented. The average classification rates for each data sets previously obtained with the same network architecture [6] are presented in brackets. The average spoof false positive rate, spoof false negative rate and average classification rate for all the test data sets are presented in the bottom of the table.

data sets		**classification errors**		
year	**sensor**	**SFPR**	**SFNR**	**ACE**
2013	biometrika	19.9	1.8	10.8 (1.8)
	crossmatch	2.2	4.4	3.3 (3.4)
	italdata	7.7	1.4	4.5 (0.4)
	swipe	5.5	5.7	5.6 (3.7)
2011	biometrika	1.2	14.1	7.6 (5.2)
	digital	3.1	5.5	4.3 (3.2)
	italdata	0.6	9.2	4.9 (8.0)
	sagem	2.7	5.3	4.0 (1.7)
2009	biometrika	19.9	2.2	11.1 (4.1)
	crossmatch	1.6	3.4	2.5 (0.6)
	identix	0.9	1.2	1.0 (0.2)
average		**5.9**	**4.9**	**5.4 (2.9)**

4.2.2 Network trained on Fingerprint Cards data

In Figure 4.32, the network learning progress is reported with help of the classification error and the network objective (i.e. loss function) versus the number of completed epochs. Already after five epochs of training, the network classified all of the 334 test images correctly. Recall that the test set contained 162 live fingerprint images and 172 spoof images. Both the training and validation classification error continued to decrease and thus the network obtained after eleven epochs is chosen as the final network. After having completed eleven epochs of training, there were only three spoof images in the test set for which the assigned spoof probability was below 0.975. These images are presented in Figure 4.33. The spoof classification scores of the left, middle and right image are 0.67, 0.87 and 0.97 respectively.

By studying the network learning process in Figure 4.32, it is clear that both the classification error and the network objective decreases rapidly during the first few epochs. While the classification accuracy of the test images reaches 100% after five epochs, the classification accuracy of the training and validation images is only marginally increased from their classification accuracies of 99.95% and 99.50% which are reached after eleven epochs.

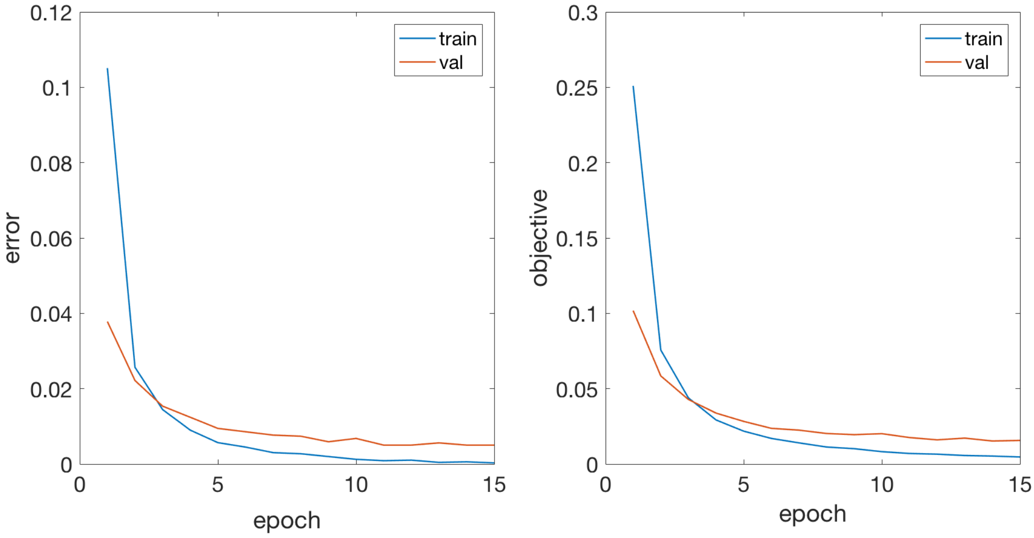


Figure 4.32: Classification error and objective during network training. The left plot displays the learning progress with help of the classification error both for the training and validation set and the right image displays the learning progress with help of the network objective for both for the training and validation set.

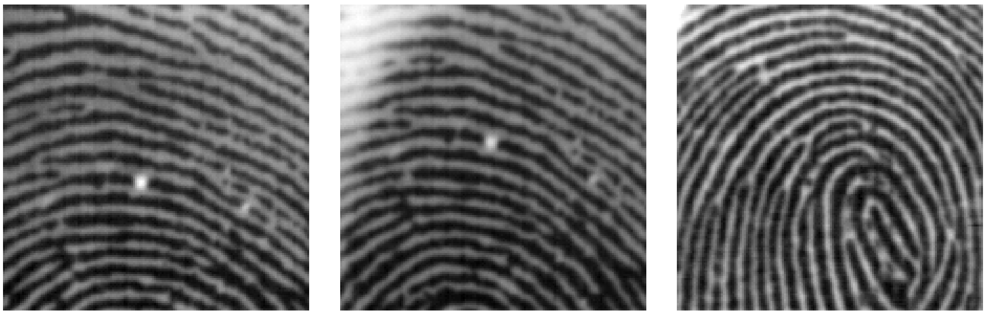


Figure 4.33: The three best spoofs according to the neural network. These three spoof images yielded the lowest spoof scores. Their spoof scores are however higher than their liveness scores, hence they are still classified as spoof images.

The fine-tuned network performed far better than expected and classified all the images in the test set correctly. These results imply that all the images are possible to classify correctly, regardless of the spoof image quality in the set. However, since the network managed to classify even the most realistic spoof images correctly with a high degree of certainty, this network architecture is not suitable to use in a spoof image quality assessment. Differentiation between the quality of the images in the set could however possibly be obtained with a more shallow network.

The images presented in Figure 4.33 clearly demonstrates that the spoof images which the network is the least certain of are the ones with ridge and valley patterns which cannot be differentiated from ridge and valley patterns of live fingerprints. Thus, the network appears to have learned to identify fingerprint images with realistic ridge and valley patterns. Moreover, the network has evidently learned to differentiate obvious spoof images from live fingerprint images.

Important to note is that the left and the middle images in Figure 4.33, which are the same images that were the most difficult for the network to classify as spoofs, are the images which were assigned the highest quality in the feature design part of the thesis (see Figure 4.31). As previously discussed in Section 4.1.6, the bright circular shapes in these images probably originate from air bubbles in the spoofs. Since the images in the training set did not contain many spoof images with air bubbles, this is evidently not a feature that the network learned during training. This is, however, a feature which the network probably could learn if the training set contained a larger amount of spoof images with this particular feature.

The low quality spoof example images in Figure 4.30, which were assigned the highest spoof probabilities by the hyperplane model, were also assigned very high spoof probabilities by the network model. These images were assigned spoof probabilities in the range $0.99 - 1.0$ by the network. Hence, these three images are easily differentiated from live fingerprint images both by the hyperplane model and the convolutional neural network. The network also assigned a very high spoof probability (0.99) to the rightmost of the high quality spoof images in Figure 4.31. The fact that the network model assigned almost the same spoof probability to these four spoof images substantiates the claim that the structure of the given network is unsuitable for spoof image quality assessment. The hyperplane model assigned a spoof probability of 0.38 to the rightmost of the images in Figure 4.33. This is considered to be a quite high spoof probability for such a realistic spoof image. A possible explanation for this unexpectedly high spoof probability could be the intensity variations along the valleys which may be seen in some areas of the image.

The difference in performance of the two trained networks in this thesis is believed to exist due to the difference in variability within their training and test data sets. The training and test images used in the first network are obtained with eight different sensors and the spoof images in the sets are created with spoofs of eight different materials, while the training and test images used in the second network are obtained with a single sensor and the spoofs are all made of the same material.

5

Conclusion

The purpose of this thesis was to develop a tool which assesses the quality of spoof fingerprint images in a data set and two methods has been evaluated for this purpose. The first method was based on manually designed features derived from visual comparisons of live and spoof images in the data set provided by Fingerprint Cards. Ten of these detected features were found to differ significantly between the live and spoof images and these features were thus used to create a support vector machine-based classifier by identifying the hyperplane model which best separated the live and spoof fingerprints in the data set. The quality of the spoof images in a data set was estimated by assessing the number of spoof images that this hyperplane model managed to classify as spoofs. Moreover, the quality of the individual spoof images in the set was defined as the liveness probability assigned by the model. Promising results were obtained from this quality assessment method. The spoof images that were assigned a low quality by the hyperplane model were images which easily could be differentiated from their live equivalents in a manual inspection. Conversely, the spoof images that were assigned a high quality were images in which the fingerprint patterns could not be differentiated from live fingerprint patterns. The presented results indicate that the implemented quality assessment was successful for the investigated data set. Further, it shows that the liveness features which were manually designed in this thesis can be used to estimate the spoof image quality.

Features for which there were significant differences between the investigated live and spoof images originated from the fingerprint skeletons, the fingerprint intensities and the ridge and valley patterns of the fingerprint images. The number of breaks in the valley skeletons and the number of spikes and bifurcations in both the ridge and valley skeletons were the first five significant liveness features that were constructed. All these features occurred more frequently in spoof fingerprint skeletons than in their live equivalents. Another significant difference was found when the curvature of the ridge skeleton was investigated. The curvature of the skeleton is however not believed to measure the skeleton jaggedness as intended, but is instead believed to measure the larger scale curvature of the skeleton. Although this assessment might not function as intended, the curvature of the ridge skeleton is still considered to be valuable features. The significant features related to the fingerprint intensities were the average intensity along the ridge skeletons and the intensity profiles perpendicular to the ridge and valley patterns in the images. Even though the results obtained for the intensity profile features did not agree with the original hypothesis, the differences found was still considered to be a valuable feature. The significant feature related to the ridge and valley patterns is the average distance to the closest valley from all the ridge pixels in a fingerprint image. Among these ten features, the intensity profiles were given the largest impact in the separating hyperplane model which best separated the two classes. The number of breaks in the valley skeletons and the number of bifurcations in the ridge skeletons were also given a relatively large impact in the separating hyperplane model.

The second method evaluated for spoof image quality assessment purposes was based on a convolutional neural network from which state-of-the-art liveness classification results recently has been reported on benchmark data sets. The initial idea was to estimate the spoof image quality of a data set based on the classification results obtained when the set was classified with the state-of-the-art liveness classification network. This network was however trained on images obtained from optical fingerprint sensors and did not manage to generalize to fingerprint images obtained from capacitive sensors due to their different characteristics. By training the same network architecture on fingerprint images obtained from capacitive sensors instead of images obtained from optical sensors, a capacitive version of the state-of-the-art liveness detection network was created. The performance of this liveness detection network was far better than expected, the network classified all the live and spoof images in the investigated set correctly and only three of the spoof images were assigned a spoof probability lower than 0.975. These are fantastic liveness detection results. However, the fact that the network managed to classify even the most realistic spoof images correctly with a high degree of certainty makes this network architecture unsuitable for spoof quality assessment. Differentiation between the images in the set could however possibly be obtained with a more shallow network.

Since the implemented liveness detection network demonstrated such extraordinary performance, it is suggested that future studies investigate the network in more detail. There are methods to investigate the features learned by the network [33, 34] and by identifying such features a better understanding of the liveness detection performed by the network might be obtained. Also, by investigating the network in more detail it could probably be reduced in size which could make it suitable for usage in on-line applications such as mobile phones.

Bibliography

[1] A. K. Jain *et al.*, *Introduction to Biometrics*, pp. 1–10, 44–47, 51–81. Springer, 2014.

[2] D. Maltoni *et al.*, *Handbook of Fingerprint Recognition*, ch. 1, 2, 3, 4, 5, 8, 9. Springer, 2009.

[3] S. Marcel *et al.*, eds., *Handbook of Biometric Anti-Spoofing - Trusted Biometrics under Spoofing Attacks*, pp. 1–10, 35–43. Advances in Computer Vision and Pattern Recognition, Springer, 2014.

[4] Fingerprint Cards AB. `https://www.fingerprints.com` [2016-10-17].

[5] L. Ghiani *et al.*, "Review of the Fingerprint Liveness Detection (LivDet) competition series: 2009 to 2015," *Image and Vision Computing*, 2016.

[6] R. F. Nogueira *et al.*, "Fingerprint liveness detection using convolutional neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1206–1213, 2016.

[7] D. Gragnaniello *et al.*, "Local contrast phase descriptor for fingerprint liveness detection," *Pattern Recognition*, vol. 48, no. 4, pp. 1050–1058, 2015.

[8] D. Gragnaniello *et al.*, "An investigation of local descriptors for biometric spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 849–863, 2015.

[9] L. Ghiani *et al.*, "Fingerprint liveness detection using binarized statistical image features," in *IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2013.

[10] J. Galbally *et al.*, "A high performance fingerprint liveness detection method based on quality related features," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 311–321, 2012.

[11] E. Lim *et al.*, "Fingerprint quality and validity analysis," in *IEEE International Conference on Image Processing*, vol. 1, pp. 469–472, 2002.

[12] Y. Chen *et al.*, "Fingerprint quality indices for predicting authentication performance," in *International Conference on Audio- and Video-Based Biometric Person Authentication*, pp. 160–170, Springer, 2005.

[13] P. Coli *et al.*, "Fingerprint silicon replicas: static and dynamic features for vitality detection using an optical capture device," *International Journal of Image and Graphics*, vol. 8, no. 4, pp. 495–512, 2008.

[14] T. P. Chen *et al.*, "Fingerprint image quality analysis," in *IEEE International Conference on Image Processing*, vol. 2, pp. 1253–1256, 2004.

[15] L. Hong *et al.*, "Fingerprint image enhancement: algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, 1998.

[16] M. Kücken and A. C. Newell, "Fingerprint formation," *Journal of Theoretical Biology*, vol. 235, no. 1, pp. 71–83, 2005.

[17] R. Cappelli and D. Maltoni, “On the spatial distribution of fingerprint singularities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 742–748, 2008.

[18] A. M. Bazen and S. H. Gerez, “Segmentation of fingerprint images,” in *ProRISC Workshop on Circuits Systems and Signal Processing*, pp. 276–280, 2001.

[19] A. M. Bazen and S. H. Gerez, “Directional field computation for fingerprints based on the principal component analysis of local gradients,” in *ProRISC Workshop on Circuits Systems and Signal Processing*, pp. 215–222, 2000.

[20] R. C. Gonzalez and R. E. Woods, *Digital image processing*, pp. 649–675, 764–765, 834–837. Pearson Prentice Hall, 2010.

[21] P. Soille, *Morphological image analysis: principles and applications*, pp. 1–10. Springer, 2013.

[22] I. Goodfellow *et al.*, *Deep Learning*, pp. 1–26, 330–372. Adaptive Computation and Machine Learning series, The MIT Press, 2016.

[23] M. A. Nielsen, *Neural Networks and Deep Learning*, ch. 1, 2, 6. Adaptive Computation and Machine Learning series, Determination Press, 2016.

[24] Y. LeCun *et al.*, “Convolutional networks and applications in vision,” in *International Symposium on Circuits and Systems*, pp. 253–256, 2010.

[25] A. Vedaldi and K. Lenc, “MatConvNet – Convolutional Neural Networks for MATLAB,” in *ACM International Conference on Multimedia*, 2015.

[26] R. A. Adams and C. Essex, *Calculus: a complete course*, pp. 867–873. Pearson Canada, 8 ed., 2013.

[27] M. Sonka *et al.*, *Image Processing, Analysis, and Machine Vision*, pp. 337–339. Cengage Learning, 2014.

[28] V. Mura *et al.*, “LivDet 2015 – Fingerprint Liveness Detection Competition 2015,” in *IEEE International Conference on Biometrics Theory, Applications and Systems*, 2015.

[29] G. L. Marcialis *et al.*, “First International Fingerprint Liveness Detection Competition – LivDet 2009,” in *International Conference on Image Analysis and Processing*, pp. 12–23, 2009.

[30] D. Yambay *et al.*, “LivDet 2011 – Fingerprint Liveness Detection Competition 2011,” in *International Conference on Biometrics (ICB)*, pp. 208–215, 2012.

[31] L. Ghiani *et al.*, “LivDet 2013 – Fingerprint Liveness Detection Competition 2013,” in *International Conference on Biometrics (ICB)*, pp. 1–6, 2013.

[32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.

[33] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European Conference on Computer Vision*, pp. 818–833, 2014.

[34] R. Girshick *et al.*, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.