VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

BIOMETRIC LIVENESS DETECTION FOR THE FINGERPRINT RECOGNITION TECHNOLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. LUKÁŠ BRABEC

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

BIOMETRICKÁ DETEKCE ŽIVOSTI PRO TECHNOLOGII ROZPOZNÁVÁNÍ OTISKŮ PRSTŮ

BIOMETRIC LIVENESS DETECTION FOR THE FINGERPRINT RECOGNITION TECHNOLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. LUKÁŠ BRABEC
AUTHOR

VEDOUCÍ PRÁCE doc. Ing. MARTIN DRAHANSKÝ, Ph.D.
SUPERVISOR

BRNO 2015

Abstrakt

Tato práce je zaměřena na detekci živosti pro technologii rozpoznávání otisků prstů. V první části této práce je popsána biometrie, biometrické systémy, rozpoznávání živosti a je navržena metoda pro detekci živosti, která je založena na spektroskopických vlastnostech lidské kůže. Druhá část práce popisuje a shrnuje výsledky experimentů po implementaci této metody, v závěru práce jsou výsledky diskutovány a je nastíněna další možná práce.

Abstract

This work focuses on liveness detection for the fingerprint recognition technology. The first part of this thesis describes biometrics, biometric systems, liveness detection and the method for liveness detection is proposed, which is based on spectroscopic characteristics of human skin. The second part describes and summarizes performed experiments. In the end, the results are discussed and further improvements are proposed.

Klíčová slova

detekce živosti, otisk prstu, biometrie, optické vlastnosti, vlnová délka

Keywords

liveness detection, fingerprint, biometrics, optical properties, wavelength

Citace

Lukáš Brabec: Biometric Liveness Detection for the Fingerprint Recognition Technology, diplomová práce, Brno, FIT VUT v Brně, 2015

Biometric Liveness Detection for the Fingerprint Recognition Technology

Prohlášení

Hereby I declare that I have written this master's thesis on my own under the supervision of doc. Ing. Martin Drahanský, Ph.D. and I have acknowledged all sources I have used while writing this thesis.

. . . .
Lukáš Brabec
May 27, 2015

Poděkování

I would like to thank doc. Ing. Martin Drahanský, Ph.D., for supervising this thesis and guiding my work.

Contents

Chapter 1

Introduction

Importance of biometrics as method of authentication and identification grew during 20^{th} century and in 2013 it became de facto mainstream in daily life when the first phones with fingerprint sensor were introduced to the market [1].

The rapid growth of popularity, aside from forensic science and surveillance, where biometrics is used to identify criminals, is mostly due to the comfort that the biometrics brings to the daily life – PINs, passwords and keys are no more needed since one can use a finger or eyes for access control. However, the most used biometric feature – fingerprint is given, cannot be changed and is easily stolen and reproduced from latent impressions on glossy surfaces. This fact means that fingerprints (and other biometric features) are more usernames than passwords [2]. Security of fingerprint authentication systems can be improved with liveness detection – determining whether provided sample is alive (i.e. real or genuine) or attacker is trying to deceive the system with a fake finger.

This work deals with liveness detection for fingerprint recognition technology, at first biometrics in general is introduced, then methods of liveness detection are described and finally a method for liveness detection using optical properties is proposed.

The key parts, such as image enhancement, dataset creation, feature extraction and machine learning algorithms are further described, implemented and used to create liveness detection system. Finally performance of proposed method is discussed along with proposal for further improvements.

Chapter 2

Biometrics

In this work, biometrics refers to automatic method of identification or verification based on recognition of unique biological features of an individual. Biometrics can be used in computer science as identification and access control, to distinguish one individual among others that are under surveillance or to help classical forensic science.

First indirect usage of biometrics dates back to the 14^{th} century in China where potters and artists used fingerprint as a signature. First usage of biometrics for identification was in the 19^{th} century with rise of dactyloscopy. During the first half of 20^{th} century importance of fingerprints grew and in the second half of century, articles introducing usage of other biometric features (e.g. face) were published [3].

2.1 Identity, Identification, Verification, Authentication

Identity is unique set of features that identifies one individual among the others. Every individual has one physical identity (e.g. fingerprints or DNA) and would have many electronic identities (e.g. email) [4, 3]. Identity can be stolen.

Identification is a process of recognition of an individual among the others. In means of biometric systems, one provides a biometric characteristic that is compared with template from database and if match is found the individual is identified. For large databases, this process is time consuming.

Verification is a process that unlike identification compares biometric data only with one record from database. Electronic identity is provided and record template from database is chosen. Output of this process is binary – yes or no.

Authentication means confirming credibility of a person. Authentication can be one to many – using identification or with provided information (e.g. name) – using verification.

2.2 Biometric System

Biometric system typically consists of two modules (see Figure 2.1) that often comes together in one software package [5, 3]:

Enrollment module obtains biometric information, extracts features (minutiae) and then saves them into template database.

Identification or verification module works as enrollment module, but instead of saving, it retrieves data from database and compares them with the obtained features.

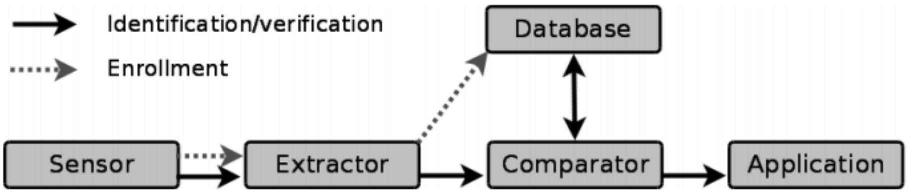


Figure 2.1: Identification/verification and enrollment in biometric system [5].

Attack on a biometric system can be targeted at several weak points. Some of these vulnerabilities are [5]:

biometric sensor can be provided with false biometric information, such as a fake finger,

output from sensor can be captured from previous communication and later replayed,

feature extraction can be modified and its output can be swapped with synthetic one,

database can be compromised and stored templates can be changed,

output from database can be blocked resulting in denial of service,

feature comparing and its output can be modified; and

output from biometric system to the application can be swapped with fake one.

2.3 Performance of Biometric Systems

Biometric system has to work with inaccuracy due to imperfect obtaining of biometric information or feature extraction. Similarity score is computed from comparison of provided biometric data and data from database. This score is then compared with a threshold, lower values are rejected and higher accepted [3]. Evidently, the threshold has big influence on the performance. For persons P and Q we distinguish four outcomes:

true acceptance – P is accepted as P,

true rejection – P is rejected as Q,

false acceptance – P is accepted as Q,

false rejection – P is rejected as P.

Setting the threshold is trade-off between comfort and security. High threshold will result in high security but low comfort due to high number of false rejections. On the other hand low threshold will result in high comfort and low security due to false acceptances. Beforehand mentioned outcomes are the basis for determining following rates [4]:

FTA (failure to acquire rate) is probability that describes inability to acquire biometric characteristic although the characteristic is present. This rate is sometimes called FTC – failure to capture rate. This number evaluates the quality of sensor and its suitability for given use case.

$$FTA = \frac{\text{Number of failed acquisition attempts}}{\text{Total number of capturing}} \tag{2.1}$$

FTE (failure to enroll rate) is fraction of users, from which was the biometric signal acquisition successful, but the system is unable to register them. This rate is sometimes called FTX – failure to extract rate. Higher rates are often casued by bad quality of samples.

$$FTA = \frac{\text{Number of failed registration attempts}}{\text{Total number of registration attempts}} \tag{2.2}$$

FTM (failure to match rate) is fraction of successful acquired biometric characteristics that, after registration, couldn't be used for template matching. This rate reflects the ability of system to make a match decision.

$$FTM = \frac{\text{Number of failed matches of registered samples}}{\text{Total number of match attempts}} \tag{2.3}$$

FAR (false acceptance rate) is probability that biometric system incorrectly matches biometric pattern and database template. The total number of comparisons includes failed attempts before the comparison (i.e. FTA, FTE, ...).

$$FAR = \frac{\text{Number of different patterns that were incorrectly matched}}{\text{Total number of comparisons of different patterns}} \tag{2.4}$$

FRR (false rejection rate) is probability that biometric system fails to detect match between input and database template. The total number, as for FAR, includes failed attempts (FTA, FTE, ...).

$$FRR = \frac{\text{Number of incorrectly rejected matches}}{\text{Total number of comparisons of same patterns}} \tag{2.5}$$

FMR (false match rate) is rate, which has the same definition as FAR – a probability that biometric system incorrectly matches biometric pattern and database template – but the total number does not include failed attempts.

FNMR (false non-match rate) is defined as FRR – a probability that biometric system fails to detect match between input and database template – but the total number does not include failed attempts.

Goal is to set the threshold on such value that FAR and FRR are lowest possible and thus the system has the highest performance, however, these rates oppose each other and lowering one will higher other and vice versa. The value where FAR equals FRR (precisely FMR equals FNMR) is called equal error rate (ERR)[4] and is often considered as ideal threshold.

Since FAR and FRR are dependant on threshold and oppose each other, trade-off between FAR and FRR is plotted to show performance as curve with FAR on one axis and FRR on the other. This curve is called receiver operating characteristic (ROC curve). Besides the ROC curve, one can plot DET curve, which differs from ROC in way how the data are plotted (the result is more linear). Plotting more ROC or DET curves in one diagram is preferred way of comparing several biometric systems.

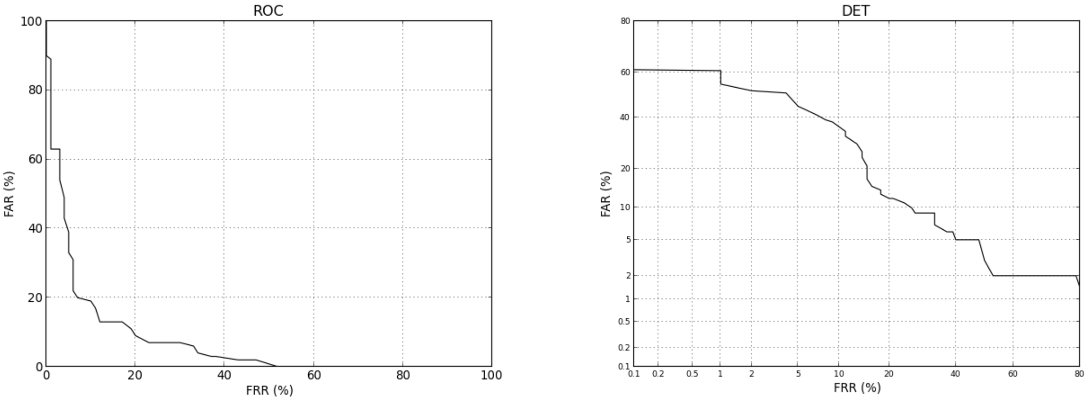


Figure 2.2: ROC curve (left image) and corresponding DET curve (right image) [6].

2.4 Biometric Features

Biometric system can work on one biometric feature, such system is called unimodal, or combination of features, i.e., multimodal system. Multimodal systems are typically more robust and have higher performance.

Biometric features can be divided into two categories [3]:

anatomical features are static non-changeable features such as 2D and 3D geometry of hand or face, fingerprints, eye retina, iris, DNA etc.; and

behavioral features are dynamic changeable features such as voice, dynamic features of signature, movement of lips etc.

Each feature has several attributes such as universality (everybody has this feature), uniqueness (no two persons with same feature), time-invariance (feature does not change in time), ease of obtaining, falsification durability, willingness to enroll, price etc. Furthermore each feature has inter-class variability (variability of feature in different samples) and intra-class variability (variability of feature in samples from same person), these variabilities must be taken to account when deciding which feature will be used.

2.5 Fingerprints as Biometric Characteristic

Human skin consists of two primary layers – epidermis, uppermost layer that serves as barrier, and dermis, an underlying layer. In the skin on human finger there are small extensions of dermis into epidermis. These extensions appear at the surface of the skin as papillary ridges also known as fingerprints. Papillary ridges form patterns on surface of a finger called arch, tended arch, whorl, left loop, right loop and twin loop. These patterns are called fingerprint classes and within them one can find deltas, cores and type lines [4]. Researchers in the field of dactyloscopy formed three laws that allow usage of fingerprints as robust biometric feature. These laws are [7]:

- There are no two poeple with the same papillary ridges.
- Ridges are time invariant (individual changes are within tolerance).
- With no dermal damage (such as deep cuts), papillary ridges can't be removed and will heal in the same manner.

Chapter 3

Liveness Detection on Finger

Attack on fingerprint recognition system can be targeted on a sensor. This means that the fake part of a human body with stolen biometric feature is used to deceive the biometric system. In the area of fingerprint biometrics it's typically a whole fake finger or a thin layer or sleeve, which is put on the attacker's finger, with an artificial papillary ridges.

When the stolen fingerprint is precisely copied onto the fake finger or sleeve, biometric system will consider this input as if it was a genuine one. Protection against this attack is called liveness detection.

Liveness detection is a process that tries to determine whether the input is alive, i.e. the finger is real. If liveness detection works properly, fakes are rejected and security of the system is significantly improved. This process can be seen on Figure 3.1.

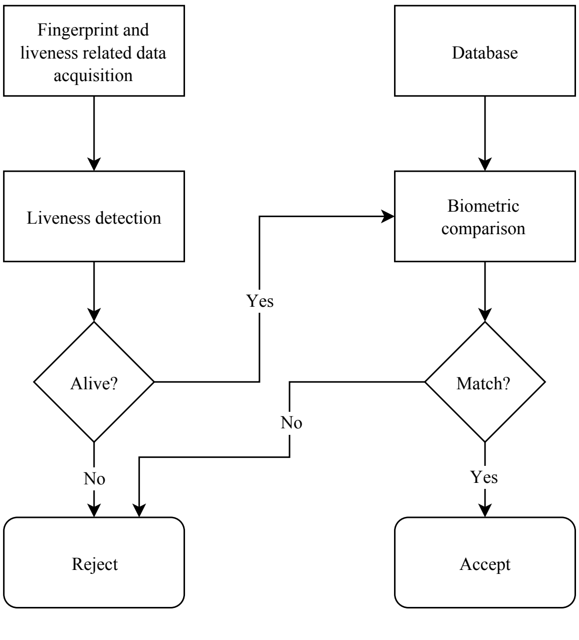


Figure 3.1: Process in biometric system including liveness detection.

Two basic rules have to be abide when detecting the liveness [8]:

- detection must be done at the same time as fingerprint acquirement; and
- it must be done on the same place, i.e. same part of same finger.

If these two rules were not followed, an attacker could deceive the biometric system: use at first the fake finger for fingerprint extraction and then present his own real finger for liveness detection or use the fake finger at sensor and the real finger at liveness detector [9]. In next several sections, some of the main methods of liveness detection on finger will be described.

3.1 Perspiration

A tip of human finger has hundreds of sweat glands which fake finger or sleeve does not posses. Observing a sweat distribution in papillary ridges can be used to tell whether the provided sample is alive or not. This is often done by collecting several image frames for small period of time (typically around 5 seconds). Drawback of this method is that it has high intra-class variability. There is an example of this method is on Figure 3.2.

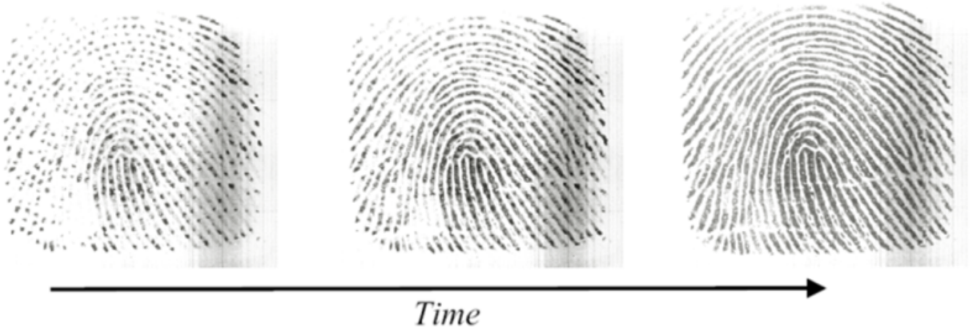


Figure 3.2: Changes of sweat distribution in papillary ridges over time [10].

3.2 Spectroscopic characteristics

Liveness detection using spectroscopic characteristic of a human skin is based on optical (primary reflexive) proprieties. Living human skin, due to chemical composition (water, arterial and venous blood, lipids, melanin etc.) and layered structure, has optical characteristic that has effect on absorbance and scattering (see Figure 3.3) [9].

Skin on a finger is illuminated with light of various wavelengths, amount of the returned light is affected by above mentioned structural and chemical properties. Various wavelengths are used because of a different ability to penetrate the skin – blue light is absorbed quickly while red or infrared light penetrates the tissue deeper [9, 11].

This method can be improved with crossed linear polarization filters, one polarizing light from light source and the other at the imaging sensor. Light seen by the sensor had to pass through environment (e.g. human skin) where scattering events randomized the polarization [12].

3.3 Ultrasonic Technology

Reflection of high frequency sound wave is different for human skin and other materials. Sound, before reflection, penetrates upper layer of skin where it is scattered. This property gives information about layers of a finger and allows ultrasonic sensor to detect liveness

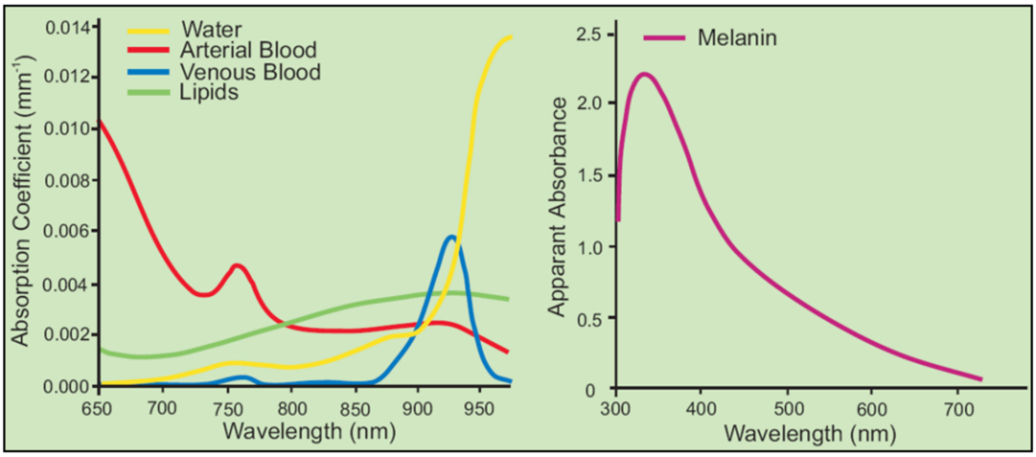


Figure 3.3: Absorption of different wavelengths in living tissue [9].

of provided input [9].The difference between signal from the real finger and the fake finger is in spectral domain – returned signal undergoes Fourier transform to obtain the these characteristics.

This technology is used in sensor from Optel, they also claim, that their sensor is capable of detecting pulse. This could also be used for liveness detection (as described later) [13].

3.4 Temperature and Temperature Stimulus

Temperature of epidermis at the tip of human a finger is typically in range of 25-37 °C and the temperature measurement can be seen as a method of liveness detection. Thermo-camera measurement can be seen on Figure 3.4. This approach has serious issues [9]:

- people with blood circulation problems will have colder hands and fingers, liveness module will wrongly decide that this is fake input, widening the working range will result in higher probability that the liveness detector will be deceived,
- thin finger sleeve or fingertip attached to attacker's finger could deceive liveness detection since temperature shielding of thin material would be within limits.

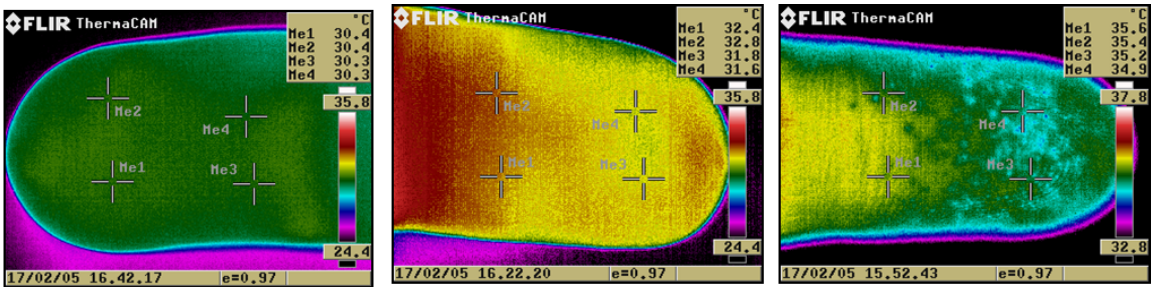


Figure 3.4: Images from thermo-camera showing fingertip and its temperature [9].

Autonomic nervous system reacts on hot or cold stimulus by increasing or decreasing the blood flow and does it with little delay after stimulus application. Contact platen of biometric system can give such stimulus and delayed changes of blood flow can be measured to determine whether the provided sample is real finger or not.

3.5 Pressure Stimulus

Liveness detection based on pressure stimulus has two general methods, both tracing changes before pressure and after [9]:

- observing elasticity of papillary ridges, where inter-papillary distance is measured (change can be up to 20%); and

- observing changes in color of a finger tip, pressed finger has significant change in green component of RGB model (up to 42), see Figure 3.5.

Drawback of this method is that the finger is pressed against the transparent platen, where latent fingerprint can be left.

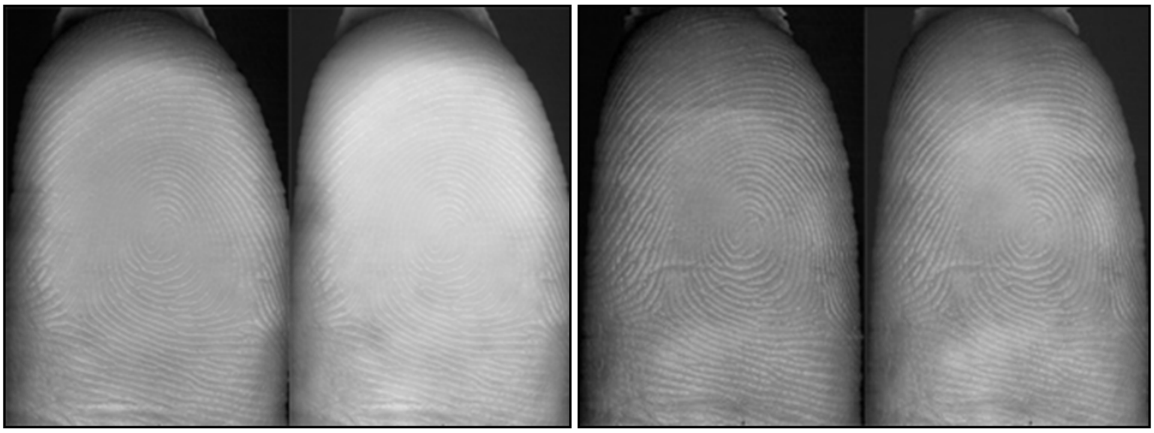


Figure 3.5: Color changes of fingertip pressed tightly (left comparison) and loosely (right comparison) [14].

3.6 Electrical Properties

Measuring electrical properties of the provided input can be used as liveness detection. Such properties are conductivity or resistivity, where electrodes are used to determine this physical property, and bio-impedance, where altering current is applied on hand and impedance is measured. Measuring of conductivity highly depends on humidity and salinity of the skin, this varies with stress and can be faked with brine or saliva applied on artificial finger, which is a drawback of this method [9].

3.7 Pulse and Blood Oxygenation

In systolic phase of cardiac cycle, heart is contracted and blood is pumped into arteries, which creates longitudinal wave that spreads out to the periphery of a human body. This wave can be measured and is colloquially known as a pulse. Pulse wave also changes volume of blood vessels and this change also affects surface of the skin, where it can be measured and used to detect liveness. It is done by using macro-objective to determine changes in inter-papillary distance or laser to measure distance changes of pulsing fingertip [14].

Oxygen level in blood varies in time and saturated blood reacts differently on near-infrared light than the unsaturated. Visible red light at 660 nm and infrared light at 940 nm is used and absorption of this light is monitored.

3.8 Biochemical Properties

Odor of sweat on fingertip can be measured by device called electronic nose, where array of chemical sensors detect molecules that evaporated from tested skin sample.

With this approach, precise sensor positioning is required – odor liveness detection has to be checked on same the part of skin as was used for fingerprint sensing and moreover it was shown that gelatin fake fingers can deceive the sensor [13].

3.9 Image Quality

Quality of fingerprint image can be monitored to detect liveness in the provided input. This approach works only with the acquired image and the only needed changes are in software. Hardware of the sensor remains the same.

This method can be seen as two-class classification for which appropriate feature has to be selected and extracted. These properties are ridge width, continuity, clarity and integrity. To measure these properties, several information sources are used: orientation filed, Gabor filters, pixel intensity and power spectrum [15, 4].

Chapter 4

Analysis and Proposed Method

In this chapter, at first, analysis of liveness detection for fingerprint recognition technology with focus on spectroscopic properties will be presented and later, based on this analysis, the method itself will be proposed.

Fingerprint sensor can be generally of two types:

touch, where transparent platen is present on the sensor, for purpose of fingerprint recognition the finger is pressed against this platen; and

touchless, where no platen is used and fingerprint is obtained when finger is situated near the biometric sensor – typically over or beneath the sensor.

This construction determines which liveness detection can be used – some of the methods mentioned beforehand need a contact with the finger to detect liveness. For purposes of this work, the sensor from Touchless Biometric Systems will be used, where no platen is present. Picture of this sensor is on Figure 4.1. Illumination in this sensor is done by array of LEDs and the image is captured by three gray-scale cameras.

Figure 4.1: 3D Enroll Station from Touchless Biometric Systems [16].

4.1 Usable wavelengths

At first, it must be determined on which wavelengths human skin reacts the most, i.e. which wavelength will be absorbed, reflected and scattered in such manner that it will be unique for the skin itself allowing to use it as liveness detection method. TBS sensor has built-in LEDs with wavelengths of 470 nm, 550 nm, 700 nm and 800 nm. A paper has been published [11], where promising usage of these LEDs was presented. However, no other wavelengths were used, LEDs illuminated the finger unevenly and spectroscopic characteristics were not studied with finger under different conditions (cold, wet etc.).

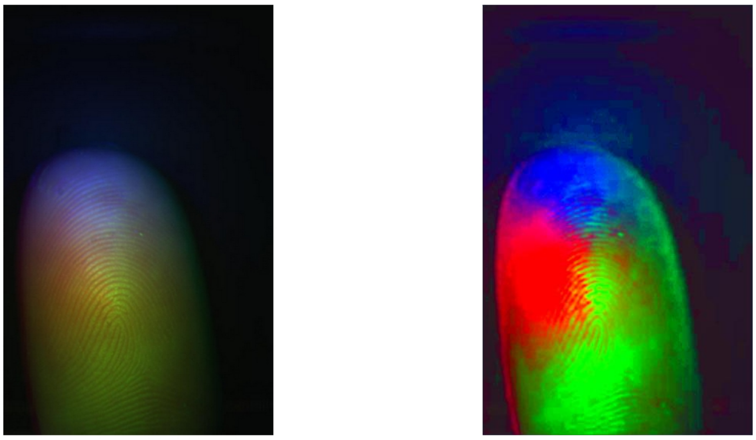
In the first stage, LEDs with wavelengths of 400 nm, 410 nm, 470 nm, 525 nm, 550 nm, 570 nm, 590 nm, 635 nm, 700 nm and 800 nm will be used, some of them built-in, other with need of hardware modification of the sensor. Lumidigm, a company that manufactures biometric sensors with liveness detection, uses LEDs at 430 nm, 530 nm and 630 nm [12]. It is expected that wavelengths around these values will be most promising.

When the most usable wavelengths will be found, a skin will be subjected to different conditions, such as wet, cold and dirt, to simulate the real world conditions. This will narrow the set of selected LEDs.

4.2 Proposed method

Finally, these LEDs will be used to obtain several images of a fingertip under various spectral conditions. Cameras in the TBS sensor are gray-scale, so an composite image in false colors will be produced. This composite image will undergo an enhancement process, such as high or low pass filtering and decorrelation stretching. These image enhancements are considered, so the properties of living skin are emphasized.

Decorrelation stretching is an image enhancement technique that stretches the color channels of image, so the colorspace is filled and colors are uncorrelated [17] (see Figure 4.2). This technique is also used by sensors from Lumidigm [18].



(a) Input image [11]

(b) Image after decorrelation stretching

Figure 4.2: Decorrelation stretch applied on multi-spectral image of a fingertip.

High or low pass filtering can be used to emphasize edges (high pass filter, Gabor filter) or to blur the image. High frequencies (edges) carry more localized information while blurred image (low frequencies) emphasizes less localized information. Spatial-Frequency domain of image can be produced by wavelet transformation. Information in this domain describes not only frequencies but also where is which frequency located.

Region of interest (ROI) needs to be specified before feature extraction. Background of image contains dark pixels, which could influence the output. In previous work [11], ROI in a shape of rectangle was used. This approach can be enhanced by detecting the area of image where the fingertip is located. Since the background has dark (if not black) pixels, image can be thresholded, separating the background from finger. The thresholded image is suitable for contour detection.

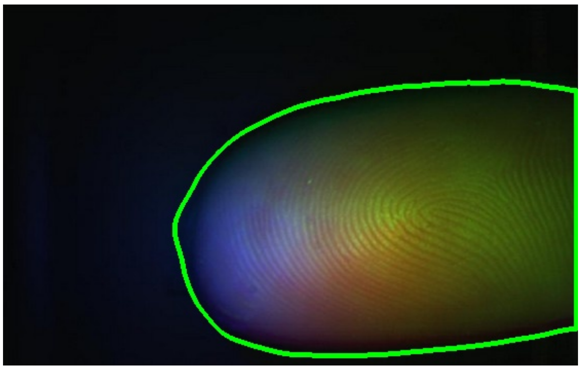


Figure 4.3: Fingertip detection using OpenCV contour detection.

Area defined by the found contour (see Figure 4.3) contains pixels of interest, however, the side parts of fingertip are also dark. Since this can cause inaccuracy in contour detection, the area described by the found contour is not final ROI. Adjustment, such as shrinkage of contour, has to be done carefully – it is expected that side of a finger, due to large reflexion angles, carries information essential to optical liveness detection.

Contour can be fit with an ellipse, which can define final ROI. Ellipse needs to be inscribed, so it does not contain background. Major and minor axis are scaled to 90% of original ones. This results in approximation of inscribed ellipse. Besides the ellipse, contour can be fit with a line. While the ellipse specifies ROI, the line specifies approximate axis of fingertip. This axis describes orientation of a grid used for feature extraction. Inscribed ellipse and finger axis are on Figure 4.4.

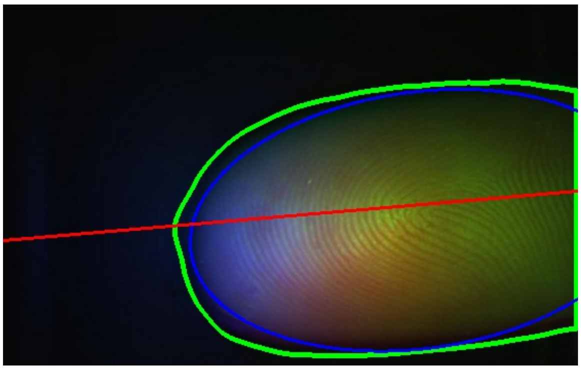


Figure 4.4: Inscribed ellipse (ROI, blue color) and axis of fingertip (red color).

With the ROI specified, feature extraction is the next step. In the previously mentioned work [11], these quantitative properties have been used with promising results:

- pixel intensity arithmetic mean,
- pixel intensity standard deviation,
- pixel intensity median,
- histogram mean and median; and
- histogram standard deviation.

This properties can be considered as local or global. Global properties are enumerated from whole the ROI, local properties are enumerated from reduced set of pixels given by an axis of fingertip. The above described decorrelation stretch enhancement increases differences among colors and since various colors penetrate human skin differently, intensities of color channels along with beforehand mentioned features will serve as input vector for machine learning algorithm. The machine learning algorithm will be selected with respect to its performance. The proposed methods of algorithms are [19]:

- feed-forward neural networks,
- support vector machines (SVMs),
- AdaBoost,
- decision trees,
- naive Bayes classifier; and
- random forests.

It is probable that the problem won't be linearly separable and it is expected that linear models will perform insufficiently. SVMs will be used with linear kernel as well as polynomial one. Process of feature extraction and liveness detection is on Figure 4.5.

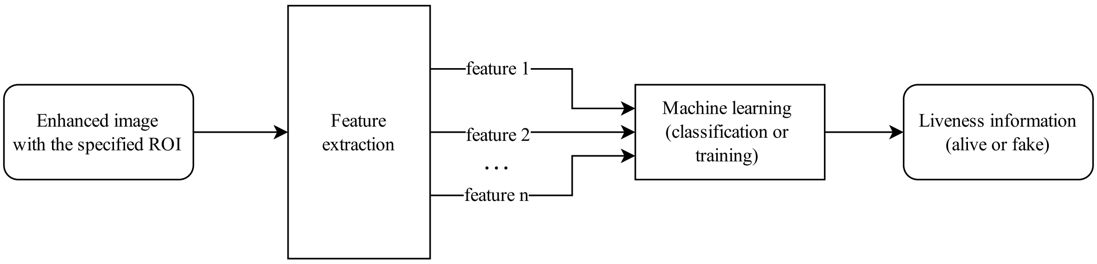


Figure 4.5: Process of feature extraction and liveness detection.

4.3 Hardware adjustments

The method proposed in previous section can be done purely in software. However, its input, the multispectral false-color image of a fingertip, can be provided only with hardware adjustments. The TBS 3D Enroll Station has a limited space inside and only two additional LEDs can be installed (typical circular package LEDs). This fact limits the wavelengths that can be used together – the built-in ones (470nm, 550nm, 700nm and 800nm) and two additional. For simple schematics see Figure 4.6.

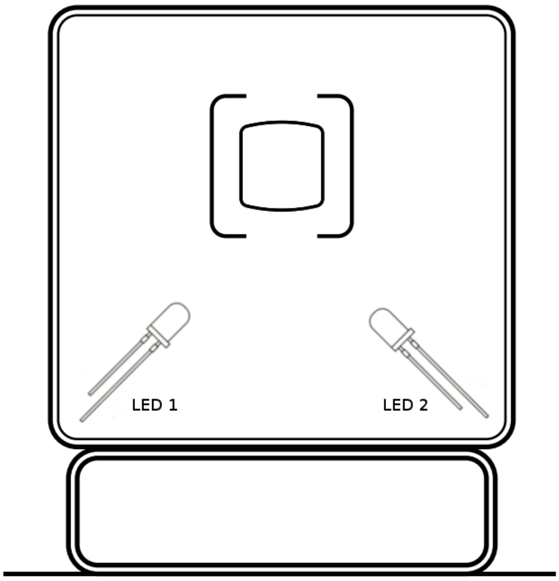


Figure 4.6: Front-view schematics of TBS 3D Enroll with two additional LEDs approximate position.

Chapter 5

Selected Wavelengths

5.1 Selected LEDs

At early stages of work, there were LEDs proposed with wavelengths 400 nm, 410 nm, 470 nm, 525 nm, 550 nm, 570 nm, 590 nm, 635 nm, 700 nm and 800 nm. These diodes were examined and it was found that external diodes with wavelengths around green and yellow part of spectre are unusable.

Human eye is more sensitive to those colors and so the manufacturers can produce LEDs with lower luminosity that are perceived as bright as more luminous blue or red. Moreover, human skin reflectance in green part of spectre is lower than in the others [20]. These two facts, lower luminosity and lower reflectance, causes that the finger was not illuminated enough, see Figure 5.2 c). The picture is highly enhanced to see any finger at all and the finger looks grayish.

Problem of this kind could be solved in two ways:

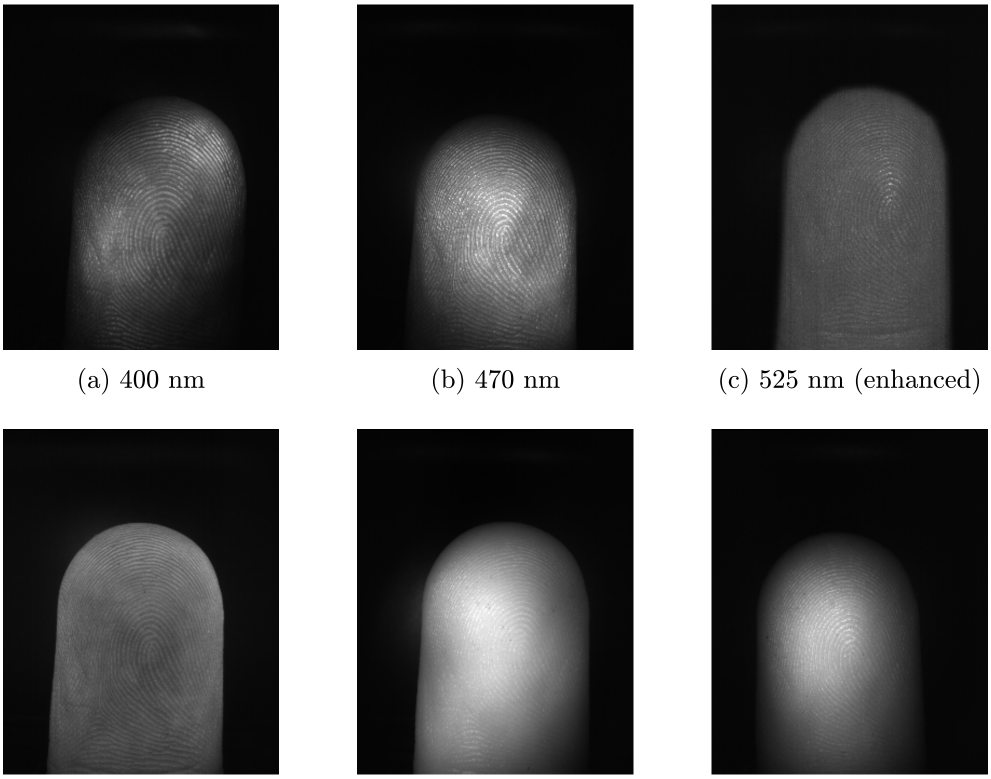
1. Use LEDs with higher luminosity. Drawback of this approach is the hardware constriction of two diodes – two bright diodes would enlighten the finger as two spot lights and the illumination would be uneven.

2. Use longer shutter speeds. Downside of longer exposure times is that it is hard to keep the finger still. Any movement brings ghosting.

Based on above mentioned reasons it was decided to use green diodes already present in the device. There is large array of these LEDs, which unlike the external diodes, produce highly even illumination, see picture c) in Figure 5.2.

On Figure 5.2, there is a comparison of illumination by internal red diodes present in the device (picture e)) and external red diodes (picture d)). The main difference is brighter illumination in case of external LEDs, which should not matter for machine learning algorithms. Since the internal blue diodes illuminates the finger unevenly [11], it was decided that the two external diodes will be from blue part of spectre. Finally, chosen were the 400nm diodes – this wavelength is on the edge between the visible and UV part of spectre and the pigment melanin present in the skin does have highly different absorption for the UV light [20].

The final combination of three colors is 400 nm, 550 nm and 700 nm – two internal and one external. The quick sequencing will produce three grayscale images under different wavelengths, where the one illuminated by 550nm can be used for fingerprint



(a) 400 nm

(b) 470 nm

(c) 525 nm (enhanced)

(a) 550 nm

(b) 635 nm

(c) 700 nm

Figure 5.2: Grayscale images of finger under different wavelengths.

recognition. This ensures the space and time requirements mentioned. The biometric identification/verification will be at the same time and same space as the liveness detection.

5.2 Circuit Connection

Prototype for connection of external diodes was created from Arduino board with breadboard shield. Schematics for involved circuit can be seen on Figure 5.3. Besides LEDs, the main part is switch and the variable resistor. The former was used to trigger the LEDs during imagining process and the later was used for fine tuning of diode brightness.

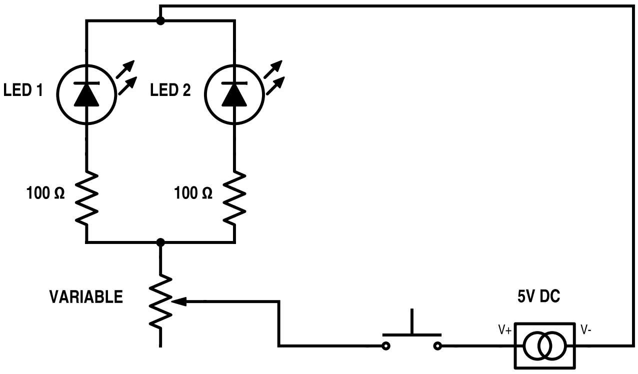


Figure 5.3: Final circuit connection.

Chapter 6

Image Enhancement

6.1 Low-pass Filter (blurring)

Small dirt or moisture will appear on acquired image as noise, which undesirable. From spectral view the noise is a high frequency component of image and so, the noise can be removed using low-pass filter. Low-pass filter is blurring or smoothing and for this project Gaussian blur was selected [21].

This blurring technique is a convolution filter with kernel that approximates 2D Gaussian distribution. For this work, kernel size was 5×5, which ensured that the noise was removed while the blurring was not strong enough to oversmooth the image. See Table 6.1 for example approximation of Gauss function in form of 5x5 kernel.

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Table 6.1: Example of 5x5 Gaussian blur kernel.

6.2 Per-channel histogram equalization

Histogram is an intensity distribution in image. It can be constructed only for gray scale images or per color channel. Some parts of histogram can be underpopulated, representing low number of pixels with given intensity. Histogram equalization is process that increases contrast of the image – it stretches the intensity into those underpopulated areas.

Cumulative distribution function is used in histogram equalization and is defined as [21]:

$$H'(i) = \sum_{j=0}^{i-1} H(j) \tag{6.1}$$

where $H(j)$ is image histogram. This function is then scaled that maximum is 255. Finally, the scaled function is used as look-up table for new pixel values. This approach is a replacement for decorrelaton color stretching – no suitable library was found to use in this work.

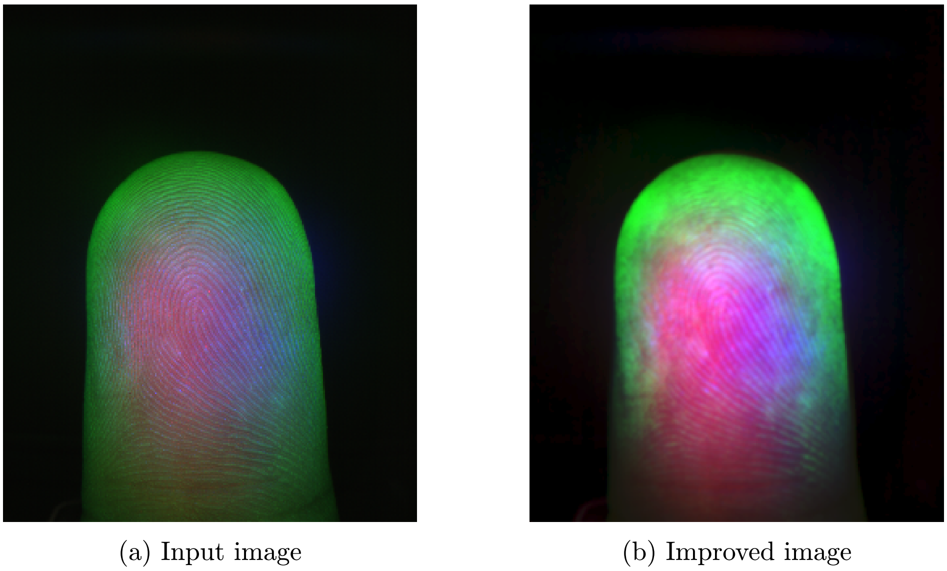
6.3 Intensity adjustment

Intensity of the image needs to be adjusted after the previous step. The formula for intensity adjustment is following:

$$\text{adjusted image} = 255 \left(\frac{\text{input image}}{255} \right)^p \tag{6.2}$$

The p is power of the adjustment effect. Numbers between 0 and 1 will produce an image with dark pixel much brighter and numbers larger than 1 will result in image with dark pixels much darker. Bright pixels get also darker or brighter, but the change is more subtle.

On Figure 6.1, there is comparison of image before and after improvement process.



(a) Input image

(b) Improved image

Figure 6.1: Comparison of sample before and after improvement process.

Chapter 7

Feature Extraction and Dataset

7.1 Feature Extraction

Feature extraction is done, as proposed, using contour approach:

1. Image is thresholded, blurred and thresholded again. This produces image with smoother blob than single thresholding.
2. Contours are found for the blob image, the largest contour is selected and base ellipse is fitted to this contour.
3. Several concentric ellipses are defined based on the base ellipse (including it), see Figure 7.1.
4. Every ellipse is then approximated with polygon with an angle of one degree between the subsequent vertices.
5. Pixel values are collected from each polygon's vertices, coordinates out of the image are omitted.
6. For pixels of each polygon, several statistical values are computed. These values together creates feature vector of the finger.

Extracted statistical values from each ellipse are:

Mean (average) $\overline{x}$ of n elements $a_1, ..., a_n$ is defined by formula [22]:

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} a_i \tag{7.1}$$

Median of ordered set with cardinality n is number $\tilde{x}$, which is for odd n [22]:

$$\tilde{x} = a_{(n+1)/2} \tag{7.2}$$

and for even n it is average of elements with indexes $n/2$ and $n/2 + 1$ [22]:

$$\tilde{x} = \frac{a_{n/2} + a_{n/2+1}}{2} \tag{7.3}$$

This number divides input ordered set into two sets with equal cardinality.

Standard deviation of input set is number σ that is defined by formula [22]:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{7.4}$$

Values of 1^{st} and 3^{rd} quartile are numbers Q_1 and Q_3 that – similary to median (which is 2^{nd} quartile) – splits ordered input set into 25% of lowest values and 75% of highest values (in case of 1^{st} quartile) and into 75% of lowest values and 25% of highest values (in case of 3^{rd} quartile).

Number of outliers, where outlier is defined as value that is not within interval:

$$\tilde{x} \pm (Q_3 - Q_1)$$

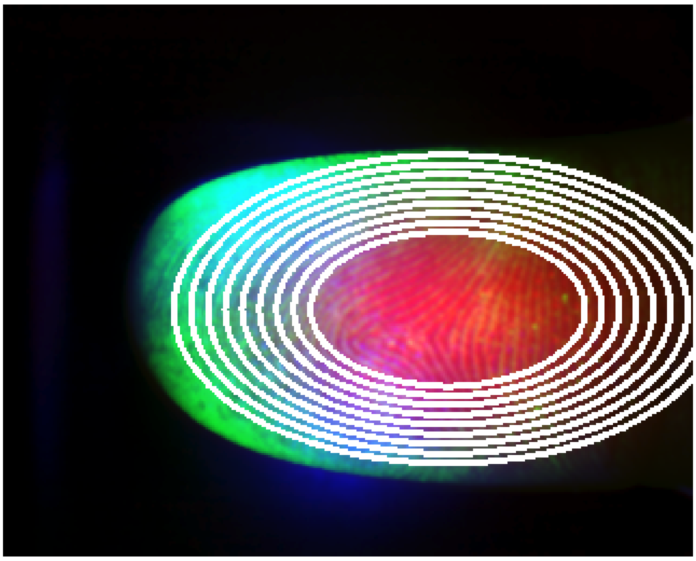


Figure 7.1: Ellipses for feature extraction.

7.2 Dataset

Used dataset has 114 samples divided into a genuine part, where there are samples of fingers under different conditions, and a fake park, where custom made fakes were created and sampled.

7.2.1 Genuine Fingers

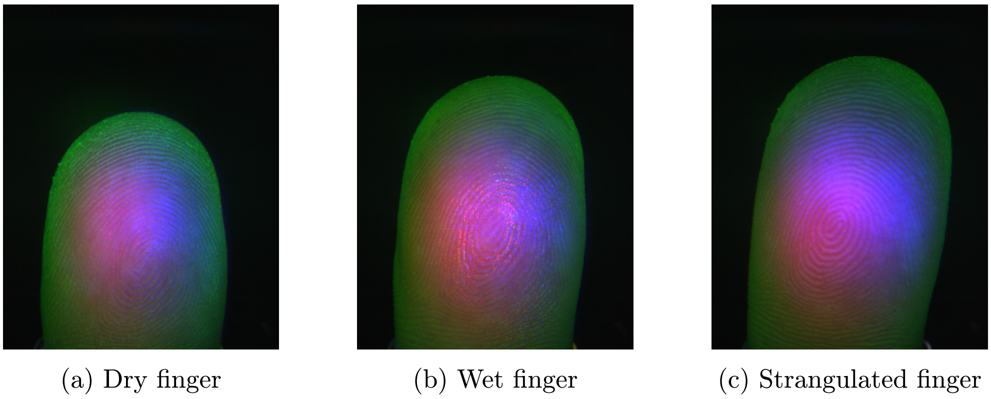
Images of genuine fingers were obtained from persons of both sex with age ranging from 23 to 69 years. Fingers were captured under three different conditions:

dry finger was captured is order to cover the most frequent use case,

wet finger represent use case when the user has moisture on a finger. This covers fingers with sweat and fingers wet from water (e.g. rain); and

strangulated finger by the rubber band was included, so the case where there is restricted blood flow (e.g. cold hands) is present, but the finger itself is real and should pass the liveness detection.

Total size of genuine part of dataset is 32 fingers each per 3 samples (dry, wet and strangulated). For example of one finger see Figure 7.2.


(a) Dry finger

(b) Wet finger

(c) Strangulated finger

Figure 7.2: Samples of one genuine finger in dataset.

7.2.2 Fake Fingers

Fakes were created from materials such as: wax, brown acrylate, polymer clay (Fimo material), starch, wood, silicone or latex. Organic samples were created from sausage and potato. Some selected fakes can be seen on Figures 7.3 and 7.4.

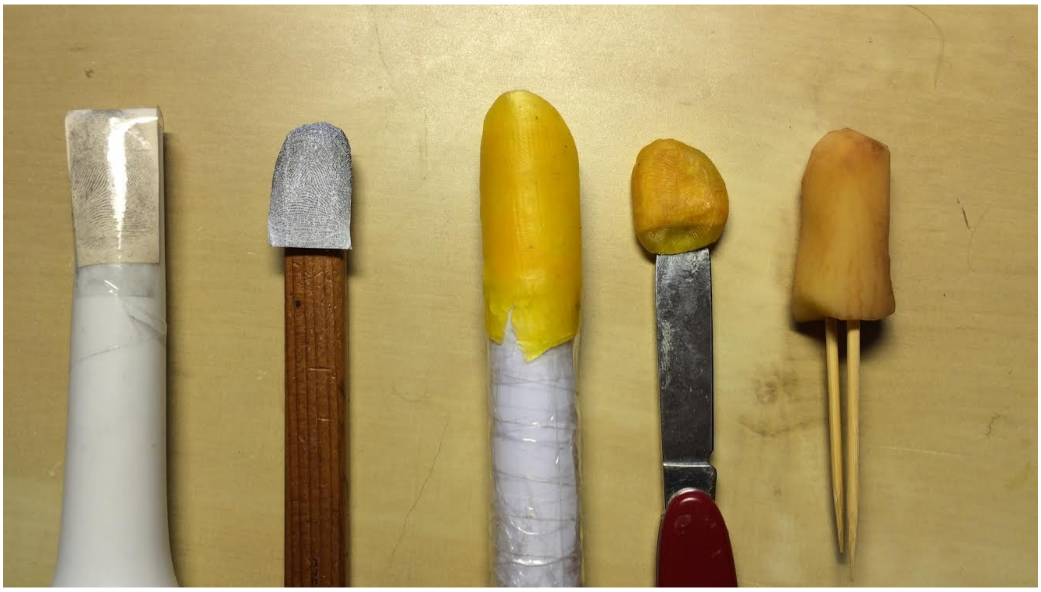


Figure 7.3: Various fake fingers used in dataset.

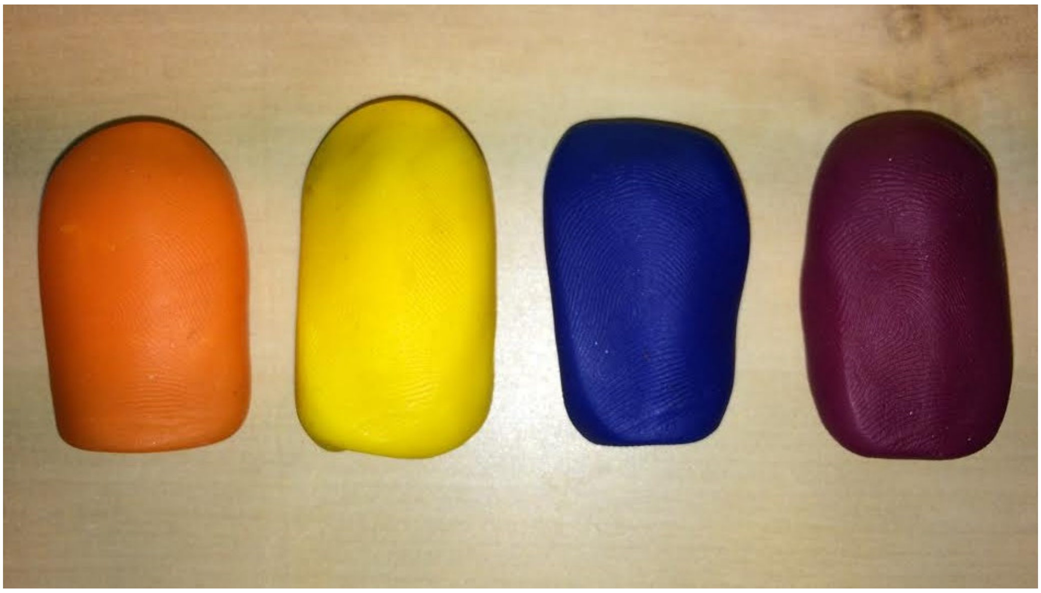


Figure 7.4: Various fake fingers from Fimo material.

Chapter 8

Machine Learning and Evaluation

8.1 Machine Learning

8.1.1 Cross-validation

To tell whether the model is usable and how successful it was, a model evaluation (validation) process is needed. For purposes of this work, 10-fold cross-validation was selected. Cross validation is a technique, which takes dataset and splits it into two subsets – training and testing part. The k-fold means that the dataset is split into k parts, where (k-1)/k of dataset is used for training and 1/k is a test part. This is done k-times and the test part is rotated over whole dataset. To have same distribution of classes in the training and testing part as is in the whole dataset, stratification is used.

8.1.2 Principal Component Analysis

Principal component analysis (PCA) is a method of preprocessing the data before they are used in a training phase. PCA transforms the dataset to new coordiate system, so that the coordinates are linearly uncorrelated (principal components). Variance along each coordinate (axis) is maximized and descending with each subsequent axis. This creates coordinate hierarchy where the last has the least variance and can be omitted with accepting small information loss [19].

In this way, PCA is used to reduce dimensionality, feature vectors are shorter which some machine learning algorithm require for better training. Lower dataset dimensionality also speeds up the training phase. On Figure 8.1, there is visualization of dataset reduced from 189 to 2 dimensions.

8.1.3 Bayesian Classifier

Bayesian classifier, more precisely Gaussian naive bayes classifier, is an algorithm from family of methods called Naive Bayes. The word *naive* refers to an assumption that there is no dependency in every pair of variables [19].

These methods are based on Bayes' theorem:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{8.1}$$

Using this theorem, afore mentioned assumption of independency and the fact that $P(x_1, ..., x_n)$ is constant, the predicted class $\hat{y}$ of input vector $x_1, ...x_n$ is given by following

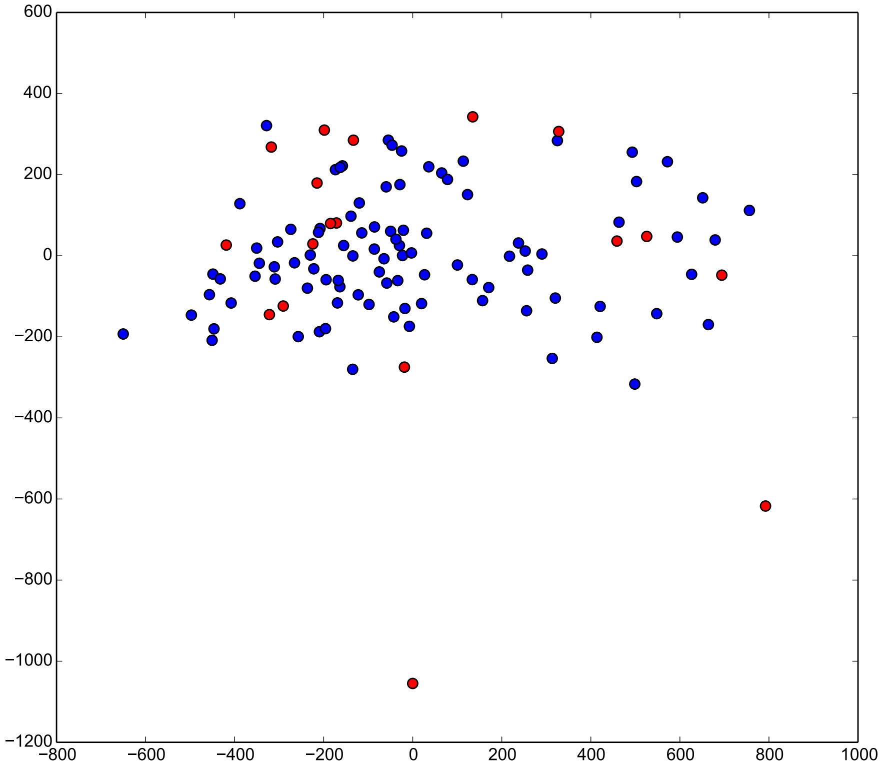


Figure 8.1: 2D scatter plot of dataset after PCA transformation.

classification rule [19]:

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y) \tag{8.2}$$

Used classifier is Gaussian because the likelihood $P(x_i|y)$ is assumed to be Gaussian. Confusion matrix is in Table 8.1. The classifier recognizes the genuine fingers (accuracy 94%), but struggles to recognize fake fingers (accuracy 28%).

	predicted fake	predicted genuine
true fake	28%	72%
true genuine	6%	94%

Table 8.1: Confusion matrix for Bayes classifier.

8.1.4 Support Vector Machine

Support vector machines (SVMs) are machine learning methods that classifies two classes by using hyperplane to divide the input space [19]. This division is done in a way that margin around – dividing hyperplane and closest input vectors – is maximized (see Figure 8.2).

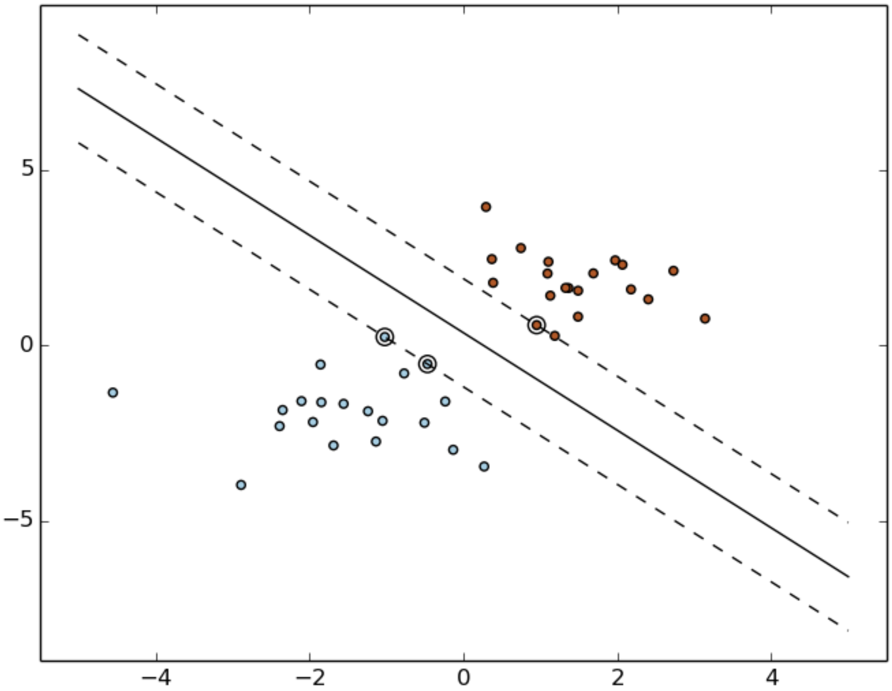


Figure 8.2: Hyperplane (solid line) of support vector classifier and margin (dashed lines) that is maximized during the training process [19].

As is, this hyperplane can classify input set into two classes if and only the input set is linearly separable. However, linear separability is not frequent case and support vector machines comes with solution colloquially called „kernel trick“. The solution maps the input space into space with higher dimension. In this higher dimensional space, input data can be linearly separable and thus a reasonable hyperplane can be found. For this work, Radius Basic Function (RBF) kernel was selected [19].

In Table 8.2 there is confusion matrix of support vector classifier with linear kernel (no mapping into higher dimensions was used) and in Table 8.3 is confusion matrix of support vector classifier using rbf kernel. It can be observed that using rbf kernel improves the accuracy especially for fake fingers.

	predicted fake	predicted genuine
true fake	56%	44%
true genuine	5%	95%

Table 8.2: Confusion matrix for SVM with linear kernel.

	predicted fake	predicted genuine
true fake	61%	39%
true genuine	2%	98%

Table 8.3: Confusion matrix for SVM with RBF kernel.

8.1.5 Decision Tree Classifier

Classifier based on decision tree is machine learning algorithm that creates tree, where nodes are decisions rules on input features and leaves represent class to which the input vector belongs [19]. Decision tree created from whole finger dataset is on Figure 8.3. Confusion matrix of decision tree classifier is in Table 8.4. The classifier performs bad when dealing with fake fingers.

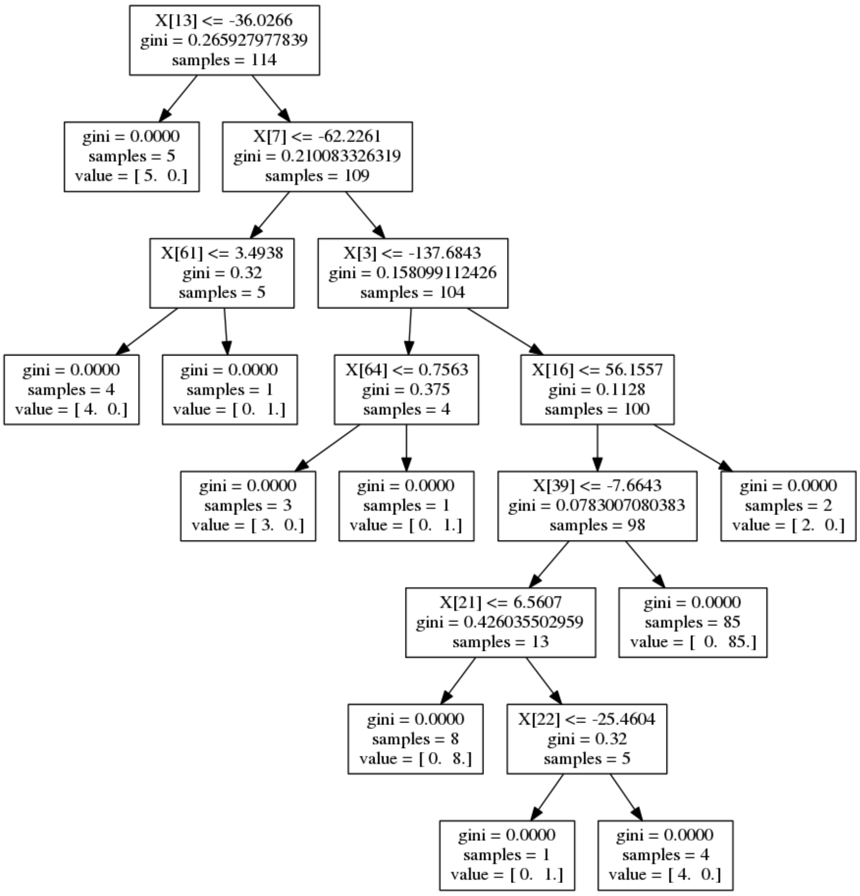


Figure 8.3: decision tree of decision tree classifier trained on whole dataset.

	predicted fake	predicted genuine
true fake	28%	72%
true genuine	15%	85%

Table 8.4: Confusion matrix for decision tree classifier.

8.1.6 Random Forest Classifier

Random forest classifier is, same as decision tree classifier, based on creation of decision tree, more precisely forest of trees. Classification is then performed by taking the mode of classification given by the single trees [19].

The single trees are created from randomly chosen samples from training set. This approach should lower the chance of over fitting problem of decision tree classifier. However, with 10 decision trees of same depth as for decision tree classifier, the classification is worse (considering the fakes) compared to decision tree classifier, see Table 8.5.

Using two deeper random trees, the accuracy of recognizing fake finger can go up to 56% (true negatives), this result is highly random (due to nature of this algorithm) and can be as low as 27%, Also true positives rate is lowered. For best run results see confusion matrix in Table 8.6.

	predicted fake	predicted genuine
true fake	11%	89%
true genuine	3%	97%

Table 8.5: Confusion matrix for random forest classifier.

	predicted fake	predicted genuine
true fake	56%	44%
true genuine	27%	73%

Table 8.6: Confusion matrix of random forest classifier using 2 deep trees.

8.1.7 AdaBoost

Another approach to classification called boosting – a process that creates and trains sequence of several weak learners. This means that multiple models are trained on repeatedly modified training set and that they are better than random guessing, but couldn't be used alone due their bad performance (hence the name weak) [19].

This sequence of weak classifier is then used to predict the class in a way that every classifier predicts class and final prediction is weighted majority vote or sum. Every classifier in sequence is trained on dataset that was weighted (a weight for each sample of training set), with increasing weights of incorrectly classified samples from previous classifier. This process ensures that later classifiers are forced to concentrate on samples hard to classify [19].

This algorithm is so-called meta, it does not itself classify but needs another algorithm that will serve for creation of weak learners. Best results were achieved using linear support vector machine, see Table 8.7 for confusion matrix.

AdaBoost with linear SVM using perceptron as base algorithm showed significant increase of accuracy regarding the true negatives rate. However, using perceptron often caused failure of training process – AdaBoost can not be done if some of the weak learners has accuracy less then 50%. For confusion matrix of successful run of AdaBoost with perceptrons, see Table 8.8. Moreover, this approach show low accuracy of recognizing genuine fingers.

	predicted fake	predicted genuine
true fake	56%	44%
true genuine	5%	95%

Table 8.7: Confusion matrix for AdaBoost using linear SVM for weak learners.

	predicted fake	predicted genuine
true fake	83%	17%
true genuine	44%	56%

Table 8.8: Confusion matrix for AdaBoost using perceptron for weak learners.

8.1.8 k-Neighbors Classifier

Algorithm called k-Nearest Neighbors used in k-Neighbors classifier differs from other used machine learning algorithm during the training phase. The training does not really train the model, but it stores the input dataset. Classification is done when unlabeled data are on input [19].

The classification process of k-Neighbors classifiers is done in two steps:

1. At first, it checks the stored data and looks for k samples that are most similar to the one on input (hence the name k-nearest neighbors).

2. Based on output from first step, the classification is done using the neighbors' classes.

The distance of two samples used in neighbors look up needs to have a metric specified, such as Minkowski metric (or its special case, the Euclidean metric). Best results were achieved using Minkowski metric with order of 2 (2D Ecuclidean metric) and $k = 3$ (3 neighbors lookup) with uniform weights. For k-Neighbors classifier performance, see Table 8.9

	predicted fake	predicted genuine
true fake	28%	72%
true genuine	3%	97%

Table 8.9: Confusion matrix for k-Neighbors classifier.

8.1.9 Bernoulli Restricted Boltzmann Machine

Preprocessing of the dataset has high influence on subsequent classification. For all previous classifiers, principal component analysis was used to reduce number of dimensions. Another approach is to train an unsupervised neural network, such as restricted Boltzmann machine.

Restricted Boltzmann machine (RBM) is neural network cabable of non-linear extraction of components, in this way, the training set is transformed (as in PCA) into another, more suitable for classification – especially with linear classifiers [19]. The input data for RBM has to be in range from 0 to 1 or binary, this means that dataset has to be scaled to this interval.

Linear support vector machine was selected as subsequent classifier, evaluation of this approach is shown in Table 8.10. RBMs are probabilistic models and the values in confusion matrix are the best achieved. Some other runs showed false positives up to 50%.

	predicted fake	predicted genuine
true fake	72%	28%
true genuine	15%	85%

Table 8.10: Confusion matrix for RBM + linear SVM classifier.

8.2 Implementation Details

Implementation was done in programming language Python using mainly libraries scikit-learn (version 1.6.0) [23] and openCV (version 2.4.9) [24]. OpenCV was used for image handling, and feature extraction. The library scikit-learn provided implementation of algorithms for machine learning (preprocessing, machine learning, metrics). Optimal hyper parameters (e.g. learning rate, number of components, number of iterations…) for classifiers were found by using GridSearchCV.

Chapter 9

Experiments summary

9.1 Machine Learning Summary and ROCs

Several classifier were examined with various results, their mean F_1 score using 10-fold cross validation is in Table 9.1. F_1 score can be computed as follows:

$$F_1 = 2\frac{precision * recall}{precision + recall} \tag{9.1}$$

where precision is defined as:

$$\frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{9.2}$$

and recall as:

$$\frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{9.3}$$

The Table 9.1 of scores cannot stand alone, itself it gives a overview how classifiers performed compared to each other, but it says less on what was the confusion. This information is in confusion matrices that were referred in subsection of given classifier.

Gaussian Classifier	Linear SVM	RBF SVM
90%	93%	95%
Random Forest Classifier	Decision Tree Classifier	k-Neighbors Classifier
73%	86 %	92%
AdaBoost Classifier (linear SVM)	AdaBoost (Perceptron)	RBM + linear SVM
93%	70%	90%

Table 9.1: F-1 score using 10-fold cross validation.

For better comparison of classifiers, see a plot with ROC curves on Figure 9.1. From the plot one can deduce, that linear and RBF SVM had the best performance. Random forest classifier performed also well, but its randomness can not ensure consistent accuracy.

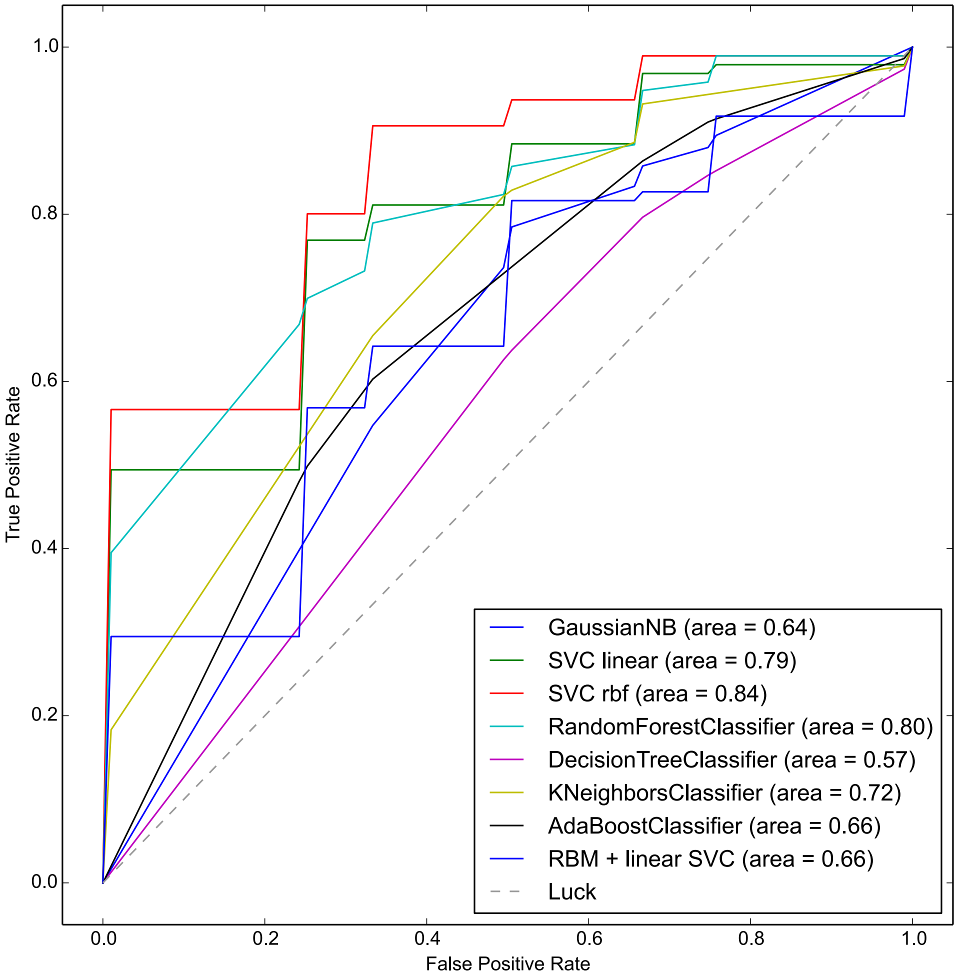
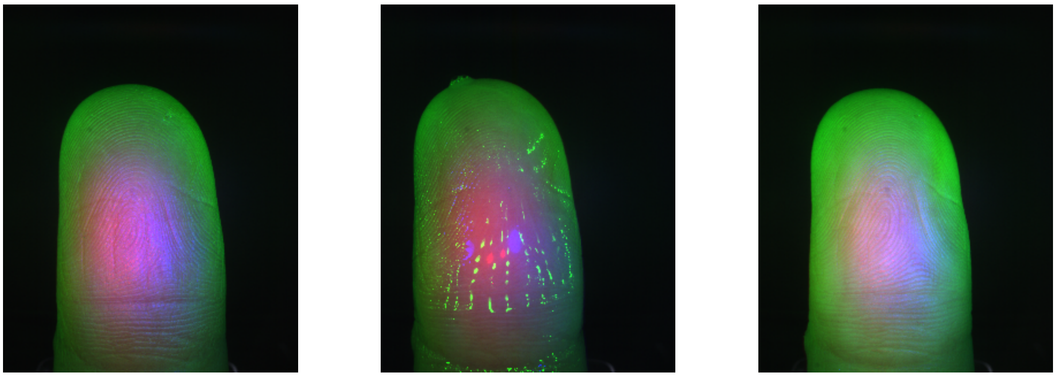


Figure 9.1: ROC curves of used classifiers.

9.2 Experiments With Finger Subsets

Some samples of genuine fingers has shown great variance in colors. On Figure 9.2, there are three images with dry, wet and strangulated finger. Strangulated sample is brighter, this is probably caused by the age of the participant (69 years old woman), such vast change in brightness was not observed with younger users. Wet finger image shows greatly changed reflectance caused by excess of applied water. This can simulate heavy rain.



(a) Dry finger (b) Wet finger (c) Strangulated finger

Figure 9.2: Samples of finger with great color differences.

	predicted fake	predicted genuine
true fake	61%	39%
true genuine	2%	98%

Table 9.2: SVM trained on whole dataset.

	predicted fake	predicted genuine
true fake	72%	28%
true genuine	6%	94%

Table 9.3: SVM trained on dry fingers only.

	predicted fake	predicted genuine
true fake	50%	50%
true genuine	2%	98%

Table 9.4: SVM trained on dry and wet fingers.

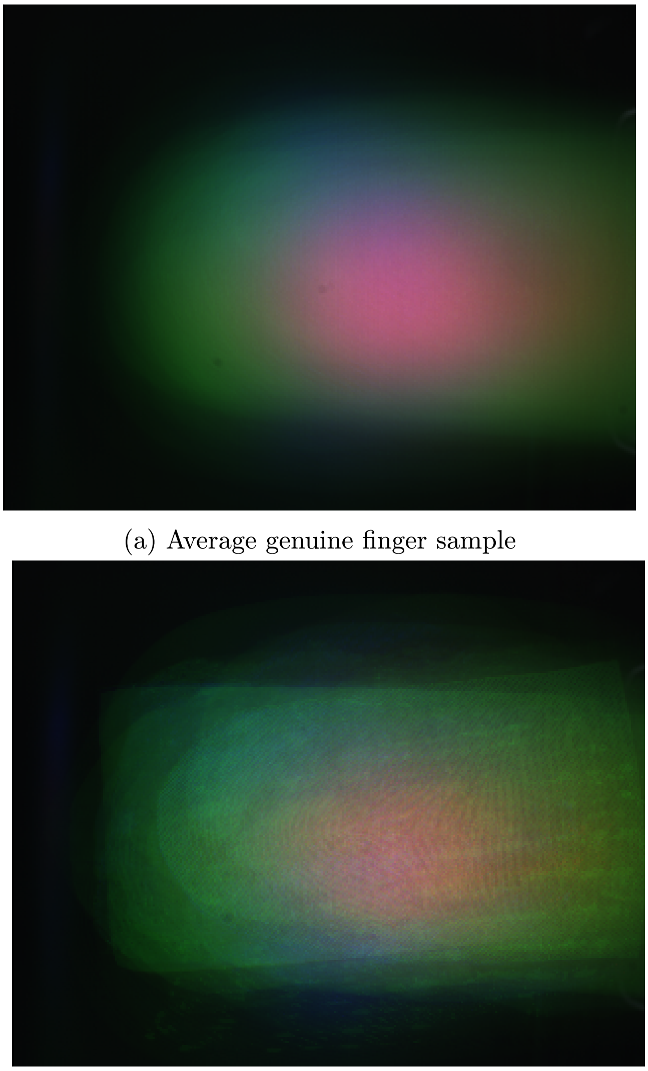
	predicted fake	predicted genuine
true fake	55%	44%
true genuine	3%	97%

Table 9.5: SVM trained on dry and strangulated fingers.

It is clear from Tables 9.2, 9.3, 9.4 and 9.5 that using only dry fingers to train the classifiers will produce the most accurate model. Drawback of this solution is that in cold weather (simulated by strangulated fingers) or rainy/hot weather (simulated by wet fingers) some users would be falsely rejected and usability of the system would be lowered. A test where the model is trained using only wet and strangulated fingers was not done, because it doesn't cover real use case.

9.3 Average Finger

To examine what is the characteristic of typical genuine finger and fake finger, two images with average samples were created, see Figure 9.3. The process used arithmetic mean per pixel and per channel. The images show what was the average illumination during the dataset creation.



(a) Average genuine finger sample

(b) Average fake finger sample

Figure 9.3: Comparison of images with averaged genuine and fake samples.

Images from Figure 9.3 were decomposed to red, green and blue channels, creating a six data series. These series were used to generate per-channel boxplot, see Figure 9.4. From this plot, it can be observed that on average, the color distribution is almost the same.

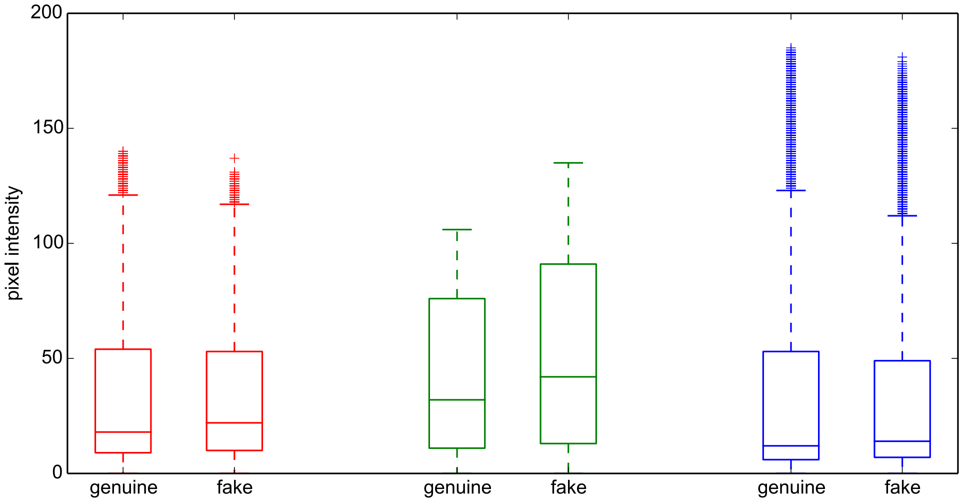


Figure 9.4: Box plot of averaged images of genuine and fake finger samples.

Chapter 10

Discussion and Future Work

Chapter 9 summarized performed experiments, some classifiers outperformed others and the best one is SVM with RBF kernel, however, recognition of fake fingers could be higher. Fake fingers alone are best classifed by AdaBoost and linear SVM with dataset preprocessed with RBM, but their recognition of genuine fingers was low. Further work with classifiers – focusing on their hyper parameters – could fully answer the question which classifier is suitable for production. It is possible that due to no free lunch theorem, the best classifier will change as the dataset grows.

The dataset consists of 114 samples (32 genuine fingers under 3 condition and 18 fakes). Increasing the number of samples in dataset will lead to more representative outcome. Moreover another conditions, such as dirt, scars and skin illness would reflect the real world usage better. The fake part of dataset could be improved by higher quality fakes with focus on color and scattering similar to human skin.

Restricted Boltzmann machine, mentioned in previous chapter, can be itself used for feature extraction, the input would be raw pixels and the output non-linearly extracted components. However, the training process of RBMs is time consuming and RBMs are highly sensitive to hyper parametrs. Properly tuned RBM is a challenge itself.

Chapter 11

Conclusion

This work dealt with liveness detection for the fingerprint recognition technology. Biometrics and biometric systems were described in chapter 2 along with basic terminology used to describe and measure biometric systems.

Chapter 3 focused on liveness detection of a finger, there were described approaches using perspiration, spectroscopic characteristics, ultrasonic technology, temperature and temperature stimulus, pressure stimulus, electrical properties, pulse and blood oxygenation, biochemical properties (odor) and image quality.

In chapter 4, spectroscopic approach was analysed and then a method for optical liveness detection was proposed: selection of wavelengths, image enhancement, determination of ROI and feature extraction.

Chapter 5 described which LEDs were used for illumination and what were the hardware adjustments. Then the chapter 6 covered implementation of image enhancement process and chapter 7 described feature extraction and how was the dataset created. Finally in in chapter 8, machine learning algorithms were described as well as their performance.

In chapter 9, results are summarized. It is stated, that best performance was provided by support vector machine classifier with RBF kernel. It is discussed how the accuracy changes when the dataset is reduced. At the end of the chapter, it is shown that on average, there are only little differences of color distribution between genuine and fake fingers. The results of this work are discussed in chapter 10.

This work showed that liveness detection using spectroscopic characteristic of finger is possible with only small hardware changes. Further improvement of classifiers and dataset could lead to real world usage. This work can be also combined with other approaches, such as the one that is shown by Tomáš Dohnálek [25], whose work focuses on liveness detection using infrared light to detect vein of genuine fingers. His work introduces hardware changes in similar scope as this one. Combining these two approaches would produce biometric system with robust liveness detection.

Bibliography

[1] Apple Inc. Press Info, "Apple announces iphone 5s—the most forward-thinking smartphone in the world." http://www.apple.com/pr/library/2013/09/10Apple-Announces-iPhone-5s-The-Most-Forward-Thinking-Smartphone-in-the-World.html. [Online; Accessed: 2015-01-09].

[2] D. Kirkland, "Fingerprints are usernames, not passwords." http://blog.dustinkirkland.com/2013/10/fingerprints-are-user-names-not.html, October 2013. [Online; Accessed: 2015-01-10].

[3] M. Drahanský, "Biometric systems – study text for course biometric system." Brno University of Technology, Faculty of Information Technology, 2006.

[4] M. Drahanský, "Biometric systems – lecture slides for course biometric system." Brno University of Technology, Faculty of Information Technology.

[5] D. Lodrová, *Security of Biometric Systems.* PhD thesis, Brno University of Technology, Faculty of Information Technology, 2013.

[6] Idiap Research Institute, "The Bob Project Documentation – Performance Evaluation." https://www.idiap.ch/software/bob/docs/releases/last/sphinx/html/TutorialsPerformance.html. [Online; Accessed: 2015-01-09].

[7] R. Houser, "Využití daktyloskopie v kriminalistické praxi," bachelor's thesis, Masaryk University Brno, Faculty of Law, 2013.

[8] M. Kluz, "Liveness Testing in Biometric Systems," master's thesis, Masaryk University Brno, Faculty of Informatics, 2005.

[9] M. Drahansky, *Liveness Detection in Biometrics.* pp. 179–198, Rijeka: InTech – Open Access Publisher, ISBN 978-953-307-487-0, 2011.

[10] S. Shuckers, L. Hornak, T. Norman, R. Derakhshani, S. Parthnasardi, "Issues for liveness detection in biometrics." West Virginia University: http://www.biometrics.org/bc2002/2_bc0130_DerakhshabiBrief.pdf, 2006. [Online; Accessed: 2015-01-11].

[11] M. Drahansky, M. Dolezel, J. Vana, E. Brezinova, J. Yim, and K. Shim, "New optical methods for liveness detection on fingers," *BioMed research international*, vol. 2013, article ID 197925, ISSN 2314-6141, 2013.

[12] R. K. Rowe, K. A. Nixon, and P. W. Butler, "Multispectral fingerprint image acquisition," in *Advances in biometrics*, pp. 3–23, Springer, 2008.

[13] D. Hejtmankova (Lodrova), “Issues for liveness detection in biometrics.” Gjovik University College, Brno University of Technology: https://www.fit.vutbr.cz/study/courses/BIO/private/Metody_testovani_zivosti.pdf, 2009. [Online; Accessed: 2015-01-09].

[14] M. Drahansky and D. Lodrová, “Liveness detection for biometric systems based on papillary lines,” *International Journal of Security and Its Applications*, vol. 2, no. 4, pp. 29–37, 2008.

[15] J. Galbally, F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia, “Fingerprint liveness detection based on quality measures,” in *Biometrics, Identity and Security (BIdS), 2009 International Conference on*, pp. 1–8, IEEE, 2009.

[16] Touchless Biometric Systems, “3D Enroll – datasheet.” http://www.tbs-biometrics.com/fileadmin/tbs-media/products/3D-Enroll/en_productsheet_3d-enroll.pdf. [Online; Accessed: 2015-01-07].

[17] J. Harman, “Using decorrelation stretch to enhance rock art images.” http://www.dstretch.com/AlgorithmDescription.html. [Online; Accessed: 2015-01-08].

[18] Robert K. Rowe, “A multispectral sensor for fingerprint spoof detection.” http://archive.sensorsmag.com/articles/0105/25/main.shtml, 2005. [Online; Accessed: 2015-01-08].

[19] Pedregosa, F. and Varoquaux, G. and Gramfort, A., “User guide: contents – scikit-learn 0.16.1 documentation.” http://scikit-learn.org/stable/user_guide.html. [Online; Accessed 2015-05-20].

[20] E. Angelopoulou, “The reflectance spectrum of human skin,” *Technical Reports (CIS)*, p. 584, 1999.

[21] OpenCV developer team, “OpenCV API Reference – OpenCV 2.4.11.0 documentation.” http://docs.opencv.org/2.4.11/modules/refman.html. [Online; Accessed 2015-05-19].

[22] H.-J. Bartsch, *Matematické vzorce.* Academia, ISBN 80-200-1448-9, 2006.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] G. Bradski, “The opencv library,” *Doctor Dobbs Journal*, vol. 25, no. 11, pp. 120–126, 2000.

[25] T. Dohnálek, “Liveness Detection on Fingers Using Vein Pattern,” master’s thesis, Brno University of Technology, Faculty of Information Technology, 2015.

Appendix A

CD Content

- **src/**: Python source code
- **tex/**: LaTeX source of this document
- **finger-samples/**: Dataset
- **xbrabe09-dip.pdf**: pdf file of this document

Appendix B

Program Usage

Global variable PATH must be set correctly in the project.py file, it has to point to folder with dataset. Python 2.7 must be installed along with libraries:

- scikit-learn
- NumPy
- SciPy
- Matplotlib

Run with no arguments to print simple help:

```
./project.py
no parameter provided
select one:
   dump         to extract features into python pickle
   eval         whole dataset evaluation
   evalpart     dataset evaluated per parts
   plot2D       to plot and save 2D representation of dataset
   ROC          to generate and save plot with ROC curves
   meanfingers  to generate average genuine and fake finger and save them
```