



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

DETEKCE ŽIVOSTI OTISKU PRSTU NA BEZDOTYKOVÉM ZAŘÍZENÍ

LIVENESS DETECTION ON TOUCHLESS FINGERPRINT SCANNER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

KATEŘINA FOŘTOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

MONA HEIDARI

BRNO 2020

Zadání bakalářské práce



Studentka: **Fořtová Kateřina**
Program: Informační technologie
Název: **Detekce živosti otisku prstu na bezdotykovém zařízení**
Liveness Detection on Touchless Fingerprint Scanner
Kategorie: Zpracování obrazu

Zadání:

1. Nastudujte literaturu v oblasti biometrie a zejména se zaměřte na falšování otisků prstů a detekci živosti.
2. Navrhněte algoritmus pro detekci živosti otisku prstu.
3. Algoritmus implementujte.
4. Algoritmus otestujte na poskytnutém datasetu.
5. Poznatky shrňte a zhodnoťte. Diskutujte možná pokračování a rozšíření vaší práce.

Literatura:

- Maltoni, D., Maio, D., Jain, A.K. and Prabhakar, S.: *Handbook of Fingerprint Recognition*. Springer, 2009, pages 512. ISBN 978-1-8488-2254-2.
- Kanich, O.: *Fingerprint Damage Simulation - A Simulation of Fingerprint Distortion, Damaged Sensor, Pressure and Moisture*, LAP LAMBERT Academic Publishing GmbH & Co. KG, 2014, p. 57. ISBN 978-3-659-63942-5.

Pro udělení zápočtu za první semestr je požadováno:

- Bod 1 a částečně body 2 a 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Heidari Mona**
Konzultant: Sakin Martin, Ing., UITS FIT VUT
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 28. května 2020
Datum schválení: 31. října 2019

Abstrakt

Tato bakalářská práce je zaměřena na detekci živosti otisků prstů s využitím bezdotykového senzoru. Shrnuje teoretický úvod do biometrie, zpracování otisků prstů a některé ze současných přístupů pro detekci živosti. Představuje nový přístup, který využívá algoritmus lokálního binárního vzoru, Sobelův a Laplaceův operátor a vlnkovou transformaci. Následná klasifikace byla provedena s využitím umělých neuronových sítí, metody podpůrných vektorů SVM a rozhodovacích stromů. Experimenty byly provedeny s datasetem nasvíceným světly o různé vlnové délce. Bylo zjištěno, že otisky prstů nasvícené červeným světlem vykazují nejlepší přesnost 90.1 % ze všech uvažovaných vlnových délek viditelného světla. Klasifikace s využitím vektoru na základě lokálního binárního vzoru dosahovala průměrné přesnosti 89.8 %, přesnost s užitým vektorem na základě Sobelova a Laplaceova operátoru byla 91.5 %. Pro vlnkovou transformaci byly využity různé Wavelet rodiny. Největší přesnosti dosahovaly vlnky z rodiny biortogonálních spline vlnek (85.1 %) a z rodiny reverzních biortogonálních spline vlnek (86.6 %).

Abstract

This Bachelor's Thesis is focused on liveness detection of fingerprints with using touchless sensor. Work summarizes theoretical introduction to biometrics, fingerprint processing and some of present researches for liveness detection. The new approach is introduced with using Local Binary Pattern algorithm, Sobel and Laplacian operator and Wavelet transform. Artificial Neural Networks, Support Vector Machines and Decision Trees were used for final classification. Several experiments with dataset illuminated by lights with various wavelengths were realized. It was discovered, that fingerprints illuminated by red light reached the best accuracy 90.1 % compared to other considered wavelengths of visible light. The classification with vector based on Local Binary Pattern achieved average accuracy 89.8 %, accuracy with vector based on Sobel and Laplacian operator was 91.5 %. Several Wavelet families were used for Wavelet transform during experiments. The best accuracy achieved wavelets of Biorthogonal spline wavelet family (85.1 %) and wavelets from Reverse biorthogonal spline wavelet family (86.6 %).

Klíčová slova

biometrie, detekce živosti, lokální binární vzor, zpracování obrazu, bezdotykové zařízení, vlnková transformace, otisky prstů

Keywords

biometrics, liveness detection, local binary pattern, image processing, touchless scanner, wavelet transform, fingerprints

Citace

FOŘTOVÁ, Kateřina. *Detekce živosti otisku prstu na bezdotykovém zařízení*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Mona Heidari

Detekce živosti otisku prstu na bezdotykovém zařízení

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením vedoucí práce Mony Heidari a konzultanta Ing. Martina Sakina. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....

Kateřina Fořtová

25. května 2020

Poděkování

Tímto bych chtěla poděkovat své vedoucí práce Moně Heidary za odbornou pomoc, rady a celkové vedení bakalářské práce k získání co nejlepších výsledků ohledně detekce živosti. Dále moje poděkování patří konzultantovi Ing. Martinovi Sakinovi, který mi pomáhal s korekturou samotného českého textu práce. Velké díky si zaslouží i všichni přátelé, kteří mi byli ochotni poskytnout své otisky prstů pro samotný dataset nezbytný pro experimenty.

Obsah

1	Úvod	3
2	Biometrie	4
2.1	Identita, identifikace, verifikace a autentizace	4
2.2	Biometrické systémy a vlastnosti	5
2.3	Skóre porovnání	6
3	Detekce otisků prstů	8
3.1	Historie	8
3.2	Zákony daktyloskopie	8
3.3	Struktura otisku prstu	9
3.4	Typy senzorů	9
3.5	Základní druhy markantů	11
3.6	Třídy otisků prstů	12
3.7	Falšování otisků prstů	13
4	Průzkum současných řešení detekce živosti otisků prstů	14
4.1	Hardwarové metody	14
4.2	Softwarové metody	15
5	Algoritmy pro předzpracování a analýzu	18
5.1	Převedení na normalizovaný obraz	18
5.2	Prahování	19
5.3	Segmentace	20
5.4	Lokální binární vzor (LBP)	21
5.5	Detektory hran Sobelův a Laplaceův operátor	22
5.6	Vlnková transformace	23
5.6.1	Rodičovské a dceřiné vlnky	24
5.6.2	Spojité vlnková transformace	24
5.6.3	Diskrétní vlnková transformace	24
5.6.4	Wavelet rodiny vhodné pro analýzu obrazu	25
5.6.5	Použité vlnky pro testování algoritmu	27
5.7	Matice souvislostí stupňů šedi v obraze (GLCM)	27
6	Rozhodovací algoritmy živosti	29
6.1	Umělá neuronová síť	29
6.2	Algoritmy podpůrných vektorů	31
6.3	Rozhodovací stromy	32

7 Implementace	34
7.1 Popis bezdotykového senzoru	34
7.2 Sběr datasetu	35
7.3 Knihovna OpenCV	36
7.4 Knihovny scikit-image a scikit-learn	36
7.5 Knihovna PyWavelets	37
7.6 Extrahované vektory z obrazu	37
7.6.1 Vektor založený na lokálním binárním vzoru	37
7.6.2 Vektor založený na Sobelově a Laplaceově operátoru	38
7.6.3 Vektor založený na vlnkové transformaci	38
7.7 Práce s implementovaným programem	38
7.7.1 Struktura programu	39
8 Experimenty a vyhodnocení	40
8.1 Analýza datasetů otisků prstů nasvícených stejnou barvou světla	40
8.2 Výsledky analýzy datasetu obsahující všechny získané snímky otisků prstů .	43
8.3 Vyhodnocení experimentů	45
9 Závěr	47
Literatura	48
A Obsah přiloženého paměťového média	52
B Manuál	53
C Výsledky analýzy datasetů otisků prstů nasvícených stejnou barvou světla	57
D Výsledky analýzy datasetu obsahující všechny získané snímky otisků prstů	60

Kapitola 1

Úvod

S rozvojem moderních technologií a komunikačních zařízení se stále více aktuální stává otázka bezpečnosti a ochrany dat. Zabezpečování údajů pomocí biometrie si získalo velký úspěch, zvláště díky snadnému přihlášení do systému a větší míře bezpečnosti, jako například u hesel nebo různých několikanásobných kódů. Není zde totiž potřeba si pamatovat údaje, stačí pouze použít svůj vlastní biometrický identifikátor, který bývá i bezpečnějším způsobem přihlášení nebo ochrany soukromých údajů.

Mnoho systémů dnes využívá zabezpečení pomocí otisku prstu. Přestože tento druh identifikace se nám může jevit jako opravdu spolehlivý, i tato metoda může být napadena. Zfalšované otisky prstů mohou být vyrobeny z různých syntetických materiálů, které nemusí být nijak moc nákladné. Proto je snahou vytvořit systém, který by mohl původ otisků prstů nějakým způsobem rozlišit a odhalit, zda je systém narušený útočníkem, nebo se jedná o živý otisk prstu.

Ve své práci se zaměřuji na implementaci kombinace různých algoritmů pro detekci živosti otisků prstů. Jedná se o algoritmy, které se zaměřují na zpracování obrazu, úpravu obrazu a výzkum, jak se dané algoritmy chovají u snímků živých a falešných otisků. Hlavním bodem analýzy je textura otisku prstu. Pro svůj výzkum jsem využila bezdotykový senzor, který poskytuje mnoho výhod ve srovnání s klasickými dotykovými přístroji. Na daném senzoru probíhal sběr datasetu živých a falešných otisků prstů. Živé otisky mi poskytlo několik osob a falešné otisky byly vyrobeny z různých materiálů poskytovaných naší fakultou. Pro práci s bezdotykovým senzorem bylo využito osvětlování prstů světlem o různé vlnové délce. Na souborech získaných ze senzoru jsem poté testovala implementované algoritmy, zkoumala jejich chování a vyhodnocovala závěry.

Výsledná analýza se sestávala z porovnání metod analýzy textury obrazu, analýzy hran a využití různých metod vlnkové transformace. Klasifikace probíhala za pomoci umělých neuronových sítí, metody podpůrných vektorů SVM a rozhodovacích stromů. Součástí mé práce je porovnání jednotlivých klasifikačních metod. Dále jsem zkoumala, jaké světlo vlnové délky viditelného spektra (červené, zelené nebo modré) vykazovalo největší přesnost a spolehlivost při mém výzkumu.

Kapitola 2

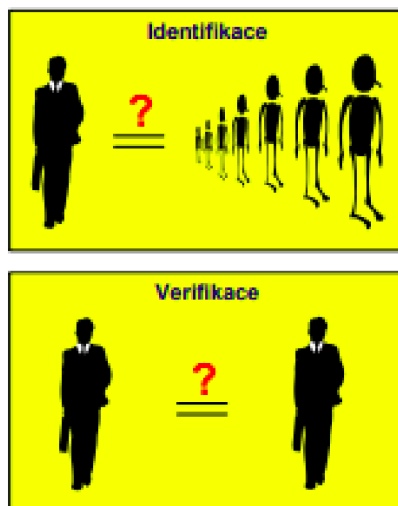
Biometrie

Biometrie je věda rozpoznávání identity člověka na základě jeho fyzických a behaviorálních vlastností, jako například tvář, otisky prstů, hlas nebo oční duhovka [17]. Slovo biometrie pochází z řeckých slov *bios* (život) a *metron* (měření). Biometrické identifikátory jsou v informatice používány pro přístupovou kontrolu a identifikaci. Skoro všechny identifikátory jsou kombinací anatomických a behaviorálních charakteristik. U otisku prstu je behaviorální složkou skutečnost, že každý uživatel použije skener otisku prstů v závislosti na svém chování. Záleží i například na úhlu přiložení prstu nebo znečištění povrchu prstu či senzoru. U identifikace tváře pak záleží i na změnách pramenících z chování člověka (např. změna životního stylu vedoucí ke změně tělesné váhy) [21].

2.1 Identita, identifikace, verifikace a autentizace

Identifikace, verifikace, identita a autentizace jsou základními čtyřmi pojmy, které biometrické systémy využívají a pracují s nimi [8].

- **identita** – Je jednoznačnou charakteristikou jedince. Rozlišujeme ji dále na:
 - **fyzická identita** – Tato identita je pouze jedna jediná. Je definována naším vzhledem a chováním.
 - **elektronická identita** – Těchto identit můžeme mít vytvořených nespočetně mnoho díky například více různým účtům na webové stránce.
- **identifikace** – Slouží pro zjištění identity osoby. Osoba předá systému svoji biometrickou vlastnost a ten musí na základě jejího vyhodnocení rozhodnout, zda je identita nalezená, nebo nenalezená. Systém k tomuto účelu používá porovnávání s databází vzorků. Jedná se o porovnání 1:N, protože se biometrická vlastnost porovnává s celou databází.
- **verifikace** – Uživatel sdělí systému svoji elektronickou identitu. V systému je pak potřeba ověřit skutečnou fyzickou identitu uživatele. Proto se hledá, zdali jeho záznam obsahuje biometrická data. Dále se porovnává, zda sobě data odpovídají. Na základě tohoto údaje systém rozhodne o potvrzení či nepotvrzení identity. Jedná se o porovnání 1:1, dochází k porovnání vstupních dat pouze s jedním záznamem v databázi.
- **autentizace** – Systém při tomto úkonu potvrzuje autentičnost (hodnověrnost) dané osoby. Může být využita při identifikaci a verifikaci. Porovnávání většinou funguje na základě pomyslného prahu.

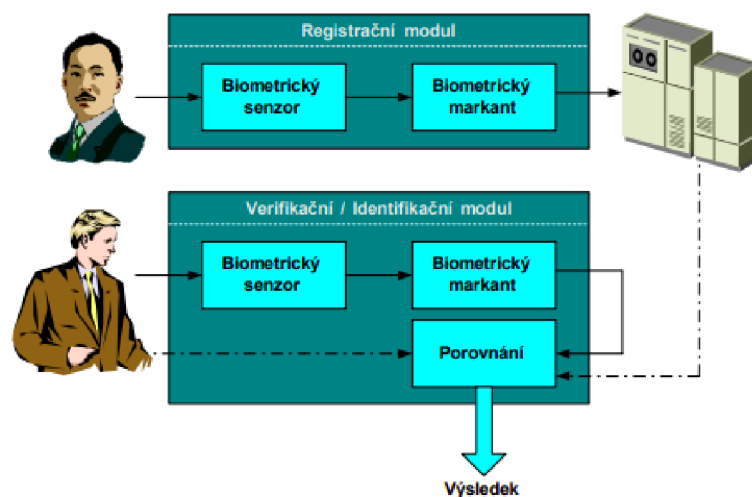


Obrázek 2.1: Rozdíl mezi identifikací a verifikací [8]

2.2 Biometrické systémy a vlastnosti

Biometrické systémy dělíme do dvou základních kategorií: [8]

- **registrační modul** – Zde je zaregistrována biometrická informace (tzv. biometrický markant) a je uložena do databáze.
- **verifikační modul** – Zde je zaregistrována biometrická informace podobně jako u registračního modulu, ale výsledek není uložen do databáze, nýbrž jsou data z databáze načítána, aby se extrahovaný biometrický markant mohl porovnat s daty v databázi.



Obrázek 2.2: Znázornění biometrického systému [8]

Biometrické vlastnosti (rysy) můžeme dělit do dvou kategorií – statické a dynamické vlastnosti. Statické vlastnosti se zabývají samotnou analýzou anatomických rysů (např. otisk prstu), jedná se o vždy přítomnou biometrickou vlastnost, která není lehce ovlivnitelná různými okolnostmi. Dynamické vlastnosti jsou pak spojeny s nějakou akcí daného uživatele. Níže jsou vypsáni zástupci obou kategorií:

- anatomické (fyzické statické) vlastnosti – otisk prstu, obličej, duhovka oka, sítnice oka, geometrie ruky, dlaň, termogram obličeje, termogram ruky, dentální obraz, podpis, tvar ucha, snímek nehtu, DNA,
- dynamické vlastnosti (chování nebo jednání) – hlas (řeč), gestikulace obličeje, podpis (dynamické vlastnosti), dynamika stisku kláves, pohyby rtů, chůze.

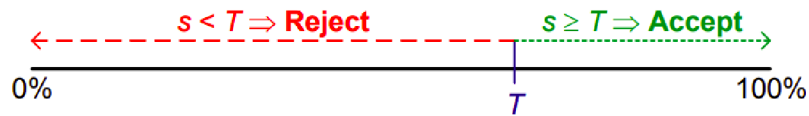
Biometrický systém můžeme právě dle daných biometrických vlastností dělit na unimodální biometrický systém a multimodální biometrický systém. Unimodální biometrický systém využívá pouze jednu danou biometrickou vlastnost. Multimodální biometrický systém pak využívá více biometrických vlastností nebo statické i dynamické příznaky pro jednu biometrickou vlastnost (např. statické a dynamické vlastnosti podpisu). Multimodální systémy jsou spolehlivější pro analýzu, avšak mají dražší pořizovací náklady než unimodální systémy [8].

Důležitými kritérii pro rozhodnutí o použití daného biometrického systému jsou atributy biometrických vlastností. Mezi nejdůležitější atributy patří:

- univerzalita – biometrickou vlastnost by měla mít každá osoba,
- jedinečnost – biometrická vlastnost je unikátní pro daného člověka,
- konstantnost – biometrická vlastnost se nemění v čase,
- získatelnost – biometrickou vlastnost můžeme kvantitativně změřit,
- výkonnost – biometrická vlastnost se nesmí změnit,
- akceptace – souhlas s nasnímáním dané biometrické vlastnosti danými subjekty,
- bezpečnost proti falšování – míra obtížnosti vytvoření falzifikátu pro danou biometrickou vlastnost,
- finanční náklady na pořízení – míra finančních nákladů potřebných k pořízení biometrického systému [8].

2.3 Skóre porovnání

Skóre porovnání udává kvantifikovanou podobnost extrahovaného vzorku a šablony. Extrahovaný vzorek obsahuje významné rysy ze vstupních dat. Šablona je pak uložena např. v databázi nebo na čipové kartě. Tato míra shody je založena na aplikaci prahu T , na základě kterého je rozhodnuto o přijetí nebo nepřijetí daného objektu [8].



Obrázek 2.3: Skóre porovnání na základě zvoleného prahu [8]

Míra chybného přijetí (FAR)

FAR (*False Acceptance Rate*) udává procentuální zastoupení identifikovaných instancí, kdy jsou neautorizované osoby akceptovány daným biometrickým systémem. Tato vlastnost je spočítána následujícím vzorcem:

$$FAR = \frac{\text{pocet porovnaní rozdílných vzorů vedoucích ke shodě}}{\text{celkový počet porovnaní rozdílných vzorů}} \cdot 100\% \quad (2.1)$$

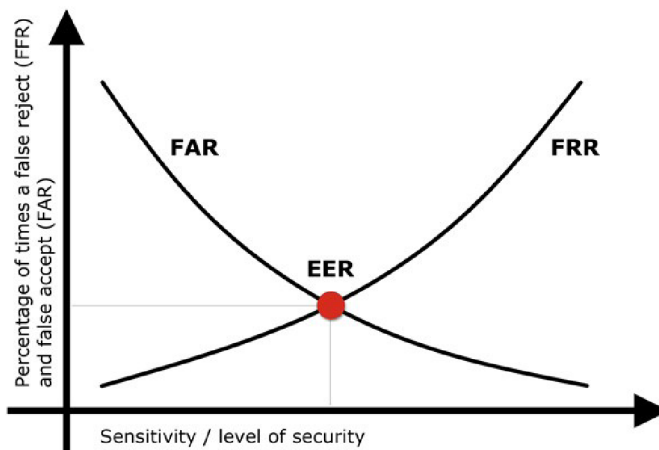
Míra chybného odmítnutí (FRR)

FRR (*False Rejection Rate*) představuje procentuální zastoupení identifikovaných instancí, kdy jsou autorizované osoby nekorektně biometrickým systémem zamítnuty. Výpočet je obdobný jako u charakteristiky FAR.

$$FRR = \frac{\text{pocet porovnaní shodných vzorů vedoucích k neshodě}}{\text{celkový počet porovnaní shodných vzorů}} \cdot 100\% \quad (2.2)$$

Souvislost FAR a FRR

Vliv charakteristik FAR a FRR znázorňuje následující graf. S větší úrovní zabezpečení se snižuje hodnota FAR, tedy je přijato stále méně a méně neautorizovaných osob. Naopak se zvyšuje riziko zamítnutí autorizovaných osob daným systémem. Průsečík těchto dvou křivek nese zkratku EER (*Equal Error Rate*) neboli Míra vyrovnání chyb. V tomto bodě bude chybně akceptován a chybně odmítnut stejný počet osob. Můžeme mít systém, který je více bezpečný, ale méně uživatelsky přívětivý nebo obráceně [34].



Obrázek 2.4: Souvislost charakteristik FAR a FRR [34]

Kapitola 3

Detekce otisků prstů

Následující kapitola se zabývá základními informacemi o otiscích prstů. Po stručném shrnutí historie analýzy otisků prstů je tato kapitola dále zaměřena například na papilární linie, různé důležité markanty v otisku prstu nebo na typy senzorů, které se používají pro jejich snímání.

3.1 Historie

Lidské otisky prstů byly objeveny na velkém počtu archeologických artefaktů a historických předmětů. I když tyto nálezy dokazují, že lidé si v té době byli vědomi unikátnosti otisků prstů, až v 16. století byla objevena první vědecká technika v jejich výzkumu. V roce 1864 anglický vědec Nehemiah Grew publikoval první vědeckou studii o hřebenech, rýhách a pórech v struktuře otisku prstu.

První detailní popis struktury otisků prstů byl předveden německým anatomistou Maye-rem v roce 1788. Jan Evangelista Purkyně pak roku 1823 vytvořil první klasifikační schéma, ve kterém otisky prstů rozdělil do devíti kategorií na základě konfigurace hřebenů [21].

3.2 Zákony daktyloskopie

Daktyloskopie je naukou o obrazcích papilárních linií. S touto vědou se pojí samotná identifikace daktyloskopických stop a osob [30]. Byly zavedeny tyto daktyloskopické zákony:

- struktura papilárních linií je unikátní pro každého jedince,
- vzor tvořený papilárními liniemi je pro každého jedince během života relativně neměnný,
- obnova papilárních linií probíhá dorůstáním kůže na povrchu prstů, papilární linie mohou být pozmeněny, pouze pokud se poškodí nebo odstraní epidermální vrstva kůže a poté již nemůže docházet k obnově,
- konfigurační typy se mohou individuálně měnit, jedná se však o malé změny, které leží v tolerančních limitech a umožňují tak systematickou klasifikaci [8].

3.3 Struktura otisku prstu

Každý otisk prstu je tvořen vzorem papilárních linií. Výška papilárních linií se pohybuje v rozmezí 0,1 – 0,4 mm a šířka v rozmezí 0,2 – 0,5 mm [8]. Průběhy papilárních linií jsou jedinečné pro každého člověka. S přibývajícím věkem se mění rozměry plošek prstů či dlaní, avšak struktura papilárních linií zůstává stejná.

Nejvýznamnější atributy papilárních linií se vyvíjí již v nitroděložní části života. Konečná podoba papilárních linií je u plodu již v 6. – 7. měsíci nitroděložního vývoje [9].

Otisky prstů dělíme na tři základní druhy:

- válený otisk (barvený, rolovaný),
- píchaný otisk (živý),
- latentní otisk (skrytý) [8].



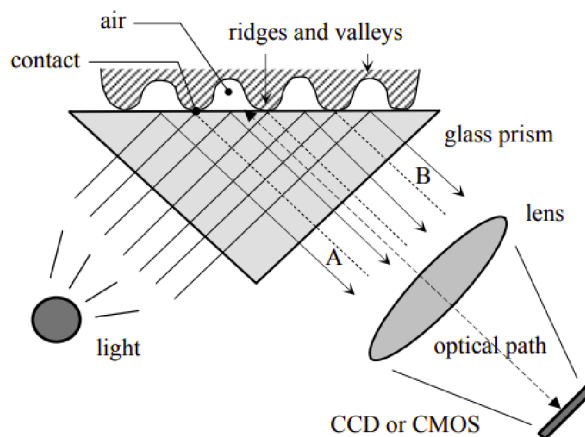
Obrázek 3.1: Válený, píchaný a latentní otisk prstu [8]

3.4 Typy senzorů

Nejdůležitější částí detekce je senzor, který zaznamenává výsledný otisk prstu. Sensory řadíme do těchto základních kategorií:

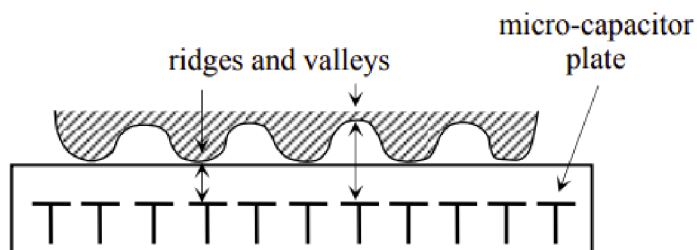
- **tlakový senzor** – Je vytvořen z materiálu citlivého na mechanické namáhání, při kterém generuje elektrický signál. Velikost elektrického signálu závisí na tlaku, který vyvíjí prst na povrch senzoru. Hřebeny a údolí papilárních linií jsou v jiných vzdálenostech od povrchu snímače, proto produkují jinou hodnotu elektrického signálu [21].
- **termický senzor** – Senzor využívá pyroelektrický materiál, který generuje elektrický náboj na základě teplotních rozdílů. Je založen na skutečnosti, že hřebeny v kontaktu s povrchem senzoru produkují jinou teplotu než údolí, která jsou vzdálenější od povrchu snímače. Senzor je obvykle přehřán na vyšší teplotu, aby se zvýšila odlišnost mezi povrchem zařízení a hřebeny linií [21].
- **optické senzory** – Využívají jednoduchý zdroj světla (LED), který osvětlí plochu prstu [8]. Nejvíce používaným zástupcem optických senzorů je infračervený spektrometr s Fourierovou transformací (FTIR). Prst se při snímání dotýká skleněného nebo plastového hranolu. Při dotyku jsou hřebeny papilárních linií v kontaktu s povrchem snímače, ale údolí papilárních linií jsou vzdálena. Světlo pocházející z hranolu dopadá na prst a je odraženo údolími papilárních linií nebo náhodně pohlceno hřebeny. Ve výsledném obrazu jsou pak hřebeny tmavá místa a údolí světlá místa. FTIR snímače

dokážou snímat 3D otisk prstu, proto mohou detekovat i zfalšované otisky prstu, které například pocházejí z fotografie [21].



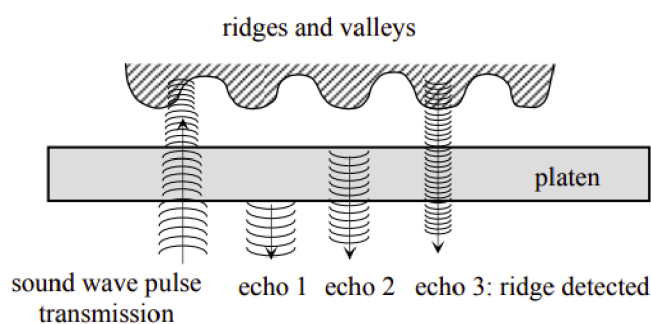
Obrázek 3.2: Senzor FTIR [21]

- **kapacitní senzory** – Jsou složeny z matice malých vodivých plošek, na níž je napařena vrstva nevodivého oxidu křemičitého [8]. Mezi povrchem prstu a každou z plošek (mikrokapacitorů) v čipu vzniká malý elektrický náboj. Při skládání výsledného otisku prstu je důležitá vzdálenost mezi povrchem otisku prstu a mikrokapacitorem. Hřebeny a údolí papilárních linií otisku prstu mají jinou vzdálenost od mikrokapacitoru, tudíž i odlišnou intenzitu ve výsledném obraze. Stejně jako u senzoru FTIR nemůže tato technologie být zneužita útočníkem při předložení fotografie. Jsou zde totiž měřeny vzdálenosti a pouze trojrozměrný povrch může být nasnímán [21].



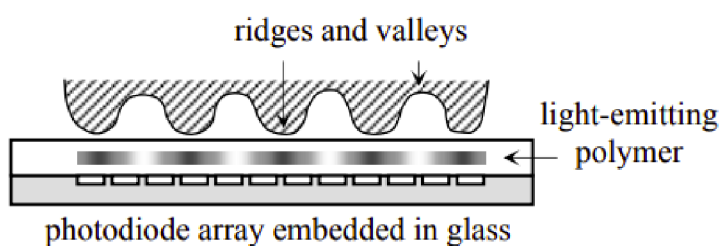
Obrázek 3.3: Kapacitní senzor [21]

- **ultrazvukové senzory** – Jsou založeny na posílání akustického signálu prstu a zachytávání odraženého signálu. Odražený signál je použit na výpočet hloubky obrazu a struktury papilárních linií. Senzor se skládá z vysílače a přijímače. Vysílač generuje krátké zvukové pulsy a přijímač detekuje odpověď po odražení těchto akustických pulsů od povrchu prstu. Senzor je schopný detekovat strukturu otisku prstu i například přes tenké rukavice, stejně jako si dokáže poradit s případnými nečistotami i mastnotou [21].



Obrázek 3.4: Ultrazvukový senzor [21]

- **elektrooptické senzory** – Tyto snímače jsou složeny ze dvou vrstev. První dokáže emitovat světlo, pokud je polarizováno správným napětím. Druhá vrstva úzce spolupracuje s první, zpracovává emitované světlo a vytváří finální digitální obraz [21].



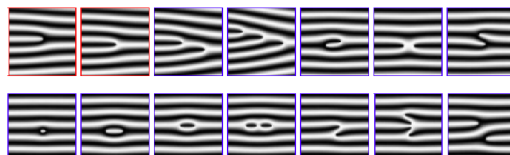
Obrázek 3.5: Elektrooptický senzor [21]

3.5 Základní druhy markantů

Otisk prstu obsahuje útvary, které tvoří papilární linie – markanty. Mezi důležité markanty patří:

- ukončení (*Line Ending*),
- jednoduchá vidlička (*Simple Bifurcation*),
- dvojitá vidlička (*Double Bifurcation*),
- trojitá vidlička (*Triple Bifurcation*),
- hák (*Hook*),
- křížení (*Crossing*),
- boční kontakt (*Side Contact*),
- bod (*Point*),
- interval (*Interval*),

- jednoduchá smyčka (*Single Whorl*),
- dvojitá smyčka (*Double Whorl*),
- jednoduchý most (*Single Bridge*),
- dvojitý most (*Twin Bridge*),
- průsečná linie (*Through Line*) [8].



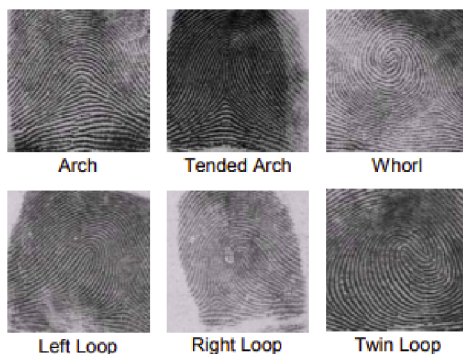
Obrázek 3.6: Základní typy markantů ve vyjmenovaném pořadí [8]

U přístupových systémů se však využívá pouze ukončení (*Line Ending*) a vidličky (*Bifurcation*) [8].

3.6 Třídy otisků prstů

Otisky prstů můžeme na základě charakteristických znaků rozdělit do následujících kategorií:

- oblouk (*Arch*),
- klenutý oblouk (*Tended Arch*),
- spirála / závit (*Whorl*),
- levá smyčka (*Left Loop*),
- pravá smyčka (*Right Loop*),
- dvojitá smyčka (*Twin Loop*) [8].



Obrázek 3.7: Třídy otisků prstů [8]

3.7 Falšování otisků prstů

Falešné biometrické reprezentace jsou obecně velkým nebezpečím, protože umožňují zločinci vydávat se za někoho jiného a tím narušit soukromí a bezpečnost jedince. Abnormální otisky prstů dělíme do dvou kategorií:

- falešné otisky prstů,
- pozměněné otisky prstů.

Falešný otisk prstu reprezentuje repliku skutečného otisku prstu vyrobeného z materiálů, jako je želatina, silikon nebo latex. Biometrické zařízení musí rozhodnout, zda otisk je živý, nebo falešný. Tato procedura se nazývá detekce živosti. Pozměněné otisky prstů jsou skutečné, avšak jejich struktura je změněna. Otisky mohou být neúplné, pokřivené. Důvodem poškození otisku mohou být nemoci kůže, např. pořezání, popálenina, poleptání silnými chemikáliemi, transplantace kůže nebo různé kožní nemoci [29].

Kapitola 4

Průzkum současných řešení detekce živosti otisků prstů

Algoritmů pro detekci živosti otisku prstu byla objevena ve studiích celá řada, jak již bylo avizováno v jedné z předchozích kapitol. Jedná se o práce, které využívají jak hardwarové metody analýzy, tak softwarové metody. Nevýhodou hardwarových metod je, že vyžadují zvláštní vybavení navíc. Tato kapitola popisuje stručně současné přístupy k analýze.

4.1 Hardwarové metody

Hardwarové metody využívají k určení živosti otisků prstů zvláštní hardwarová zařízení. Důležitým opatřením pro tyto metody je skutečnost, aby byla daná hardwarová zařízení integrována takovým způsobem, aby analýzu neohrozila kombinace útočnickova živého prstu s falešným. Zde jsou uvedeny některé z daných přístupů.

Pulsní oxyometrie

Tato metoda využívá měření obsahu kyslíku v krvi. Používá LED a fotodetektor na opačných stranách prstu. Měření závisí na srdečním tepu, a proto může trvat několik sekund, než jsou spočítány jeden nebo dva celé cykly srdečního tepu [20].

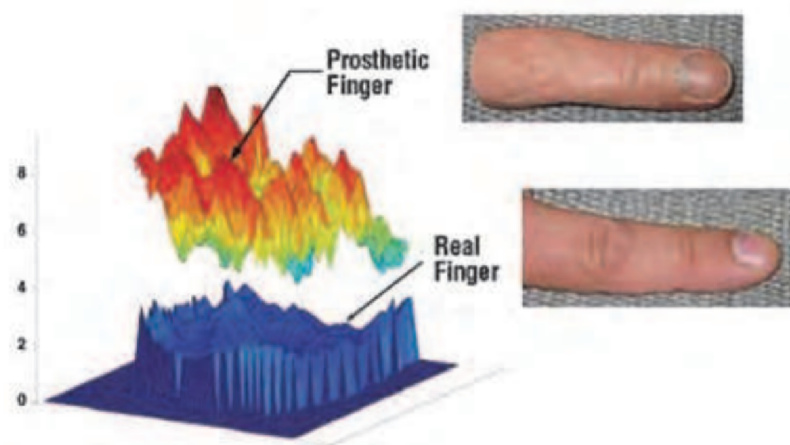
Teplý a studený podnět

Lidský prst dokáže odlišně reagovat na daný podnět, který je umístěný u senzoru. Výsledná informace je zaznamenaná, konkrétně se jedná o kolísání rychlosti průtoku krve v periferních cévách. Při teplém podnětu se amplituda rychlosti toku zvyšuje, při studeném podnětu pak snižuje. Tato metoda však může být ovlivněna citlivou lidskou periferní nervovou soustavou, kdy chvíli trvá, než správně zareaguje na teplý nebo studený podnět u daného člověka [10].

Multispektrální analýza

Kůže člověka má některé unikátní optické charakteristiky díky chemickému složení, které ovlivňuje například absorpci optických vlastností. Díky snímkům nasvíceným světlem o různé vlnové délce můžeme měřit různé vlastnosti lidské kůže a následně tedy odhalit případný falešný otisk prstu. Krátké vlnové délky prst absorbuje a dlouhé vlnové délky pak prst prosvítí. Pomocí kamery je měřeno odražené světlo od nasvíceného prstu. Měření může být

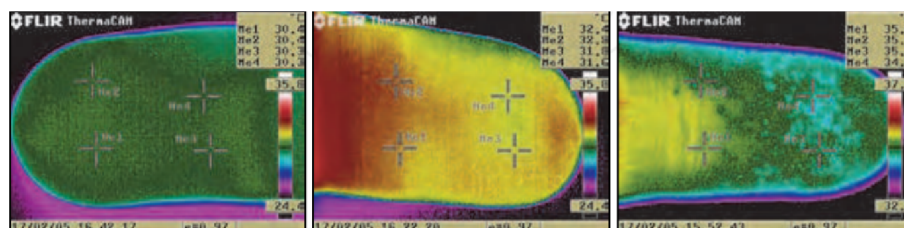
poté vizualizováno do grafu a na základě výsledného spektra a matematických metod je stanoven výsledek [10].



Obrázek 4.1: Rozdíl výsledku multispektrální analýzy mezi živým a falešným prstem [10]

Teplota

Teplota lidské kůže u prstu se pohybuje okolo 25 – 37 stupňů Celsia. Nicméně tento rozsah musí být širší, aby mohl systém pracovat pod různými podmínkami, i díky lidem, kteří mají problém s cirkulací krve. Takle skutečnost však může vést k chybným rozhodnutím při této metodě. Dalším možným řešením je zahřátí prstu. Použitím falešného otisku dosáhneme rozdílu zhruba dvou stupňů Celsia. Rozdíl je tedy malý a tato metoda není dostatečně spolehlivou pro bezpečnou detekci živosti otisků prstů [10].



Obrázek 4.2: Termoskeny prstů pomocí termokamery FLIR [10]

4.2 Softwarové metody

Tyto metody užívají k detekci živosti informace, které jsou již obsažené v obrazu otisku prstu. Jedná se například o znaky deformace kůže, detekci pórů a další. Výhodou je, že tyto metody nemusí využívat žádné přídavné hardwarové zařízení a mohou být tedy i levnější na samotné nasazení.

Deformace kůže a elasticita

Tyto techniky zkoumají, jak se deformuje lidská kůže, když je prst přitlačen na povrch senzoru. Bylo zjištěno, že živé prsty se na povrchu snímače chovají způsobem, kdy se u nich projevuje značné množství nelineárního zkreslení. Falešné prsty jsou více tuhé než kůže a deformace je nižší, dokonce i když jsou falešné otisky vyrobeny z vysoce elastických materiálů [20].

Detekce potu

I bříško prstu obsahuje potní žlázy, proto detekce pórů může posloužit pro analýzu živosti. Na základě uběhnutého času se změní změna velikosti pórů, kdy se vzor stává postupně tmavším. Důležitou statickou charakteristikou je pak variabilita odstínů šedi okolo hřebenů papilárních linií. Dynamickou vlastností je pak vývoj v čase při změně extrahovaného signálu z hřebenů. Falešné snímky pak nedokáží pořádně poskytnout tyto vlastnosti kvůli neschopnosti provést hlavně dané dynamické vlastnosti, které jsou u živých prstů běžné [27].



Obrázek 4.3: Srovnání vývoje detekce potu v čase mezi živým a falešným prstem [27]

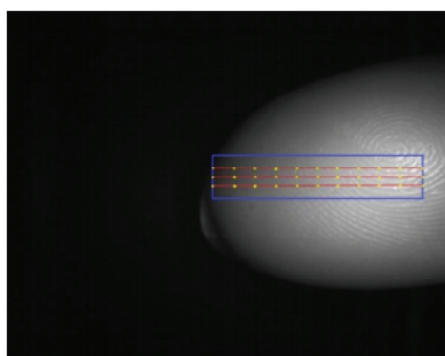
Využití detekce textury otisku prstu

Tato metoda byla představena s využitím bezdotykového senzoru. Extrahovaný vektor byl složen z 512 deskriptorů algoritmu vylepšeného lokálního binárního vzoru (*Improved Local Binary Pattern* (ILBP)), který vedle klasických hlavních vlastností LBP představuje další nové vlastnosti. Místo počítání binárního kódu je pracováno s průměrem, výsledné číslo není 8-bitové jako u algoritmu LBP, ale 9-bitové. Druhá složka vektoru je založena na matici GLCM (*Grey Level Cooccurrence Matrix*) a jejích čtyř vlastností (kontrast, korelace, energie a homogenita). Je nutno podotknout, že tyto vlastnosti jsou průměrem vlastností získaných z 8 GLCM matic, které jsou u každého pixelu spočítány pro každý směr. Dále je na celý vektor aplikována metoda *Principal Component Analysis* (PCA), která redukuje dimenzi příznakového prostoru. Pro finální trénování a testování byla použita umělá neuronová síť,

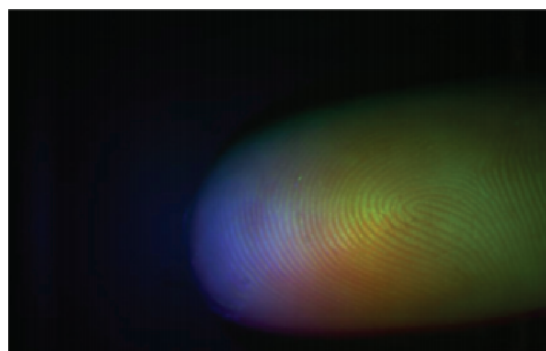
která předpověděla nejen živost a neživost daného otisku prstu, ale i materiál, ze kterého jsou umělé otisky prstu vyrobeny [43].

Nasvicování prstů světlem o různé vlnové délce

Jedná se opět o přístup, kdy bylo experimentováno se snímky s využitím bezdotykového senzoru. Prst byl nasvicován červeným, zeleným a modrým světlem. Na základě zkoumání byl vytvořen obdélník o fixní délce, ze kterého se analyzovalo daných obsažených 11 pixelů. Globální vlastnosti extrahované z celého obdélníku byly následující – aritmetický průměr intenzity pixelů, standardní odchylka intenzity pixelů, průměr, medián a standardní odchylka pro extrahovaný histogram. Dále byly snímky po nasvícení všemi třemi barvami světla spojeny do jediného obrázku a bylo experimentováno s různými barevnými modely, další extrahovanou vlastností se stal 3-kanálový histogram získaný z tohoto snímku. Pro následnou klasifikaci byla použita umělá neuronová síť a náhodný les [7].



Obrázek 4.4: Obdélník s 11 extrahovanými pixely [7]



Obrázek 4.5: Sloučené snímky nasvícené různými barvami světla (RGB model) [7]

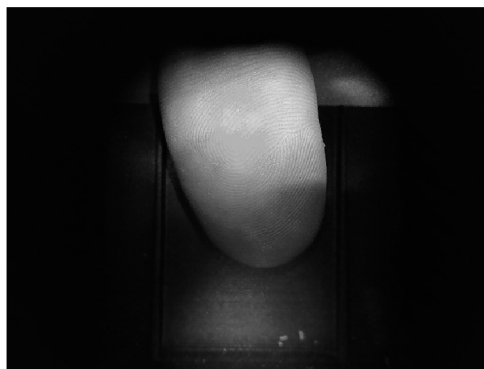
Kapitola 5

Algoritmy pro předzpracování a analýzu

Tato kapitola se zabývá popisem použitých algoritmů pro předzpracování obrazu a následně vybraných algoritmů pro samotnou detekci živosti otisků prstů. Pro hlavní analýzu bylo vybráno více algoritmů, aby pak bylo možné dané přístupy mezi sebou porovnat a stanovit ten nejlepší pro detekci živosti otisku prstu na užitém bezdotykovém senzoru. Samotná detekce se pak sestává z aplikace různých řešení a získání vlastností ze samotného obrazu otisku prstu, které jsou použitelné pro rozlišení živých a falešných vzorků.

5.1 Převedení na normalizovaný obraz

Po načtení obrazu bylo potřeba obraz převést na šedotónový. Poté byla provedena normalizace obrazu. Normalizace je proces, při kterém je upraven rozsah hodnot intenzity pixelů. Základem je zvýšit dynamický rozsah šedotónového jasu. U otisků prstů tento algoritmus nemění ostrost hřebenů a údolí papilárních linií. Hlavním účelem je minimalizovat změny šedoúrovňových hodnot podél hřebenů a údolí [15].



Obrázek 5.1: Šedotónový obraz živého otisku prstu



Obrázek 5.2: Normalizovaný obraz živého otisku prstu

5.2 Prahování

Při prahování jsou světlé objekty odděleny od tmavých pomocí zadaných prahů. Tento algoritmus je nutno použít před samotnou segmentací obrazu pomocí morfologických metod. Jednoduché prahování závisí na oddělení světlých a tmavých objektů podle daných předem zvolených konstantních prahů. Existují však i další metody prahování, které poskytují lepší analýzu a výsledky a při mém výzkumu byly použity. Jedná se o níže popsané metody.

Prahování pomocí metody Otsu

Při globálním prahování je nutné si vybrat konstantní práh již před analýzou obrazu. Metoda Otsu tento globální optimální práh vybírá automaticky, na základě daného vstupního obrazu [26]. Výpočet daného optimálního prahu zahrnuje iteraci přes všechny možné prahové hodnoty a výpočet rozložení pixelů na jednotlivých stranách prahu, pixelů, které spadají buď do popředí, nebo pozadí obrazu. Účelem je pak najít hodnotu prahu, kdy součet rozložení na popředí a pozadí je na svém minimu [13].

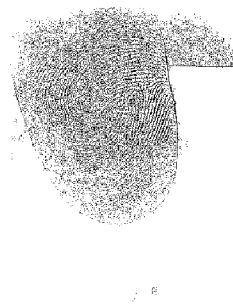
Adaptivní prahování

Adaptivní prahování slouží pro analýzu obrazu při situacích, kdy určité části obrazu mají jiné světelné podmínky, např. při výskytu stínů. Metoda nevyužívá konstantního prahu. Algoritmus určí pro pixel hodnotu prahu na základě okolí okolo něj. Výsledkem jsou jiné hodnoty prahů pro různé oblasti obrazu. Pro adaptivní prahování bylo využito následující metody: [26]

- Adaptivní průměrné prahování (*Adaptive Mean Thresholding*) – Hodnota prahu je průměrem okolí, které je dáno velikostí bloku. Dále je od tohoto výsledku odečtena konstanta C .
- Adaptivní prahování Gaussian (*Adaptive Gaussian Thresholding*) – Výslednou hodnotu prahu udává vážená suma okolních hodnot, kdy váhy jsou Gaussian oknem. I zde je poté odečtena konstanta C .



Obrázek 5.3: Prahování pomocí Otsu metody



Obrázek 5.4: Prahování pomocí Gaussian metody



Obrázek 5.5: Prahování pomocí Mean metody

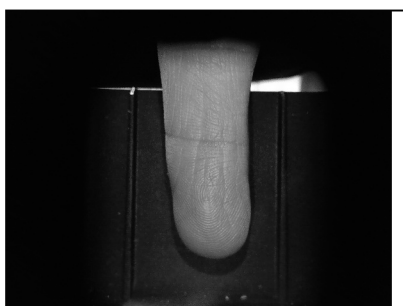
5.3 Segmentace

Při segmentaci je vybrán ze vstupního obrazu pouze otisk prstu, což napomáhá eliminovat pozadí obrazu. Bylo využito morfologického otevření – kombinace eroze a následné dilatace obrazu.

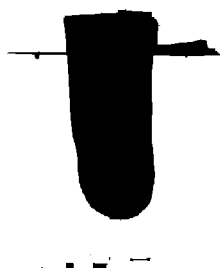
Morfologické operace se používají obvykle na úpravu binárního obrazu. Potřebují dva vstupy – daný vstupní obraz a jádro [25]. Tyto operace se používají často na obrazové předzpracování – odstraňují šum a zjednodušují tvary objektů. Eroze je morfologickou operací, která je užitečná pro odstranění malých bílých částí v obraze. Dilatace je opakem eroze a používá se k zaplnění mezer [16].

Tento algoritmus má za cíl vytvořit masku, která bude určovat pozici otisku prstu v obraze. Struktura objektu se tedy při vzniku masky zjednoduší. Výsledná maska je následně aplikována na normalizovaný obraz a je tedy vybrána část obrazu, která obsahuje otisk prstu, pozadí zůstává bílé.

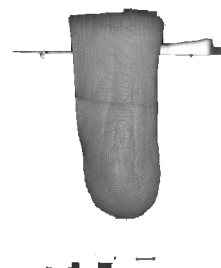
Získaná maska vzniká prahováním, pro které byly využity tři výše zmíněné metody – metoda Otsu a dále adaptivní prahovací metody Mean a Gaussian.



Obrázek 5.6: Vstupní normalizovaný obraz

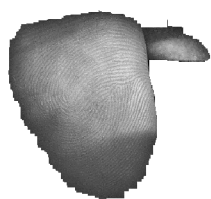


Obrázek 5.7: Získaná maska

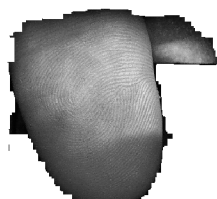


Obrázek 5.8: Výsledný segmentovaný obraz

Následující obrázky ukazují rozdílné výsledky segmentace s využitím různých prahovacích metod.



Obrázek 5.9: Segmentace při použití Otsu prahování



Obrázek 5.10: Segmentace při Gaussian prahování



Obrázek 5.11: Segmentace při použití Mean prahování

5.4 Lokální binární vzor (LBP)

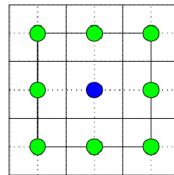
Lokální binární vzor (*Local Binary Pattern* – LBP) je jedním z hlavních algoritmů, které byly využity pro detekci živosti v této práci. Jedná se o efektivní algoritmus, který se používá pro analýzu textury objektů. Lokální binární vzor je užitečný pro informace o změnách intenzity mezi jednotlivými pixely obrázku a jejich sousedy. Vzorec pro tuto metodu vypadá následovně:

$$LBP(x_c, y_c) = \sum_{n=0}^7 2^n (I_n - I(x_c, y_c)), \quad (5.1)$$

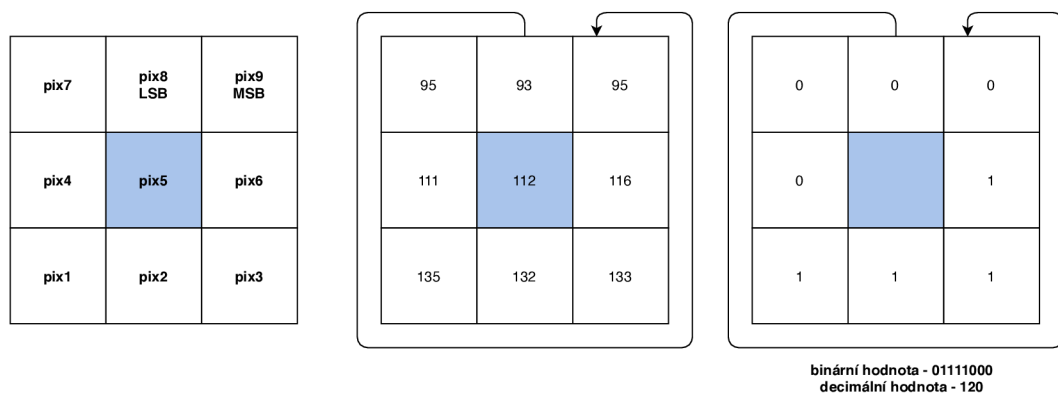
kde I_n jsou hodnoty sousedních pixelů, $I(x_c, y_c)$ je hodnota středového pixelu a n je index souseda [11].

Postup algoritmu je následující:

1. Pro každý pixel $I(x_c, y_c)$ je zjištěno okno o velikosti 3×3 pixelů, kde námi procházený pixel $I(x_c, y_c)$ je středem s danými osmi sousedními pixely I_n .
2. Pro každý pixel I_n v 3×3 okně je spočítána nová hodnota na základě tohoto pravidla – Pokud je hodnota zjišťovaného pixelu I_n větší nebo rovna hodnotě středového pixelu $I(x_c, y_c)$, pak je novou hodnotou zjišťovaného I_n pixelu 1, jinak 0.
3. Po spočítání nových hodnot celého 3×3 okna je spočítána nová binární hodnota středového pixelu $I(x_c, y_c)$.



Obrázek 5.12: Okno o velikosti 9 pixelů se středovým pixelem a jeho sousedy pro analýzu lokálního binárního vzoru [12]



Obrázek 5.13: Princip výpočtu lokálního binárního vzoru

Pro výpočet lokálního binárního vzoru byl použitý algoritmus, který lze zapsat následujícím pseudokódem:

Data: Předpracovaný obraz otisku prstu

Result: Obraz otisku prstu zpracován LBP a výsledný histogram

begin;

for *každý obrázek* **do**

for *každý pixel v obrazu* **do**

 zjistí pro každý pixel jeho sousední pixely v okně 3×3 vyjma krajních pixelů;

for *každý pixel v okně 3×3* **do**

 porovnej každý sousední pixel se středovým pixelem;

if *daný sousední pixel* >= *středový pixel* **then**

 | nastav daný sousední pixel na novou hodnotu 1;

else

 | nastav daný sousední pixel na novou hodnotu 0;

end

end

 získaný 8-bitový binární kód převed na decimální;

 daný středový pixel nastav na získanou decimální hodnotu;

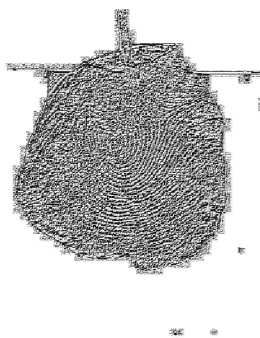
end

 vykresli obraz po zpracování algoritmem lokálního binárního vzoru;

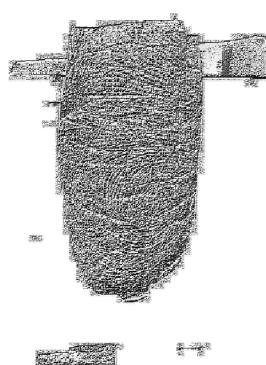
 vygeneruj výsledný histogram obrazu;

end

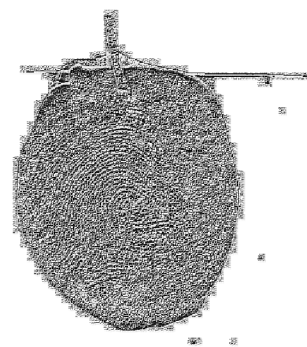
Výsledný předzpracovaný obraz po aplikaci lokálního binárního vzoru může vypadat například následovně:



Obrázek 5.14: Falešný otisk prstu po aplikaci algoritmu LBP



Obrázek 5.15: Živý otisk prstu po aplikaci algoritmu LBP



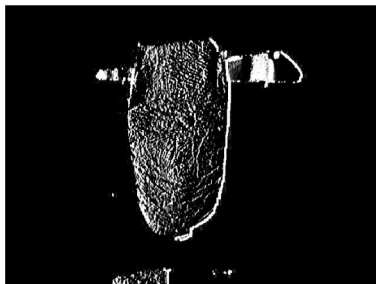
Obrázek 5.16: Falešný otisk prstu po aplikaci algoritmu LBP

5.5 Detektory hran Sobelův a Laplaceův operátor

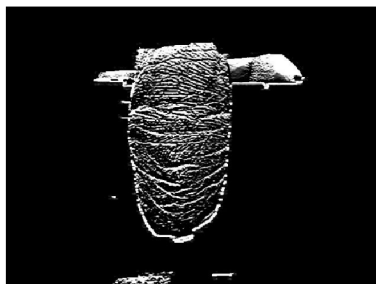
Sobelův a Laplaceův operátor jsou vhodným řešením detekce hran v obrazu a tedy také popisem textury. Umožňují rychlé a snadné řešení, jak snížit počet dat – pixelů pro analýzu a zvýraznit strukturální aspekt vstupního obrazu.

Sobelův operátor je metodou založenou na gradientu. Principem je spočítání derivací prvního řádu pro obraz při ose x a ose y . Získáme tak dva odlišné výsledky, se kterými můžeme dále pracovat. Pro analýzu bylo využito jádro (*kernel*) o velikosti 5×5 , které se využívá pro aproximování daných výsledků – derivací 1. řádu, protože samotný obraz není spojitý.

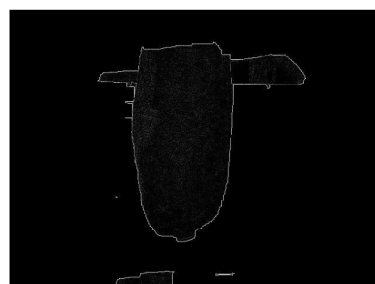
Laplaceův operátor využívá pouze jedno jádro. Získáme tedy jeden finální výsledek, nikoliv dva výsledky pro osu x a y jako u předchozího detektoru hran [38].



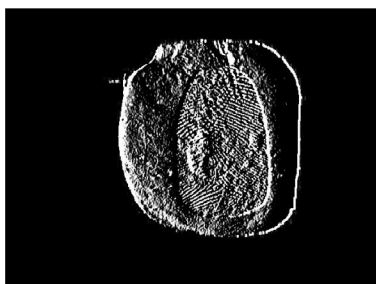
Obrázek 5.17: Výsledek aplikace Sobelova operátoru pro osu x živého otisku prstu



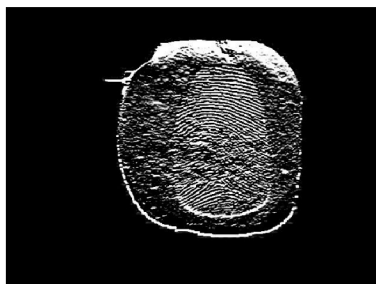
Obrázek 5.18: Výsledek aplikace Sobelova operátoru pro osu y živého otisku prstu



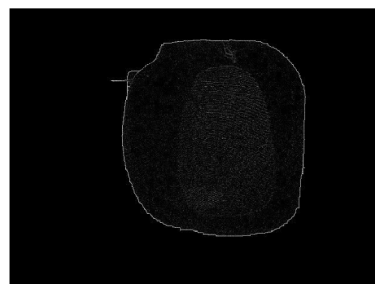
Obrázek 5.19: Výsledek aplikace Laplaceova operátoru pro živý otisk prstu



Obrázek 5.20: Výsledek aplikace Sobelova operátoru pro osu x falešného otisku prstu



Obrázek 5.21: Výsledek aplikace Sobelova operátoru pro osu y falešného otisku prstu



Obrázek 5.22: Výsledek aplikace Laplaceova operátoru pro falešný otisk prstu

5.6 Vlnková transformace

Vlnková transformace (*Wavelet transform*) je další z metod, které mohou být použity při klasifikaci živosti nebo při rozpoznávání. Jedná se o skupinu transformací, které se liší podle dané báze funkce – vlnky. Zadané vlnky jsou odvozené od základní funkce – mateřské vlnky (*mother wavelet*). Vlnková transformace je zástupcem lineární transformace. Signál (obraz) se analyzuje pomocí násobení funkcí okna a ortogonálním rozkladem, podobně jako u dalších integrálních lineárních transformací.

Vlnková transformace může být využita také pro kompresi obrazu nebo pro filtraci šumu [14]. Obecně představuje alternativu ke krátkodobé Fourierově transformaci. Fourierova transformace byla velmi využívána v minulosti, ale nyní se analýza a výzkum více

zaměřují na víceměřítkové transformace (*Multiresolution Transform* – MT), ke kterým patří i vlnková transformace. Fourierova transformace využívá pro rozklad signálu do spektrální roviny periodické sinusové a kosinusové funkce, báze vlnkové transformace je tvořena časově omezenými funkcemi – vlnkami. Vlnky mohou být nadefinovány podle charakteru analyzovaných dat a můžeme dosáhnout i velmi přesné lokalizace prudkých změn v signálu a lepšího vyhodnocení v případě nestacionárních a neperiodických signálů [1].

5.6.1 Rodičovské a dceřiné vlnky

Mateřská vlnka je základní vlnkovou funkcí. Určuje tvar vlnky a pokrývá celý definiční obor, který nás zajímá. Mateřskou vlnku značíme symbolem Ψ . Má nulovou střední hodnotu.

Otcovská vlnka pak poskytuje funkci určující měřítko, tvoří ze základní vlnky další s různým měřítkem a dovoluje vyjádřit detaily aproximované funkce, které zkoumáme. Otcovská vlnka se značí symbolem Φ .

Odvozené vlnky nazýváme dceřinými vlnkami. Tyto vlnky se odvozují od rodičovských vlnek za pomoci generující báze funkce $\Psi_{s,\tau}$, kde parametr s určuje měřítko vlnkové funkce a τ pak její posun [14].

5.6.2 Spojitá vlnková transformace

Spojitou vlnkovou transformaci (*Continuous Wavelet Transform* – CWT) lze zapsat následující rovnicí: [22]

$$y(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \Psi^*\left(\frac{t-b}{a}\right) dt \quad (5.2)$$

Signál $x(t)$ je korelován s vlnkami, které jsou odvozené od mateřské vlnky $\Psi(t)$. Symbol $*$ je pak značením komplexně sdružené funkce. Výsledkem je funkce $y(a, b)$, kde parametr a určuje umístění vlnky na časové ose a parametr b pak měřítko. Konstanta $\frac{1}{\sqrt{a}}$ pak slouží pro normalizaci energie vlnky při změnách měřítka.

Pro případ symetrické vlnkové funkce se postupně hledá míra korelace (podobnosti) mezi naším vstupním signálem a vlnkou s měnícím se měřítkem. Ve výsledku se tedy můžou extrahovat různě velké detaily z dat, která jsou zpracována [1].

5.6.3 Diskrétní vlnková transformace

Diskrétní vlnková transformace (*Discrete Wavelet Transform* – DWT) pro svoje fungování nepotřebuje nadbytečné množství koeficientů, které produkuje právě spojitá vlnková transformace, ale koeficienty redukuje. Používají se koeficienty, které odpovídají měřítkům $s = 2^j$, kde parametr j může nabývat hodnot z množiny $\{0, 1, 2, \dots, n\}$. Díky vhodné dvojkové závislosti parametrů s a p můžeme vytvořit z vlnky Ψ ortonormální bázi. Necht parametry $j, k \in \mathbb{Z}$, pak pro s a p platí:

$$s = 2^j \quad (5.3)$$

$$p = k2^j \quad (5.4)$$

Pak můžeme vzorec diskrétní vlnkové transformace napsat následujícím způsobem:

$$\Psi_{j,k} = \frac{1}{\sqrt{2^j}} \Psi \frac{n - k2^j}{2^j} \quad (5.5)$$

Diskrétní vlnková transformace při zpracování obrazu se využívá hlavně pro detekci hran a textur, kompresi, filtraci šumu a získání důležitých vlastností pro další klasifikaci [1].

5.6.4 Wavelet rodiny vhodné pro analýzu obrazu

Existuje mnoho druhů mateřských vlnek. Na základě mateřské vlnky můžeme odvodit celou danou rodinu. Vlnky mají různé vlastnosti:

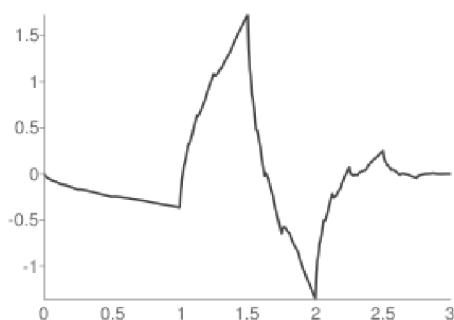
- symetrie,
- asymetrie,
- ortogonálnost – jedná se o vlnku, jejíž vlnková transformace je ortogonální, to znamená, že inverzní vlnková transformace je sdružená vlnkové transformaci [24],
- biortogonálnost – vlastnost vlnky, kdy je daná vlnková transformace invertovaná, ale ne nutně ortogonální [33].

Pro diskrétní vlnkovou transformaci se používají například následující Wavelet rodiny.

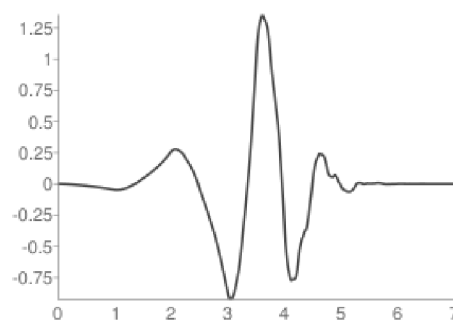
Vlnky Daubechies

Jedná se o asymetrické, ortogonální a biortogonální vlnky. Značí se jako dbA , kde A vyjadřuje počet nulových momentů. Pro práci můžeme využít vlnky Daubechies $db1$ až $db20$, běžně užívány jsou však vlnky typu $db1$ až $db10$ [33]. Dosahují vysoké míry komprese. Poskytují stejné parametry filtrů pro přímou i zpětnou transformaci, zajišťují rychlou a přesnou rekonstrukci [14].

Každá Daubechies vlnka má počet nulových momentů rovných polovině svých koeficientů. Například vlnka $db1$ (také jinak zvaná jako Haar vlnka) má jeden nulový moment, $db2$ pak dva nulové momenty atd. Nulový moment určuje schopnost vlnky reprezentovat chování polynomu nebo informace v signálu. Například vlnka $db1$ s jedním nulovým momentem snadno kóduje konstantní složky signálu. Daubechies vlnka $db2$ snadno pracuje s konstantními a lineárními signálovými složkami, $db3$ pak pracuje s konstantními, lineárními a kvadratickými složkami signálu [5].



Obrázek 5.23: Vlnková funkce pro vlnku Daubechies $db2$ [33]

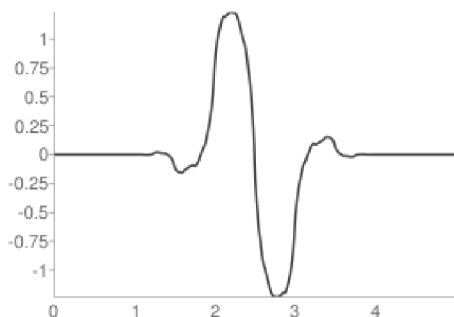


Obrázek 5.24: Vlnková funkce pro vlnku Daubechies $db4$ [33]

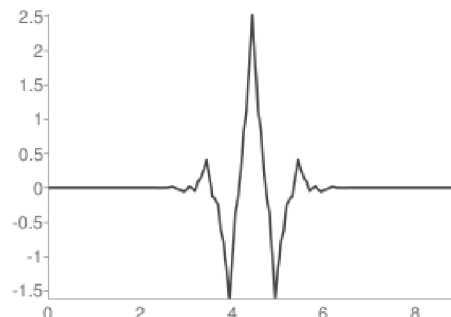
Biortogonální spline vlnky

Biortogonální spline vlnky jsou symetrické, biortogonální, ale nejsou ortogonální. Pro analýzu můžeme využít vlnky $bior1.1$, $bior1.3$, $bior1.5$, $bior2.2$, $bior2.4$, $bior2.6$, $bior2.8$, $bior3.1$, $bior3.3$, $bior3.5$, $bior3.7$, $bior3.9$, $bior4.4$, $bior5.5$ a $bior6.8$ [33]. Jinak se také nazývají jako

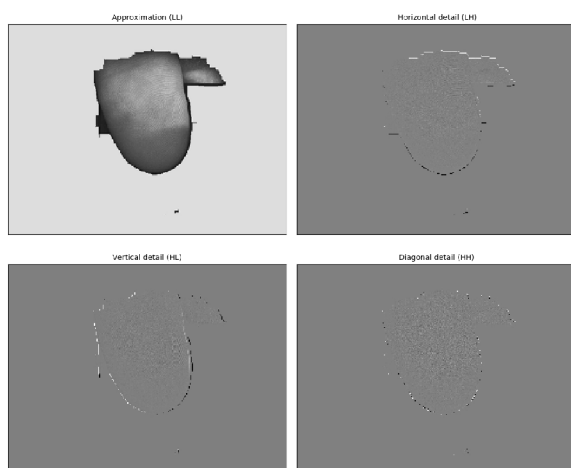
Cohen–Daubechies–Feauveau vlnky. Liší se vlastnostmi a tvarem od vlnek Daubechies, ale princip konstrukce vlnek je podobný. Biortogonální spline vlnky jsou generovány pomocí B-splinu [41]. Jsou využity ve standardu komprese obrazu JPEG 2000, dále se používají například při kompresi otisků prstů pro FBI [42].



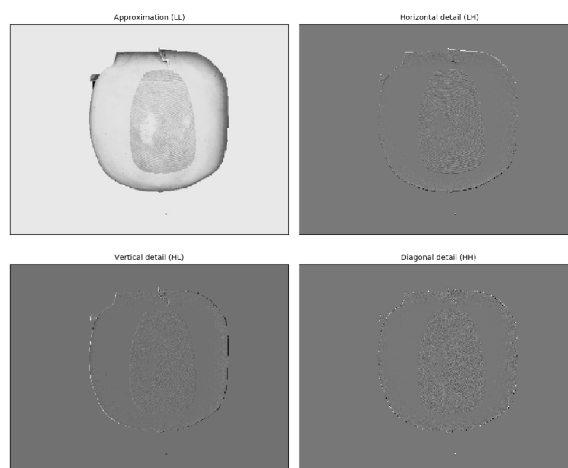
Obrázek 5.25: Dekompoziční vlnková funkce pro bior1.3 [33]



Obrázek 5.26: Dekompoziční vlnková funkce pro bior2.4 [33]



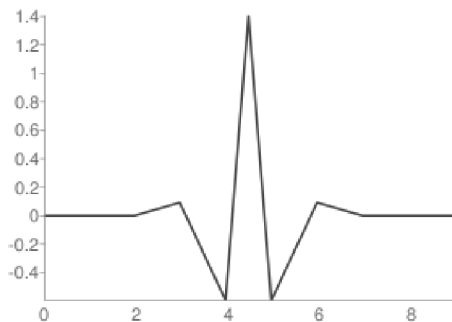
Obrázek 5.27: Vlnková transformace bior1.5 pro živý otisk prstu



Obrázek 5.28: Vlnková transformace bior1.5 pro falešný otisk prstu

Reverzní biortogonální spline vlnky

Reverzní biortogonální spline vlnky jsou symetrické, biortogonální a nejsou ortogonální. Pro práci můžeme využít tyto vlnky z dané rodiny – *rbio1.1*, *rbio1.3*, *rbio1.5*, *rbio2.2*, *rbio2.4*, *rbio2.6*, *rbio2.8*, *rbio3.1*, *rbio3.3*, *rbio3.5*, *rbio3.7*, *rbio3.9*, *rbio4.4*, *rbio5.5* a *rbio6.8* [33]. Tato rodina vlnek byla v minulém výzkumu využita například pro rozpoznávání oční duhovky [40].



Obrázek 5.29: Dekompoziční vlnková funkce pro rbio2.4 [33]

5.6.5 Použité vlnky pro testování algoritmu

Pro testování byly využity tři metody Wavelet rodiny biortogonálních spline vlnek (*bior1.3*, *bior1.5*, *bior2.4*), dvě vlnky rodiny Daubechies (*db2*, *db4*) a jedna vlnka z rodiny reverzních biortogonálních spline vlnek (*rbio3.1*). Cílem práce je poté mimo jiné tyto vlnky analyzovat a porovnat na základě jejich přesnosti při detekci živosti otisků prstů.

5.7 Matice souvislostí stupňů šedi v obraze (GLCM)

Matice GLCM (*Grey Level Cooccurrence Matrix*) slouží jako klasifikátor textury v obraze. Jedná se o čtvercovou matici, která vyjadřuje, jak často se určité kombinace hodnot pixelů v obraze vyskytují. Jedná se tedy o statistickou analýzu dané textury. Existuje až čtrnáct charakteristik, které lze z matice pro daný obraz zjistit. Pro svoji práci jsem využila čtyři z nich – kontrast, korelaci, energii a homogenitu.

Nechť $L \times L$ reprezentuje matici GLCM, kde prvky na souřadnicích (i, j) reprezentují počet výskytů dvou sousedních pixelů, které mají hodnoty i a j v původním obraze. $P_{i,j}$ představuje normalizovanou matici GLCM, proměnná *levels* dále počet stupňů šedi v obraze. Proměnné μ_i , μ_j , σ_i a σ_j charakterizují střední hodnotu a směrodatnou odchylku [43].

Kontrast slouží pro měření intenzity mezi pixely a jejich sousedy v celém analyzovaném obraze. Jeho rozsah je $\langle 0, (size(GLCM, 1) - 1)^2 \rangle$. Kontrast o hodnotě nula představuje konstantní obraz [23] [39].

$$contrast = \sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2 \quad (5.6)$$

Korelace měří míru korelace pixelu ke svým sousedům. Nabývá hodnot v rozsahu daném intervalem $\langle -1, 1 \rangle$. Tyto koncové míry charakterizují plně negativně nebo pozitivně korelovaný obraz [23]. Tuto charakteristiku můžeme spočítat podle následujícího vzorce: [39]

$$correlation = \sum_{i,j=0}^{levels-1} \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \quad (5.7)$$

Energie počítá se sumou čtverců jednotlivých hodnot obsažených v GLCM matici. Výsledek energie leží v intervalu $\langle 0, 1 \rangle$. Pro konstantní vstupní obraz má energie hodnotu 1. Jinak je tato vlastnost také nazývána jako uniformita. Energie je založená na pravidelnosti, vysoké hodnoty představují obraz, který je pravidelně uspořádán [23].

S pojmem energie úzce souvisí vlastnost ASM (*Angular Second Moment*), který se spočítá podle následující rovnice: [39]

$$ASM = \sum_{i,j=0}^{levels-1} P_{i,j}^2 \quad (5.8)$$

Výsledná energie se pak spočítá jako: [39]

$$energy = \sqrt{ASM} \quad (5.9)$$

Poslední extrahovanou charakteristikou byla homogenita. Tato vlastnost představuje blízkost distribuce prvků v GLCM matici ve srovnání s danou GLCM diagonálou. Homogenita může nabývat hodnot z rozsahu $(0, 1)$, kde homogenita s hodnotou jedna je typická pro diagonální GLCM matici [23] [39].

$$homogeneity = \sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1 + (i - j)^2} \quad (5.10)$$

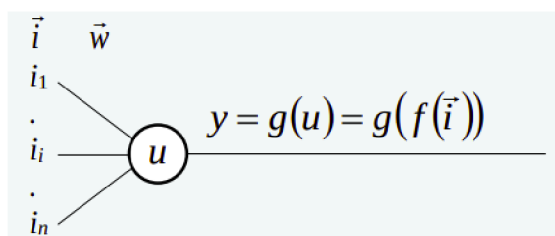
Kapitola 6

Rozhodovací algoritmy živosti

Pro samotné rozhodnutí o živosti či neživosti otisku prstu je potřeba navrhnout a použít některý z klasifikátorů, které jsou založeny na bázi umělé inteligence. Pro vlastní výzkum jsem použila neuronovou síť, klasifikátor SVM (*Support Vector Machines*) a rozhodovací stromy. Tato kapitola popisuje stručný úvod do umělé inteligence a popisy jednotlivých užitých přístupů.

6.1 Umělá neuronová síť

Umělá neuronová síť byla sestavena jako abstrakce podobná lidskému myšlení. Umělý neuron se nazývá perceptron a byl inspirován právě nervovou buňkou – neuronem. Výstupní signály umělých neuronů jsou statické na rozdíl od těch skutečných. Obecné schéma umělého neuronu je znázorněno na následujícím obrázku, proměnná u představuje vnitřní potenciál, f pak bázovou funkci, g aktivační funkci, i hodnotu daného vstupu a w váhu:



Obrázek 6.1: Obecný model umělého neuronu [44]

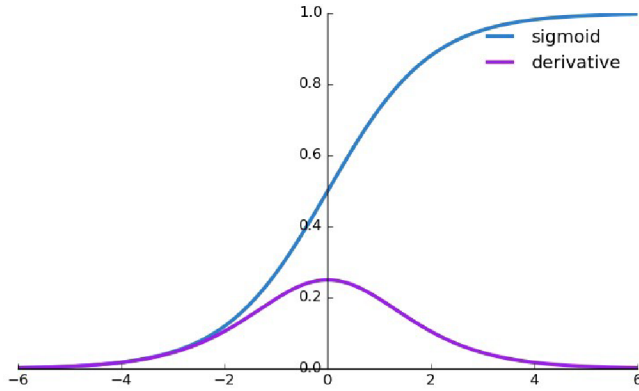
Každá neuronová síť funguje na následujícím principu: Váhy se vynásobí s daným vstupem, dále probíhá přičtení speciální hodnoty zvané *bias*. Poté se aplikuje daná aktivační funkce a vstup se předá další vrstvě. Provádí se zpětná propagace, při které se aktualizují váhy. Vnitřní stav sítě můžeme tedy charakterizovat následujícím předpisem: $w_0x_0 + \sum_{i=1}^n P_{i,j}w_ix_i$, kde *bias* představuje prvek s hodnotou x_0 a s vahou w_0 a proměnné x_1, x_2, \dots, x_i představují vstupní hodnoty s danými vahami w_1, w_2, \dots, w_i . V některých případech může být výstup daných hodnot velmi velký, pokud pracujeme s více vrstvami neuronové sítě, hodnoty se mohou stále stávat vyššími a samotný výpočet pak nekontrolovatelným. Proto hraje důležitou roli aktivační funkce, která na výstupu pracuje s hodnotami v určitém intervalu, např. $(-1, 1)$ nebo $(0, 1)$ [31].

V rámci bakalářské práce jsem jako aktivační funkci využila sigmoidu. Jedná se o nelineární funkci nabývající hodnot v intervalu $(0, 1)$. Proto je vhodná pro modely, u kterých

máme předpovědět pravděpodobnost vstupu. Rovnice funkce je následující:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (6.1)$$

Sigmoida je snadno diferencovatelná, můžeme ji lehce použít pro aktualizaci daných vah.



Obrázek 6.2: Graf sigmoidy a její derivace [31]

Zderivovaná sigmoida má pak následující předpis:

$$g(x) = e^{-x} * (1 + e^{-x})^{-2} \quad (6.2)$$

Dále pro sigmoidu a její derivaci platí následující pravidlo:

$$\frac{df(x)}{f(x)} = f(x) * (1 - f(x)) \quad (6.3)$$

Při zpětné propagaci bylo využito delta pravidlo (*delta rule*). Jedná se o pravidlo, které slouží pro aktualizaci vah vstupů pro umělou neuronovou síť s jednou vrstvou. Pokud máme neuron j a aktivační funkci $f(x)$, delta pravidlo pro i -tou váhu neuronu j – váhu w_{ji} má následující vzorec:

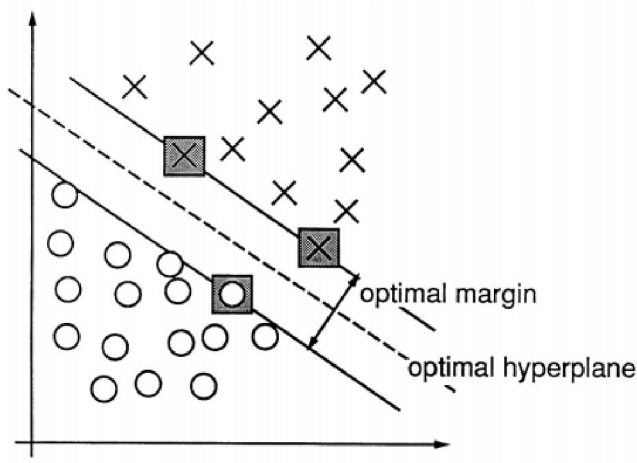
$$\Delta_{ji} = \alpha(t_j - y_j)f'(h_j)x_i \quad (6.4)$$

Parametr α je malá konstanta zvaná *learning rate*, která rozhoduje o velikosti kroku při každé iteraci, většinou se jedná o malou kladnou hodnotu ležící v intervalu $\langle 0, 1 \rangle$ [4]. Funkce $f(x)$ je aktivační funkcí, v našem případě se jedná o sigmoidu, $f'(x)$ je derivace aktivační funkce $f(x)$, t_j jsou výstupy daných vstupů, v našem případě se jedná o výsledky trénovaných vektorů, u kterých víme, zda jsou dané otisky prstů živé (konstanta 1), nebo falešné (konstanta 0). Parametr h_j je váženou sumou vstupů neuronů. Parametr y_j je aktuálním výstupem vstupních vektorů, x_i je i -tým vstupem [35].

Důležitým parametrem pro práci umělé neuronové sítě je počet epoch. Každou epochu se provádí zpracování dat a zpětná propagace, kdy se aktualizují hodnoty vah vstupních vektorů. Po procesu trénování je schopna umělá neuronová síť předpovědět výsledky neznámých vzorků. Čím větší počet epoch probíhá trénování umělé neuronové sítě, tím dokáže algoritmus s větší jistotou rozlišit, zda je otisk prstu živý, nebo falešný. Menší nevýhodou je, že trénování sítě při vyšším počtu epoch trvá déle, řádově několik sekund, výhodou však je stále nižší chyba a také větší jistota ve finální klasifikaci testovaných otisků prstů.

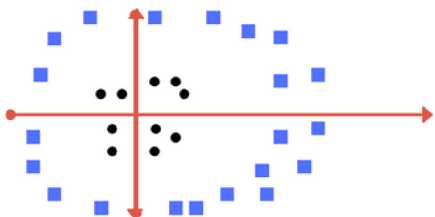
6.2 Algoritmy podpůrných vektorů

Algoritmy podpůrných vektorů (*Support Vector Machines* – SVM) jsou další metodou strojového učení sloužící pro klasifikaci. Principem algoritmu je nalézt nadrovinu pro členění daných vektorů. Data určená pro trénování, u kterých víme, jaké mají výsledky, slouží právě pro nalezení optimální nadroviny, která potom rozdělí do kategorií testované vzorky, u kterých předpovídáme výsledek. V dvourozměrném prostoru je nadrovinou přímka, která dělí rovinu na dvě části (poloroviny), kdy vzorky z jedné třídy leží v jedné polorovině, vzorky druhé třídy pak v druhé polorovině.

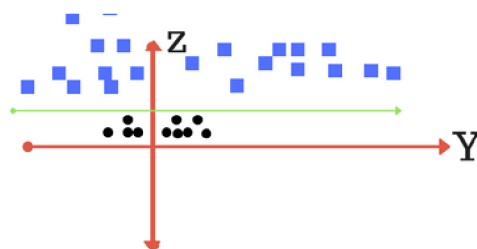


Obrázek 6.3: Příklad problému v dvourozměrném prostoru a nalezení optimální nadroviny a rozpětí (převzato ze studie, na které se podílel jeden z původních výzkumníků algoritmu) [6]

Pokud není možné vzorky rozumně rozdělit polorovinou, aplikujeme transformaci a přidáme jeden další rozměr, osu z . Nechtě hodnoty bodů na ose z mají hodnotu $w = x^2 + y^2$. V tomto případě můžeme měnit vzdálenosti bodu od osy z . Pokud nyní načtneme graf o souřadnicích yz , vidíme zde rozdělení, které můžeme zase oddělit přímkou.

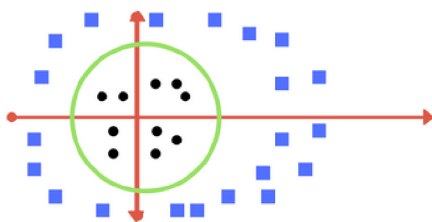


Obrázek 6.4: Vstupní vzorky, které není možné rozdělit polorovinou [28]



Obrázek 6.5: Graf o souřadnicích yz s rozdělením na dané poloroviny [28]

Pokud zpět transformujeme tuto přímku do grafu o závislosti osy y na ose x , vzorky jsou nyní rozděleny hranicí ve tvaru kružnice. Tato transformace se nazývá jádrová transformace. Slouží tedy pro převedení lineárně neseparovatelné úlohy na úlohu, která je již lineárně separovatelná a můžeme tedy nalézt finální rozdělující nadrovinu [28].



Obrázek 6.6: Nalezení finální rozdělující nadroviny [28]

Výhodami SVM jsou:

- jedná se o efektivní metodu v prostorech o více dimenzích,
- SVM je efektivní i v případech, kdy počet dimenzí je větší než počet vzorků,
- jedná se o paměťově efektivní algoritmus [36].

Je však na druhou stranu nevýhodou, že pokud je počet vlastností o mnoho vyšší jak počet vzorků, nemusíme dostávat tolik přesné výsledky.

6.3 Rozhodovací stromy

Cílem dalšího z klasifikačních algoritmů rozhodovacích stromů (*Decision Trees* – DTs) je předpovědět výsledek na základě jednoduchých rozhodovacích pravidel, která jsou odvozená od vlastností dat. Tato klasifikační metoda tedy pro svoji práci využívá graf, který můžeme označit jako strom, protože je souvislý a neobsahuje žádnou kružnici. Každý list stromu reprezentuje test pro atribut, každá větev pak značí výsledek daného testu. Cesty od kořene k listu reprezentují klasifikační pravidla [2].

Výhody rozhodovacích stromů jsou:

- jsou jednoduché na pochopení a vizualizaci, stromy lze také vizualizovat,
- vstupní data nemusí být různě upravována nebo normalizována,
- rozhodovací stromy mohou být využity při problematice, kdy je současně modelem řešeno více klasifikačních problémů,
- používají model white box – bílou skříňku, pokud je daná situace v modelu pozorovatelná, vysvětlení pro podmínky může být snadno objasněno pomocí boolovské logiky, např. umělá neuronová síť je reprezentantem modelu black box – černé skříňky, kdy mohou být výsledky obtížnější interpretovat [37].

Algoritmus však může v některých případech vytvořit až příliš komplexní strom, který data špatně zpracovává a generalizuje. Tato situace se nazývá *overfitting*. Jedná se tedy o chybu v modelování, kdy je model příliš složitý a funkce je příliš kompatibilní s omezenou sadou datových bodů. Při nepatrně nepřesných údajích může finální model získat podstatné chyby a predikce může být pak nepřesná. Dále nedokáže rozhodovací strom snadno zpracovat nenumernická data.

Rozhodovací stromy jsou považovány za jedny z nejlepších a nejpoužívanějších metod pro klasifikaci a regresi. Metody založené na rozhodovacích stromech poskytují vysokou přesnost, stabilitu a snadnost interpretace. Na rozdíl od lineárních modelů jsou schopny snadno řešit i nelineární vztahy [2].

Kapitola 7

Implementace

Cílem samotné implementace bylo aplikovat algoritmy pro předzpracování obrazu a následně využít postupy, které mohou zajistit na základě analýzy textury výsledné rozhodnutí o živosti či neživosti daného otisku prstu. Tato kapitola popisuje nejen samotnou implementaci programu, jeho základní funkcionalitu a logiku, ale také nedílnou součást – postup sběru datasetu na využitém bezdotykovém zařízení. Tento dataset byl využitý jak pro trénování, tak pro testování vzorků s neznámým výsledkem a předpovězení jejich živosti či neživosti. Dále tato kapitola popisuje i samotný bezdotykový senzor, jeho výhody a nevýhody oproti klasickým dotykovým senzorům pro snímání otisku prstu a také jednu z dalších nejdůležitějších částí implementace – extrahované vektory z daných otisků prstů založených na několika principech – algoritmu lokálního binárního vzoru, Sobelova a Laplaceova operátoru a vlnkové transformace.

Pro implementaci byl využit jazyk Python 3.6 s dalšími podporovanými knihovnamí. Pro vývoj bylo použito prostředí operačního systému Ubuntu 18.04.4 LTS.

7.1 Popis bezdotykového senzoru

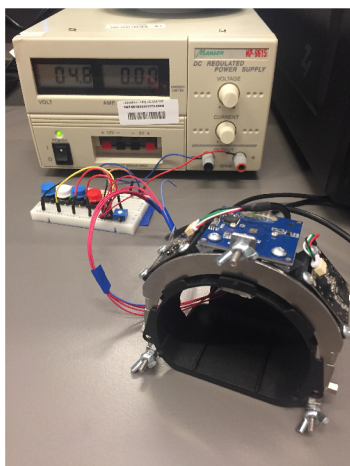
Bezdotykový senzor je představitelem poslední generace senzorů pro snímání otisků prstů. Je založen na typu snímání, kdy daný prst nepřichází do kontaktu s žádným povrchem, jako je to u dotykových senzorů.

Díky kontaktu s daným povrchem se mohou obrazy otisků prstů od stejného objektu lišit. Dále je i použití dotykových senzorů otisků prstů nehygienické v místech, kde ho denně použijí stovky nebo tisíce lidí. Dané povrchy pro analýzu jsou u dotykového senzoru ploché, ale sám povrch prstu plochý není. Objekt musí tedy vyvinout jistý tlak na senzor, který dokáže výsledný obraz zkusit.

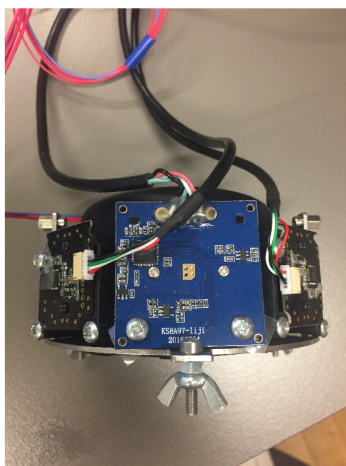
Bezdotykový senzor pro snímání otisků prstů tyto problémy dokáže vyřešit. Snímek otisku prstu je získán bez nutnosti zkusení při dotyku povrchu a problémy týkající se hygieny jsou také vyřešeny, protože se povrchu senzoru nemusíme dotýkat. Dokážeme rychle zachytit celý daný snímání prostor. Bezdotykový senzor může sloužit k 2D i 3D snímání, při analýze je možné použít různé algoritmy pro analýzu textury nebo nasvícení světlem o různých vlnových délkách. Bohužel díky absenci povrchu pro snímání otisků prstů nemusí být snímky dostatečně ostré, protože daný subjekt se musí naučit držet svůj prst v co nejvíce statické poloze s co největším potlačením jakéhokoliv třesu [19].

Bezdotykový senzor použitý v rámci bakalářské práce je složen ze tří kamer a světél o různých vlnových délkách, které dokáží prst nasvítit. Pro výzkum byla použita prostřední

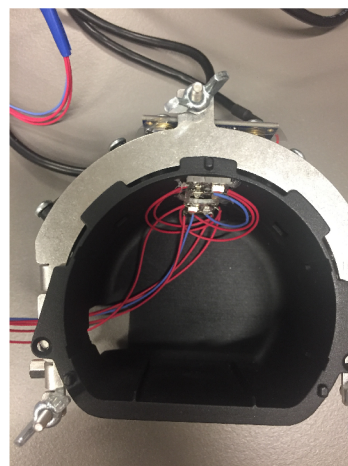
kamera díky její nejlepší kvalitě. Prsty byly nasvicovány modrým, zeleným a červeným světlem. Byly provedeny experimenty i s ultrafialovým zářením, ale toto světlo bylo velmi slabé, výsledky nebyly dobré, a proto nebylo při výsledné analýze nakonec použito.



Obrázek 7.1: Zapojený bezdotykový senzor



Obrázek 7.2: Pohled shora na bezdotykový senzor



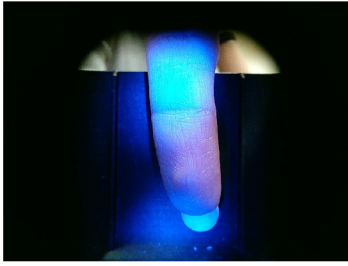
Obrázek 7.3: Detail bezdotykového senzoru

7.2 Sběr datasetu

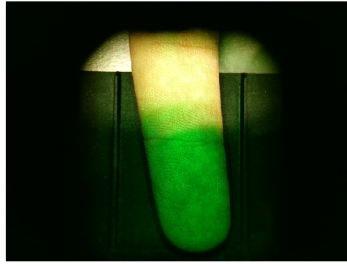
Snímky živých otisků prstů byly získány od dobrovolníků. Prsty byly nasvicovány jednotlivými světly. Bylo potřeba dbát na skutečnost, aby nebyly obrazy přesvětlené a vynikly tak co nejlépe papilární linie a textura povrchu prstu. I přes vynaloženou snahu se mohou na snímcích objevit stíny, avšak jsem se snažila co nejvíce simulovat skutečné používání senzoru, aby dokázal analyzovat živost s co největším rozpětím, v různých podmínkách a s robustností.

Zjistila jsem, že pro některé osoby bylo zpočátku trochu nezvyklé držet prst opravdu ve statické poloze při snímání, což beru jako mírnou nevýhodu bezdotykového senzoru. Dalším mírným problémem při sbírání otisků prstů od subjektů bylo dbání na správné natočení prstu, aby vynikl celý prst a ne například pouze jeho část, pokud účastník prst příliš nesprávně natočil a kamera nemohla celý otisk správně snímat. Sběr datasetu probíhal při více setkáních, kdy bylo možné co nejvíce eliminovat případné chyby a všichni účastníci byly již více obeznámeni s používaným typem bezdotykového senzoru. Pro svoji analýzu jsem dále použila 36 falešných otisků prstů. I zde bylo potřeba dbát na co největší vyniknutí papilárních linií. Bylo nezbytné experimentovat s různými hodnotami napětí pro dané barvy světla právě kvůli zajištění co nejmenšího možného přesvětlení otisků prstů a nejlepšímu vyniknutí jejich textury.

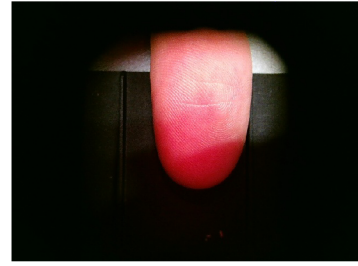
Finální dataset obsahuje 216 snímků otisků prstů nasvícenými jedním z daných světél odpovídající vlnové délky (modrá, zelená nebo červená barva světla) – 108 snímků zobrazuje živý otisk prstu, 108 snímků pak znázorňuje falešný otisk prstu.



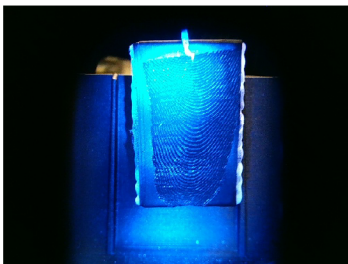
Obrázek 7.4: Obrázek žilvého otisku prstu s nasvícením modrým světlem



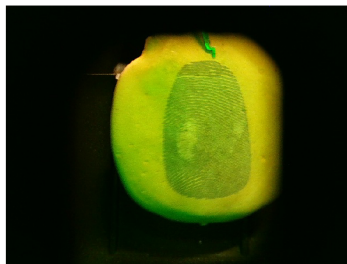
Obrázek 7.5: Obrázek žilvého otisku prstu s nasvícením zeleným světlem



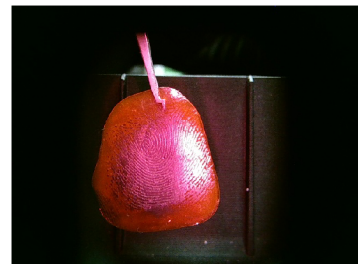
Obrázek 7.6: Obrázek žilvého otisku prstu s nasvícením červeným světlem



Obrázek 7.7: Obrázek falešného otisku prstu s nasvícením modrým světlem



Obrázek 7.8: Obrázek falešného otisku prstu s nasvícením zeleným světlem



Obrázek 7.9: Obrázek falešného otisku prstu s nasvícením červeným světlem

7.3 Knihovna OpenCV

OpenCV je open source knihovna, která slouží pro práci s obrazem a získávání důležitých dat. Vedle programovacího jazyka Python může být využita například i s jazykem Java nebo C++. Samotná knihovna byla napsána v jazyce C/C++ [18]. Tato knihovna byla využita při práci s otisky prstů, její funkce byly nápomocny při předzpracování obrazu a dále napomáhala i při implementaci algoritmu pro lokální binární vzor.

7.4 Knihovny scikit-image a scikit-learn

Pokročilejším ekvivalentem pro knihovnu OpenCV je knihovna scikit-image, která však umožňuje pouze podporu pro jazyk Python. Byla nápomocná při extrahování matice GLCM z obrazu, jejíž některé vlastnosti byly klíčové pro extrahovaný vektor při všech uvažovaných a zkoumaných formách analýzy. Knihovna scikit-learn se pak primárně specializuje na podporu strojového učení. Pro klasifikaci byla využita funkce pro metodu SVM a taky pro rozhodovací strom. Knihovna scikit-learn představuje snadnou práci s algoritmy umělé inteligence, algoritmy jsou dobře optimalizované a rychlé. Umožňuje experimenty s různými metodami a přístupy, na základě kterých můžeme uvážit co nejvíce metod klasifikace, které mezi sebou můžeme porovnat a zkoumat chování v různých podmínkách.

7.5 Knihovna PyWavelets

Knihovna PyWavelets představuje open source knihovnu umožňující aplikování vlnkové transformace při práci v jazyce Python. Knihovna je částečně implementována v jazyce C. PyWavelets poskytuje 1D či 2D diskrétní vlnkovou transformaci a také 1D spojitou vlnkovou transformaci. Obsahuje více než 100 vestavěných vlnkových filtrů a podporuje i tvorbu vlastních vlnek uživatelem. Poskytuje co nejpřesnější výsledky, které jsou kompatibilní například s MATLAB Wavelet Toolbox (TM) v programovacím jazyce MATLAB.

Knihovna byla využita pro diskrétní vlnkovou transformaci analyzovaných otisků prstů. Vedle využitých vlnkových rodin (biortogonální spline vlnky, reverzní biortogonální spline vlnky, vlnky Daubechies) poskytuje pro možnou analýzu i další vlnkové rodiny, jako např. Symlety nebo Coiflety [32].

7.6 Extrahované vektory z obrazu

Tři různé vektory byly extrahovány pro detekci živosti. Pro každý vektor byla jednou z nejdůležitějších charakteristik matice GLCM a její vlastnosti – kontrast, korelace, energie a homogenita. Vektory jsou založené na algoritmu lokálního binárního vzoru, Sobelova a Laplaceova operátoru a vlnkové transformaci.

7.6.1 Vektor založený na lokálním binárním vzoru

Tento vektor je založen na analýze histogramu obrazu zpracovaného algoritmem LBP. První část vektoru tvoří částečné sumy daného histogramu. Histogram obrazu po zpracování lokálním binárním vzorem tvoří 256 hodnot stupně šedi a na nich závislém počtu pixelů. Parametr $value_b$ udává hodnotu počtu pixelů pro daný stupeň šedi b , kde b je z množiny $\{0, 1, 2, \dots, 255\}$.

Histogram tedy můžeme zapsat jako uspořádanou množinu, kterou tvoří dané hodnoty:

$$histogram = \{value_b, value_{b+1}, \dots, value_{255}\} \quad (7.1)$$

Pro analýzu byl využit experiment, kdy byly extrahovány čtyři částečné sumy histogramu. Každá z nich udává součet pixelů pro dané rozmezí stupňů šedi.

$$sum_1 = \sum_{b=0}^{b=63} value_b \quad (7.2)$$

$$sum_2 = \sum_{b=64}^{b=127} value_b \quad (7.3)$$

$$sum_3 = \sum_{b=128}^{b=191} value_b \quad (7.4)$$

$$sum_4 = \sum_{b=192}^{b=255} value_b \quad (7.5)$$

Druhou část extrahovaného vektoru v tvoří čtyři zjišťované vlastnosti z matice GLCM – kontrast f_1 , homogenita f_2 , energie f_3 a korelace f_4 . Tyto charakteristiky jsou získány z předzpracovaného šedotónového obrazu.

Výsledný vektor v_1 určený pro analýzu můžeme tedy zapsat následovně:

$$v_1 = [sum_1, sum_2, sum_3, sum_4, f_1, f_2, f_3, f_4] \quad (7.6)$$

7.6.2 Vektor založený na Sobelově a Laplaceově operátoru

Pro extrakci hran a textury pomocí Sobelova operátoru bylo využito jádra o velikosti 5×5 . OpenCV umožňuje snadné extrahování pomocí Sobelova operátoru pouze pro osu x , nebo pro osu y . Získáme tedy odlišné výsledky pro horizontální a vertikální část. Poté byl obraz zpracován Laplaceovým operátorem, který poskytuje pouze jeden výsledek a nikoliv rozložení pro obě osy.

Vektor pak obsahuje dvanáct hodnot. Z GLCM matice jsou opět získány důležité vlastnosti – kontrast f_1 , homogenita f_2 , energie f_3 a korelace f_4 , které jsou spočítány pro oba výsledky zpracované s využitím Sobelova operátoru Sob_x a Sob_y a jeden výsledek získaný Laplaceovým operátorem $Lapl$.

Vektor v_2 , který je extrahován z obrazu, tedy vypadá následovně:

$$v_2 = [f_{1_{Lapl}}, f_{2_{Lapl}}, f_{3_{Lapl}}, f_{4_{Lapl}}, f_{1_{Sob_x}}, f_{2_{Sob_x}}, f_{3_{Sob_x}}, f_{4_{Sob_x}}, f_{1_{Sob_y}}, f_{2_{Sob_y}}, f_{3_{Sob_y}}, f_{4_{Sob_y}}] \quad (7.7)$$

7.6.3 Vektor založený na vlnkové transformaci

Knihovna PyWavelets poskytuje mnoho metod založených na různých vlnkových rodinách. Pro diskrétní vlnkovou transformaci byla využita analýza pomocí rodiny Daubechies vlnek, biortogonálních spline vlnek a reverzních biortogonálních spline vlnek. Po zpracování je výsledek rozložen na tři detaily – horizontální LH , vertikální HL a diagonální HH . Znovu byly z každého detailu extrahovány vlastnosti matice GLCM – kontrast f_1 , homogenita f_2 , energie f_3 a korelace f_4 . Získáme tedy zase dvanáct různých hodnot, které budou využity pro finální vektor v_3 :

$$v_3 = [f_{1_{LH}}, f_{2_{LH}}, f_{3_{LH}}, f_{4_{LH}}, f_{1_{HL}}, f_{2_{HL}}, f_{3_{HL}}, f_{4_{HL}}, f_{1_{HH}}, f_{2_{HH}}, f_{3_{HH}}, f_{4_{HH}}] \quad (7.8)$$

7.7 Práce s implementovaným programem

Cílem implementovaného programu je zejména:

- předzpracování vstupního obrazu otisku prstu získaného ze senzoru,
- extrakce vektorů na základě uživatelských pokynů,
- klasifikace a výpočet její přesnosti, FAR a FRR,
- zobrazení zpracovaných otisků prstů uživateli.

Program využívá open source knihoven OpenCV, scikit-image, scikit-learn, PyWavelets a dalších, které jsou uvedeny v Readme na přiloženém paměťovém médiu a v manuálu. Všechny knihovny lze snadno stáhnout např. pomocí příkazu v terminálu pro instalaci knihoven v jazyce Python 3.6 `pip3`.

Program může být spuštěn ve dvou režimech:

- extrakce vektorů a klasifikace – Umožňuje dle navržené metody získat vektory z obrazu a následně provést trénování klasifikační sítě a testování neznámých vzorků.

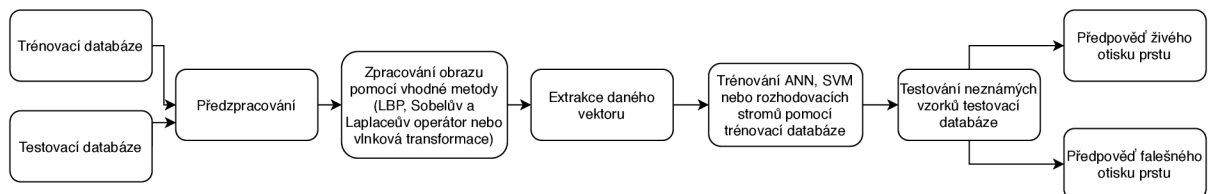
- metoda zobrazení jednotlivých technik – Na vstupním obraze zadaném uživatelem se provedou techniky a algoritmy dílčí pro analýzu – segmentace, lokální binární vzor, Sobelův a Laplaceův operátor a vlnková transformace. Výsledný zpracovaný otisk prstu se poté zobrazí uživateli jako výsledek.

7.7.1 Struktura programu

Celý dataset otisků prstů je uložen ve složce `./dataset`. Pro ukládání snímků v této složce určených k trénování slouží podsložky `/blueTrain`, `/greenTrain` a `/redTrain` pro jednotlivé rozdělené datasety na otisky prstů nasvícených stejnou barvou světla a složka `/allTrain` pro práci s datasetem obsahujícím snímky nasvícenými jednou z těchto tří barev. Testované snímky se pak nacházejí ve složkách `/blueTest`, `/greenTest`, `/redTest` a `/allTest`. Je důležité dodržovat názvy snímků, pro živé otisky prstů se musí v souboru vyskytovat slovo `live`, u falešných pak slovo `fake`.

Pokud program provádí extrakci vektorů a následnou klasifikaci, jsou využité dva cykly `for` – jeden prochází snímky určené k trénování a druhý k testování. Všechny extrahované vektory jsou ukládány do `csv` souborů. U trénovaných snímků se vytvoří dva soubory – jeden pro zápis samotného vektoru na nový řádek `csv` souboru a druhý pro zápis výsledku daného otisku prstu (konstanta 1 – živý otisk, konstanta 0 – falešný otisk). U trénovaných snímků se vytvoří jeden `csv` soubor, který obsahuje na každém řádku jako první položku název souboru a pak extrahovaný vektor. Všechny `csv` jsou ukládány do složky `./csvFiles`. Při klasifikaci pomocí umělé neuronové sítě, metody SVM nebo rozhodovacího stromu je provedeno trénování na základě vektorů získaných ze snímků k tomu určených. Poté se získá daná předpovězená hodnota podle klasifikace a porovná se s názvem obrázku, který je součástí souboru `csv`. Opět, pokud obsahuje název obrázku slovo `live`, jedná se o živý otisk, pokud `fake`, tak falešný. Získaná předpovězená hodnota se porovná s tímto názvem souboru. Uživateli se poté zobrazí název souboru a předpovězená hodnota (konstanta 0 nebo 1), u klasifikace neuronové sítě je získáno přímo procentuální zastoupení pro živost a neživost daného otisku prstu. Počítá se hromadná průměrná přesnost celého testovaného datasetu. Dále se finálně vedle přesnosti uživateli zobrazí i vypočtené FAR a FRR.

Pokud chce uživatel pouze zobrazit obraz otisku prstu s využitím některé z metod (segmentace, lokální binární vzor, Sobelův a Laplaceův operátor, vlnková transformace), program pouze zavolá příslušný modul, předá mu cestu k danému obrázku, případně další parametry (typ prahování pro segmentaci, typ vlnky u vlnkové transformace) a provede následné zpracování a zobrazení finálního výsledku uživateli. Soubory obsahují i ukázky segmentovaného obrázku, detailů vlnkové transformace a Sobelova a Laplaceova operátoru, průběžně se při zpracování ukládají do složky `./processedImg` a případně přepisují při iteraci složek.



Obrázek 7.10: Schéma implementované problematiky detekce živosti

Kapitola 8

Experimenty a vyhodnocení

V následující kapitole se zaměřím na analýzu získaného a použitého datasetu. Tato kapitola shrnuje průběh experimentů, které byly v rámci práce realizovány. V první části jsem rozdělila dané snímky na tři různé menší datasety, každý s obrazy nasvícenými světlem o stejné vlnové délce (zeleným, modrým a červeným). Ve druhé části jsem analyzovala celý získaný dataset obsahující kombinaci snímků otisků prstů o různých nasvícených světlech, tedy soubor všech snímků získaných z použitého bezdotykového senzoru. Mým úkolem bylo analyzovat různé přístupy, metody klasifikace, segmentace či u datasetu se stejným nasvíceným světlem zjistit, jaké světlo poskytuje největší přesnost při analýze. Nedílnou součástí je vyhodnocení všech provedených experimentů a porovnání různých užitých přístupů.

8.1 Analýza datasetů otisků prstů nasvícených stejnou barvou světla

Cílem experimentu je zjistit, jaká segmentace, extrahovaný vektor, algoritmus pro klasifikaci a hlavně barva světla poskytují nejlepší výsledky při detekci živosti otisků prstů. Získaný dataset byl rozdělen na tři menší, kde každý obsahoval snímky nasvícené stejnou barvou světla. Dataset byl dále rozdělen na snímky sloužící pro trénování a snímky sloužící pro testování. Pro trénování byl vždy zachován stejný poměr snímků živých otisků prstů a falešných otisků prstů. U trénovacích snímků jsem programu předložila i jejich výsledky, které byly znázorněny konstantou – číslem 1 pro živý otisk prstu a číslem 0 pro falešný otisk prstu. Klasifikátor měl dále po trénování na daných snímcích předpovědět výsledek pro testované otisky prstů, u kterých algoritmus předem nedokázal rozhodnout, zda jsou živé nebo falešné. Níže jsou uvedeny počty snímků všech tří rozdělených datasetů – datasetů, u kterých jsou otisky získané ze senzoru nasvícené pouze modrou, zelenou nebo červenou barvou světla.

	Počet trénovaných vzorků	Počet testovaných vzorků
Živé otisky prstů	21	18
Falešné otisky prstů	21	9
Suma	42	27

Tabulka 8.1: Dataset snímků nasvícených pouze modrým světlem

	Počet trénovaných vzorků	Počet testovaných vzorků
Živé otisky prstů	23	11
Falešné otisky prstů	23	17
Suma	46	28

Tabulka 8.2: Dataset snímků nasvícených pouze zeleným světlem

	Počet trénovaných vzorků	Počet testovaných vzorků
Živé otisky prstů	23	12
Falešné otisky prstů	23	15
Suma	46	27

Tabulka 8.3: Dataset snímků nasvícených pouze červeným světlem

Klasifikátory SVM, rozhodovací stromy a umělé neuronové sítě byly určeny pro finální klasifikaci testovaných otisků prstů. Můžeme vidět, že umělá neuronová síť dosahuje nejvyšší přesnosti. Přesnost této metody tkví zejména ve skutečnosti, že neuronová síť může být trénována pomocí velkého počtu epoch. Počet epoch je parametr, který definuje, kolikrát algoritmus učení bude pracovat s celým trénovacím datasetem [3]. Můžeme nastavit několik tisíce epoch, ale i více. Algoritmus se pak stává stále více jistějším ve finální analýze pro testované vzorky a výsledná síť by se měla stále více přiklánět k jednomu či druhému výsledku (konstanta 1 – živý otisk prstu, konstanta 0 – falešný otisk prstu). Samozřejmě však je nutno podotknout, že čím více epoch je umělá neuronová síť trénovaná, tím může finální klasifikace trvat delší dobu.

	Průměrná přesnost klasifikace [%]
ANN	90.131
SVM	84.392
DTs	82.983

Tabulka 8.4: Srovnání přesnosti použitých klasifikačních metod

Dále byly testovány průměrné přesnosti všech metod segmentace – s využitím prahování pomocí algoritmu Otsu nebo adaptivního prahování Mean a Gaussian. Bylo zjištěno, že ačkoliv by mohla být segmentace s využitím adaptivního prahování lepší, protože tato metoda je vhodná i pro snímky s různými světelnými podmínkami, s obsahem stínů atd., nakonec nejvyšší přesnost vykazuje segmentace s Otsu prahováním pomocí konstantního optimálního prahu závislého na daném vstupním obrazu.

	Průměrná přesnost klasifikace [%]
Otsu	88.574
Mean	85.882
Gaussian	82.418

Tabulka 8.5: Srovnání přesnosti jednotlivých metod segmentace s využitím typem prahování

Pro všechny využití druhy segmentace byla testována každá ze zvolených metod kromě metody s extrahovaným vektorem na základě Sobelova a Laplaceova operátoru. Pro tuto metodu byla zvolena pouze segmentace s Otsu prahováním, protože při využití adaptivního

prahování Mean a Gaussian nevykazovala křivka míry chyby pro umělou neuronovou síť správné vlastnosti. Níže můžeme vidět průměrnou přesnost pro všechny metody, při kterých se uvažovaly všechny tři typy segmentace. Metoda využívající extrahovaného vektoru na základě lokálního binárního vzoru byla nejpřesnější, dosahuje přesnosti průměrně 90.2%. Nejspolehlivější metodou založenou na vlnkové transformaci byla metoda *rbio3.1* spadající do rodiny reverzních biortogonálních spline vlnek. Nejspolehlivější metodou biortogonálních spline vlnek byla vlnka *bior1.3*. Ačkoliv jsou vlnky Daubechies hojně používanými při zpracování obrazu, jejich přesnost je ve výsledku v provedeném experimentu jednou z nejnižších. Přesto však všechny tyto metody dosahují přesnosti vyšší než 80% a průměrná přesnost vektoru založeném na vlnkové transformaci je 84.9%.

	Průměrná přesnost klasifikace [%]
LBP	90.197
rbio3.1	86.258
bior1.3	85.303
bior1.5	85.146
db4	84.994
bior2.4	84.411
db2	83.064

Tabulka 8.6: Srovnání přesnosti jednotlivých metod využívající všechny typy segmentace

Protože metoda založená na vektoru získaného z obrazu založeného na Sobelově a Laplaceově operátoru využívá jen segmentaci s prahováním Otsu, bylo potřeba spočítat i průměrnou přesnost metod pouze s použitím tohoto prahování. Klasifikace podle těchto dvou algoritmů fungujících jako detektory hran je poměrně přesná a hlavně velmi rychlá. U datasetů nasvícených pouze jednou stejnou barvou světla dosahuje průměrné přesnosti 90.3%. Pro segmentaci s využitím pouze Otsu prahování se překvapivě nejvíce přesnou stává metoda vlnkové transformace *bior1.5*, která v tabulce výše byla při využití všech typů segmentace umístěna až na čtvrtém místě.

	Průměrná přesnost klasifikace [%]
bior1.5	90.638
rbio3.1	90.285
Sobelův a Laplaceův operátor	90.256
LBP	89.830
bior1.3	89.051
db4	88.242
db2	86.993
bior2.4	84.979

Tabulka 8.7: Srovnání přesnosti jednotlivých metod, které využívají pouze segmentaci s prahováním Otsu

Na konec tohoto experimentu můžeme stanovit, které světlo odpovídající vlnové délky se podílelo na největší přesnosti při detekci živosti na použitém bezdotykovém senzoru. Bylo zjištěno, že nejlepší přesnost vykazují snímky nasvícené červeným světlem. Proto byly nejvhodnější použitou barvou světla na užitém bezdotykovém senzoru.

Modré světlo má nejkratší vlnovou délku ve srovnání se zeleným a červeným světlem, je nejvíce absorbováno prstem. Na druhou stranu červené světlo s nejdelší vlnovou délkou dokáže prst co nejvíce prosvítit při hardwarové detekci živosti, která užívá multispektrální analýzu a CMOS nebo CCD detektor, který má měřit odražené světlo od otisku prstu [10].

Červené světlo je tedy nejen světlem, které celý prst dokáže prosvítit, ale snímky daného světla jsou nejpřesnější pro naši představenou metodu. Avšak všechny tři barvy LED světél vykazovaly při analýze přesnost nad 80 %.

	Průměrná přesnost klasifikace [%]
Červená barva světla	90.067
Modrá barva světla	86.756
Zelená barva světla	80.682

Tabulka 8.8: Srovnání přesnosti jednotlivých barev světla

8.2 Výsledky analýzy datasetu obsahující všechny získané snímky otisků prstů

Na tento experiment byl použit celý nerozdělený dataset snímků získaných z bezdotykového zařízení. Dataset tedy obsahuje jak nasvícené otisky prstů modrým světlem, tak i zeleným a červeným. Cílem tohoto pokusu je opět najít nejlepší formu klasifikace, prahování pro použitou segmentaci a nejpřesnější metody. Je nutné však podotknout, že extrakce vektorů pro klasifikaci trvá déle vzhledem k počtu snímků, zvláště u vlnkové transformace. Výsledkem jsou však velmi dobré průměrné přesnosti.

Níže můžeme vidět tabulku počtu trénovaných otisků prstů a testovaných otisků, které měly neznámý výsledek a sloužily pro předpovězení jejich finální živosti či neživosti. Dále vidíme míru zastoupení jednotlivých kategorií na základě nasvíceného světla a počty živých a falešných otisků prstů. I zde byla každá kategorie pro trénovací snímky zastoupena stejným poměrem.

Barva světla	Otisky	Trénované vzorky	Testované vzorky
Modrá	Živé	23	16
	Falešné	23	7
Zelená	Živé	23	11
	Falešné	23	17
Červená	Živé	23	12
	Falešné	23	15
Suma		138	78

Tabulka 8.9: Vlastnosti analyzovaného datasetu pro daný experiment

Pro experiment jsou opět použity všechny tři metody klasifikace – umělé neuronové sítě (ANN), SVM a rozhodovací stromy (DTs). Opět v průměrné přesnosti zaujímá první místo umělá neuronová síť, následována SVM. Jako poslední se pak umístily rozhodovací stromy, které i tak však dosahují přesnosti nad 83 %.

	Průměrná přesnost klasifikace [%]
ANN	88.753
SVM	85.023
DTs	83.625

Tabulka 8.10: Srovnání přesnosti použitých klasifikačních metod

I u tohoto většího získaného datasetu čítající více než 200 snímků je metoda Otsu prahování ta, která dává nejpřesnější výsledky. Metody využívající segmentaci s adaptivním prahováním, metody Mean a Gaussian, se umístily opět na druhém a třetím místě, stejně jako u předchozího pokusu.

	Průměrná přesnost klasifikace [%]
Otsu	87.729
Mean	85.348
Gaussian	83.333

Tabulka 8.11: Srovnání přesnosti jednotlivých metod segmentace s využitým typem prahování

Nejdůležitějším úkolem bylo poté porovnat všechny metody a znovu nalézt tu, která vykazuje největší přesnost. Prvně porovnáme metody, u kterých byly využity všechny tři typy prahování při segmentaci obrazu. Jedná se tedy o všechny přístupy s výjimkou vektoru založeného na Sobelově a Laplaceově operátoru. Vektor založený na lokálním binárním vzoru vykazuje největší přesnost klasifikace 89.3 %. Nejpřesnějšími přístupy založenými na vlnkové transformaci jsou pak *rbio3.1* s 86.9 % a *bior1.3* s 86.3 %. Metody spadající do Daubechies vlnkové rodiny jsou jedny z nejméně přesných při našem experimentu – vlnka *db4* má průměrnou přesnost 84.6 % a vlnka *db2* pak 81.8 %. Při tomto experimentu je celková průměrná přesnost při klasifikaci pomocí vektoru založeném na vlnkové transformaci 84.8 %.

	Průměrná přesnost klasifikace [%]
LBP	89.316
rbio3.1	86.895
bior1.3	86.325
bior2.4	85.185
db4	84.615
bior1.5	84.188
db2	81.766

Tabulka 8.12: Srovnání přesnosti jednotlivých metod využívající všechny typy segmentace

Následující tabulka opět udává průměrnou přesnost metod, avšak pouze při využití Otsu prahování. Tímto způsobem tedy můžeme porovnat všech osm metod i s využitím Sobelova a Laplaceova operátoru. Při využití našeho kompletního získaného datasetu ze senzoru vykazuje největší přesnost právě vektor založený na Sobelově a Laplaceově operátoru. Tento nejrychlejší způsob extrakce vektoru pro rozsáhlejší dataset klasifikuje otisky prstů s neznámým výsledkem s průměrnou přesností 92.7 %. Lokální binární vzor je pak druhým nej-

lepším přístupem pro finální klasifikaci s průměrnou přesností 91.9 %. Biortogonální spline vlnková rodina je pak nejlepší metodou vlnkové transformace pro tuto část experimentu. Vlnky *bior1.3* a *bior2.4* jsou nejvíce přesné se shodným výsledkem 88.5 %. Bohužel vlnka *rbio3.1* z rodiny reverzních biortogonálních spline vlnek je jednou z nejhorších při tomto přístupu, má přesnost 87.2 %. Nutno však podotknout, že segmentace s využitím Otsu prahování je celkově velmi přesnou a všechny metody dosahují nejméně 82 %, ale v mnohých případech mnohem více.

	Průměrná přesnost klasifikace [%]
Sobelův a Laplaceův operátor	92.735
LBP	91.880
bior1.3	88.462
bior2.4	88.462
db2	87.607
bior1.5	87.607
rbio3.1	87.179
db4	82.906

Tabulka 8.13: Srovnání přesnosti jednotlivých metod, které využívají pouze segmentaci s prahováním Otsu

8.3 Vyhodnocení experimentů

Extrahovaný dataset určený pro experimenty obsahuje celkem 216 snímků – 108 snímků živých otisků prstů a 108 snímků falešných otisků prstů nasvícených jedním ze tří barevných LED světél – modrým, zeleným nebo červeným. Pro klasifikaci testovaných snímků a tedy předpovězení, zda se jedná o živý, nebo falešný otisk prstu, byla využita umělá neuronová síť, algoritmy podpurných vektorů a rozhodovací stromy. Pro segmentaci otisku prstů z obrazu byly využity tři typy prahování – prahování pomocí Otsu s využitím konstantního prahu závislém na daném snímku a adaptivní prahování Mean a Gaussian s variabilní hodnotou prahu pro každý pixel obrazu. Tři různé vektory byly extrahované z obrazu pro provedené experimenty – vektor založený na lokálním binárním vzoru, vektor založený na Sobelově a Laplaceově operátoru a vektor založený na vlnkové transformaci.

V první části experimentu byl dataset rozdělen na tři menší, ve kterých byly pouze snímky nasvícené jedním z těchto světél. Cílem bylo nalézt nejspolehlivější klasifikátor, typ prahování při segmentaci, metodu a konečně barvu světla, při které dávají nasvícené otisky prstů nejlepší výsledky přesnosti. Nejspolehlivější metodou klasifikace byla umělá neuronová síť s přesností 90.1 %. I jednoduchá neuronová síť dokáže být velmi přesnou, implementovaná neuronová síť v programu je trénována pro 500 000 epoch. Byly provedeny pokusy i pro trénování s větším počtem epoch, ale neuronová síť pak vykazovala již známky situace zvané *overfitting*, kdy při větším počtu epoch se chyba zvyšovala a nikoliv snižovala. Klasifikace pomocí neuronové sítě trvá pouze několik málo sekund, hodně záleží na počtu epoch, kdy více epoch trvá programu delší dobu pro výpočet.

Přestože jsem se snažila použít i metody adaptivního prahování, u kterých jsem předpokládala, že budou pro snímky ze senzoru vhodnější než segmentace s prahováním Otsu, právě tato metoda s konstantním prahem v závislosti na daném vstupním obrazu byla nejpresnější, dosahovala 88.6 %. Výhodu této metody také vidím v tom, že při ní nemusí člověk

nastavovat různé parametry jako u adaptivních metod prahování (velikost bloku, konstanta C), hodnota prahu je vybrána automaticky, bez jakéhokoliv zkoumání, jaká konstanta C nebo velikost bloku fungují nejlépe pro snímky z našeho použitého senzoru.

Nyní můžeme porovnat jednotlivé použité metody. Pro metodu využívající Sobelova a Laplaceova operátoru byla zvolena pouze Otsu segmentace. Celkově největších přesností dosahovala metoda využívající lokální binární vzor (90.2%), metoda používající Sobelův a Laplaceův operátor (90.3%) a vlnková transformace s použitou vlnkou z rodiny biortogonálních spline vlnek (průměr tří testovaných vlnek činil 85.0%) a reverzních biortogonálních spline vlnek (86.3%). Průměrná přesnost klasifikace s využitím vlnkové transformace byla 84.8%.

Nakonec byla určena barva světla, která dávala při použití nejlepší výsledky. Snímky nasvícené červenou barvou světla prováděly klasifikaci s přesností 90.1%. Otisky prstů nasvícené červeným světlem tedy poskytovaly nejlepší výsledky, na rozdíl od modré a hlavně zelené barvy, kdy otisky prstů nasvícené modrou barvou se klasifikovaly s průměrnou přesností 86.8% a prsty nasvícené zeleným světlem pak 80.7%.

Cílem druhého experimentu bylo pracovat se všemi získanými snímky z bezdotykového senzoru. Těchto 216 vzorků bylo rozděleno znovu na snímky určené pro trénování s předem známým výsledkem a snímky určené pro testování, u kterých měl klasifikátor předpovědět, zda se jedná o živý nebo falešný otisk prstu. Nejlepším klasifikátorem se opět stala umělá neuronová síť jako v předchozím experimentu. Zde však dosahovala mírně nižší přesnosti než v předchozím případě – 88.8%. Otsu metoda je opět nejlepším algoritmem využitým pro segmentaci. Dosahuje opět opravdu nepatrně nižší přesnosti jako u prvního experimentu – 87.7%.

Nejlepší metody jsou shodné s prvním experimentem, akorát celkové přesnosti se mírně změnily. Přesnost lokálního binárního vzoru se snížila na 89.3%. Přesnost nejlepších metod vlnkové transformace se však průměrně zvýšila. Biortogonální spline vlnky dosahují průměrné přesnosti 85.2% a reverzní biortogonální spline vlnky pak 86.9%. Celková průměrná přesnost metod vlnkové transformace je 84.9%, jen nepatrně se zvýšila ve srovnání s prvním experimentem. Sobelův a Laplaceův operátor klasifikuje otisky prstů s přesností 92.7%. Metoda založená právě na Sobelově a Laplaceově operátoru představuje v kombinaci s extrahovanými charakteristikami opravdu velmi přesnou a rychlou klasifikaci i u tohoto rozsáhlejšího datasetu. Vlnková transformace je pak nejpomalejším řešením, ale umožňuje různé přístupy k analýze, protože knihovna PyWavelets nabízí různé vlnkové rodiny, které mezi sebou můžeme porovnat a stanovit ty nejvhodnější pro zkoumanou detekci živosti.

Na základě obou experimentů můžeme stanovit závěrečný úsudek. Pro detekci živosti na našem bezdotykovém zařízení je nejlepším řešením použít pro nasvícování otisků prstů světlo červené barvy. Nejrychlejším a nejspolehlivějším řešením pro klasifikaci snímků je pak získat předzpracované snímky s využitím Sobelova a Laplaceova operátoru, z takto zpracovaných výsledků pak určit důležité vlastnosti pro GLCM matici a sestavit výsledný vektor. Poté je provedeno trénování neuronové sítě na vzorcích se známým výsledkem a výsledná predikce živosti a neživosti testovaných otisků prstů. Další možností je využití vlnkové transformace nebo lokálního binárního vzoru. Lokální binární vzor je druhou nejrychlejší metodou, jeho přesnost je opravdu pouze nepatrně nižší než u Sobelova a Laplaceova operátoru. Při využití vlnkové transformace je vhodné využít biortogonální spline vlnky nebo reverzní biortogonální spline vlnky. Správnost klasifikace je průměrně vyšší než 80%. Tato metoda je však nejpomalejším řešením, ale i tak poskytuje dobré výsledky a možné podklady pro další výzkum díky možnosti experimentovat s různými vlnkovými rodinami a parametry.

Kapitola 9

Závěr

Cílem mé bakalářské práce bylo vytvořit algoritmus, který by dokázal detekovat živost nebo neživost otisku prstu na užitém bezdotykovém zařízení. Pro výzkum jsem si vybrala algoritmy, které fungují jako dobré deskriptory textury obrazu – lokální binární vzor, Sobelův a Laplaceův operátor a vlnkovou transformaci. Velkou roli pro extrahované vektory hrála matice GLCM a její získané vlastnosti – kontrast, korelace, homogenita a energie. V rámci své práce jsem porovnála právě více extrahovaných vektorů na základě různých faktorů, stejně jako bylo porovnáno více metod určených pro trénování známých otisků prstů a testování neznámých vzorků – umělá neuronová síť, SVM a rozhodovací stromy. Bylo provedeno více experimentů, které byly poté detailně shrnuty. Algoritmus byl otestován na datasetu, který byl získán právě z daného bezdotykového zařízení.

Celkově nejlepší metodou pro klasifikaci byla umělá neuronová síť s přesností 89.4 % a nejlepším prahováním využitým při předzpracování otisku prstu Otsu prahování s průměrnou přesností 88.2 %. Oba vektory založené na lokálním binárním vzoru a Sobelově a Laplaceově operátoru vykazovaly průměrnou přesnost okolo 90 %, vektor založený na vlnkové transformaci klasifikoval přesně v 85 % případů. Snímky nasvícené červenou barvou světla měly v porovnání s modrým a zeleným světlem nejlepší přesnost pro klasifikaci testovaných vzorků – 90 %. Záměr nalezení algoritmu pro detekci živosti byl tedy splněn.

Mnoho prací pro detekci živosti otisků prstů se více zaměřuje na užití tradičních dotykových senzorů. Užití bezdotykového senzoru a představení získaných výsledků je tedy přispěním do výzkumu, který se věnuje této méně zastoupené problematice. Rozšíření možné budoucí práce by mohlo zahrnovat rozšíření datasetu snímků získaných z bezdotykového senzoru, použití dalších rodin vlnkové transformace nebo nasvícení snímků světlem o jiné vlnové délce.

Moje vedoucí práce Mona Heidari mi nabídla napsat článek pro konferenci ICBES'20 ohledně mé bakalářské práce a dosažených výsledků. Ačkoliv v době dokončování stále nevíme, jak článek dopadl v hodnocení a zpětné vazbě, jednalo se o novou zkušenost, protože psaní vědeckých článků není opravdu nic snadného, i co se týče pokročilého odborného projevu v angličtině. Původní podoba odeslaného článku na konferenci je zahrnuta na paměťovém médiu.

Práce mi umožnila více proniknout do problematiky biometrie a bioinformatiky, kterým se bakalářské studium příliš nevěnuje, a přesvědčila mě, že bych se tomuto zaměření chtěla věnovat i během magisterského studia, protože v něm vidím velký potenciál.

Literatura

- [1] ANISIMOVA, E., BEDNÁŘ, J. a PÁTA, P. Zpracování obrazu pomocí vlnkové transformace. *Elektrorevue*. 2013, sv. 15, č. 4, s. 238–246. ISSN 1213-1539.
- [2] BRID, R. S. Decision Trees — A simple way to visualize a decision. *Medium* [online]. 2018. Aktualizováno 26. 10. 2018 [cit. 10. dubna 2020]. Dostupné z: <https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb>.
- [3] BROWNLEE, J. Difference Between a Batch and an Epoch in a Neural Network. *Machine Learning Mastery* [online]. 2018. Aktualizováno 26. 10. 2019 [cit. 15. března 2020]. Dostupné z: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>.
- [4] BROWNLEE, J. Understand the Impact of Learning Rate on Neural Network Performance. *Machine Learning Mastery* [online]. 2019. Aktualizováno 6. 2. 2020 [cit. 16. března 2020]. Dostupné z: <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>.
- [5] BULTHEEL, A. a HUYBRECHS, D. *Wavelets with applications in signal and image processing* [online]. 2014 [cit. 2020-03-05]. Dostupné z: <https://people.cs.kuleuven.be/~daan.huybrechs/teaching/wavelets2014.pdf>.
- [6] CORTES, C. a VAPNIK, V. Support-Vector Networks. *Machine Learning*. 1995, sv. 9, č. 20, s. 273–297. ISSN 0885-6125.
- [7] DRAHANSKY, M., DOLEZEL, M., VANA, J., BREZINOVA, E., YIM, J. et al. New Optical Methods for Liveness Detection on Fingers. *BioMed Research International*. 2013, sv. 2013, s. 74–84. ISSN 2314-6133.
- [8] DRAHANSKÝ, M. *Biometrické systémy - Studijní opora* [online]. 2006 [cit. 2019-11-16]. Dostupné z: https://www.fit.vutbr.cz/study/courses/BIO/private/BIO_Studijni_opora.pdf.
- [9] DRAHANSKÝ, M., BŘEZINOVÁ, E., ORSÁG, F. a HEJTMÁNKOVÁ, D. Dermatologické faktory ovlivňující snímání otisků prstů. *Ministerstvo vnitra České republiky* [online]. 2010. Aktualizováno 17. 11. 2019 [cit. 17. listopadu 2019]. Dostupné z: <https://www.mvcr.cz/clanek/dermatologicke-faktory-ovlivnujici-snimani-otisku-prstu.aspx>.
- [10] DRAHANSKÝ, M. et al. *Advanced Biometric Technologies*. 1. vyd. InTech - Open Access Publisher, 2011. ISBN 978-953-307-487-0.

- [11] GAIKWAD, A. Evaluation of Fingerprint Identification Based on Local Binary Pattern (LBP). *International Journal for Research in Engineering Application & Management (IJREAM)*. 2018, sv. 4, č. 4, s. 395–400. ISSN 2454-9150.
- [12] GRAGNANIELLO, D. *Local image descriptors for biometric liveness detection* [online]. 2015 [cit. 2019-11-28]. Dostupné z: <https://pdfs.semanticscholar.org/e96d/5c8299ebfe60c9e4cc4fab07f2d7953908a0.pdf>.
- [13] GREENSTED, A. Otsu Thresholding. *The Lab Book Pages* [online]. 2010. Aktualizováno 17. 6. 2010 [cit. 3. března 2020]. Dostupné z: <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>.
- [14] HLAVÁČ, V. *Vlnková transformace* [online]. 2015 [cit. 2020-03-04]. Dostupné z: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZpr0br/14WaveletsCz.pdf>.
- [15] HONG, L., WAN, Y. a JAIN, A. Fingerprint Image Enhancement: Algorithm and Performance Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1998, sv. 20, č. 8, s. 777–789. ISSN 0182-8828.
- [16] HORÁK, K. *Cvičení 10 - Morfologické operace* [online]. 2017 [cit. 2019-11-21]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Exercise10/content_cz.php.
- [17] JAIN, A. K., FLYNN, P. a ROSS, A. A. *Handbook of Biometrics*. 1. vyd. Springer, 2008. ISBN 978-0-387-71040-2.
- [18] KULHARY, R. OpenCV – Overview. *GeeksforGeeks* [online]. Aktualizováno 12. 2. 2020 [cit. 12. února 2020]. Dostupné z: <https://www.geeksforgeeks.org/opencv-overview/>.
- [19] LEE, C., LEE, S. a KIM, J. A Study of Touchless Fingerprint Recognition System. In: YEUNG, D.-Y., KWOK, J. T., FRED, A., ROLI, F. a RIDDER, D. de, ed. *Structural, Syntactic, and Statistical Pattern Recognition - Joint IAPR International Workshops, SSPR 2006 and SPR 2006, Hong Kong, China, August 17-19, 2006. Proceedings*. Hong Kong, China: Springer, Srpen 2006, s. 358–365. ISBN 978-3540896883.
- [20] LI, S. Z. et al. *Encyclopedia of Biometrics*. 1. vyd. Springer, 2009. ISBN 978-0-387-73003-5.
- [21] MALTONI, D., MAIO, D., JAIN, A. K. a PRABHAKAR, S. *Handbook of Fingerprint Recognition*. 2. vyd. Springer, 2009. ISBN 978-1-84882-253-5.
- [22] MATHWORKS. Continuous Wavelet Transform and Scale-Based Analysis. *MATLAB R2019b - documentation* [online]. 2020. Aktualizováno 4. 3. 2020 [cit. 4. března 2020]. Dostupné z: <https://www.mathworks.com/help/wavelet/gs/continuous-wavelet-transform-and-scale-based-analysis.html>.
- [23] MATHWORKS. graycoprops - Properties of gray-level co-occurrence matrix. *MATLAB R2019b - documentation* [online]. 2020. Aktualizováno 17. 2. 2020 [cit. 17. února 2020]. Dostupné z: <https://www.mathworks.com/help/images/ref/graycoprops.html>.
- [24] MORSCHE, H. ter. *Wavelet basics* [online]. 2019 [cit. 2020-03-05]. Dostupné z: https://www.win.tue.nl/casa/meetings/seminar/previous/_index_files/wavelets1.pdf.

- [25] OPENCV. Morphological Transformations. *OpenCV - documentation* [online]. 2019. Aktualizováno 21. 11. 2019 [cit. 21. listopadu 2019]. Dostupné z: https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html.
- [26] OPENCV. Image Thresholding. *OpenCV - documentation* [online]. 2020. Aktualizováno 3. 3. 2020 [cit. 3. března 2020]. Dostupné z: https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html.
- [27] PARTHASARADHI, S., DERAKHSHANI, R. a HORNAK, L. A. Time-Series Detection of Perspiration as a Liveness Test in Fingerprint Devices. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 2005, sv. 8, č. 3, s. 335–343. ISSN 1094-6977.
- [28] PATEL, S. Chapter 2 : SVM (Support Vector Machine) — Theory. *Medium* [online]. 2017. Aktualizováno 2. 3. 2020 [cit. 2. března 2020]. Dostupné z: <https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72>.
- [29] PETROVICI, A. a LAZAR, C. *Identifying Fingerprint Alteration Using the Reliability Map of the Orientation Field* [online]. 2010 [cit. 2019-11-17]. Dostupné z: http://ace.ucv.ro/analele/2010_vol1/08Petrovici_Lazar.pdf.
- [30] POLICIE. Kriminální daktýloskopie. *Policie České republiky* [online]. 2019. Aktualizováno 2. 2. 2020 [cit. 2. února 2020]. Dostupné z: <https://www.policie.cz/clanek/kriminalisticka-daktyloskopie-266095.aspx>.
- [31] PUSULURI, S. Activation functions and weight initialization in deep learning. *Medium* [online]. 2019. Aktualizováno 17. 2. 2020 [cit. 17. února 2020]. Dostupné z: <https://medium.com/@sakeshpusuluri123/activation-functions-and-weight-initialization-in-deep-learning-ebc326e62a5c>.
- [32] PYWAVELETS. *PyWavelets - Wavelet Transforms in Python* [online]. 2020 [cit. 2020-03-10]. Dostupné z: <https://pywavelets.readthedocs.io/en/latest/>.
- [33] PYWAVELETS. *Wavelet Properties Browser* [online]. 2020. Aktualizováno 15. 3. 2020 [cit. 16. března 2020]. Dostupné z: <http://wavelets.pybytes.com/>.
- [34] RECOGTECH. FAR and FRR: security level versus user convenience. *Recogtech* [online]. Aktualizováno 11. 2. 2020 [cit. 11. února 2020]. Dostupné z: <https://www.recogtech.com/en/knowledge-base/security-level-versus-user-convenience>.
- [35] RUSSELL, I. *The Delta Rule* [online]. 2016 [cit. 2020-03-16]. Dostupné z: <https://web.archive.org/web/20160304032228/http://uhavax.hartford.edu/compsci/neural-networks-delta-rule.html>.
- [36] SCIKITLEARN. 1.4. Support Vector Machines. *Scikit-learn 0.22.2 - documentation* [online]. 2020. Aktualizováno 3. 3. 2020 [cit. 3. března 2020]. Dostupné z: <https://scikit-learn.org/stable/modules/svm.html>.
- [37] SCIKITLEARN. 1.10. Decision Trees. *Scikit-learn 0.22.2 - documentation* [online]. 2020. Aktualizováno 3. 3. 2020 [cit. 3. března 2020]. Dostupné z: <https://scikit-learn.org/stable/modules/tree.html>.

- [38] SINHA, U. The Sobel and Laplacian Edge Detectors. *AI Shack* [online]. 2017. Aktualizováno 11. 3. 2020 [cit. 11. března 2020]. Dostupné z: <https://www.aishack.in/tutorials/sobel-laplacian-edge-detectors/>.
- [39] SKIMAGE. Module: feature.texture - greycmatrix. *Skimage 0.7.0 - documentation* [online]. 2011. Aktualizováno 17. 2. 2020 [cit. 17. února 2020]. Dostupné z: <https://scikit-image.org/docs/0.7.0/api/skimage.feature.texture.html>.
- [40] SZEWCZYK, R., GRABOWSKI, K., NAPIERALSKA, M., SANKOWSKI, W., ZUBERT, M. et al. A reliable iris recognition algorithm based on reverse biorthogonal wavelet transform. *Pattern Recognition Letters*. 2012, sv. 33, č. 8, s. 1019–1026. ISSN 0167-8655.
- [41] VONESCH, C., BLU, T. a UNSER, M. Generalized biorthogonal Daubechies wavelets. In: PAPADAKIS, M., LAINE, A. F. a UNSER, M. A., ed. *Proceedings Volume 5914, Wavelets XI*. San Diego, USA: SPIE, 2005.
- [42] WU, Q., MERCHANT, F. a CASTLEMAN, K. *Microscope Image Processing*. Elsevier, 2008. ISBN 978-0-12-372578-3.
- [43] ZAGHETTO, C., MENDELSON, M., ZAGHETTO, A. a B. VIDAL, F. de. Liveness Detection on Touchless Fingerprint Devices Using Texture Descriptors and Artificial Neural Networks. In: IEEE. *2017 IEEE International Joint Conference on Biometrics (IJCB)*. Denver, USA: IEEE, říjen 2017, s. 406–412. ISBN 978-1-5386-1124-1.
- [44] ZBOŘIL, F. a ZBOŘIL, F. V. *Přednáška 9 - Rozpoznávání* [online]. 2019 [cit. 2020-02-17]. Dostupné z: https://www.fit.vutbr.cz/study/courses/IZU/private/1819izu_9.pdf.

Příloha A

Obsah přiloženého paměťového média

Na přiloženém paměťovém médiu se nachází:

- zdrojový kód bakalářské práce napsaný v jazyku \LaTeX ,
- text bakalářské práce ve formátu PDF,
- zdrojové kódy implementovaného programu,
- *Readme* s popisem práce s programem a postupem instalace potřebných knihoven,
- tabulky výsledků programu získaných během provedených experimentů ve formátu `xlsx`,
- původní článek odeslaný na konferenci ICBES'20.

Příloha B

Manuál

Pro správné fungování programu je nutno mít nainstalovaný Python 3.6 a následující knihovny:

- `numpy` (knihovna NumPy),
- `cv2` (knihovna OpenCV),
- `matplotlib`,
- `skimage` (knihovna scikit-image),
- `pywt` (knihovna PyWavelets),
- `sklearn` (knihovna scikit-learn).

Program se spouští ve tvaru `python3.6 bt.py arguments`, kde `arguments` jsou jednotlivé argumenty uvedené níže s jejich podrobným vysvětlením a funkcí.

Režim extrakce vektorů a klasifikace

Program je možné spustit v následujících variantách:

```
-test lbp -s otsu|gauss|mean -clasif ann|svm|dts -img b|g|r|all
```

Jedná se o extrakci vektorů a následnou klasifikaci na základě vektoru založeném na lokálním binárním vzoru. V tomto případě si uživatel vybere danou metodu segmentace s užitím různých typů prahování (`otsu` pro využití Otsu prahování, `gauss` pro využití adaptivního prahování Gaussian, `mean` pro adaptivní prahování Mean). Dále si uživatel vybere daný princip klasifikace – umělou neuronovou síť `ann`, SVM `svm` nebo rozhodovací stromy `dts`. Poslední a nedílnou součástí je určit, jaké snímky chceme analyzovat – nasvícené pouze modrým světlem (`b`), zeleným (`g`), červeným (`r`), nebo chceme zkoumat všechny snímky získaného datasetu (`all`).

```
-test sobel -clasif ann|svm|dts -img b|g|r|all
```

Tento příkaz programu je určený pro extrakci a klasifikaci pomocí Sobelova a Laplaceova operátoru. Přepínač `-s` tato metoda nevyužívá, segmentace je přímo provedena s využitím prahovací metody Otsu.

```
-test wavelet -t <type> -s otsu|gauss|mean -clasif ann|svm|dts -img b|g|r|all
```

Každý obraz otisku prstu je analyzován pomocí vlnkové transformace. Získaný vektor může být pomalejší pro získání zejména u velkých datasetů. Diskrétní vlnková transformace však využívá mnoho různých Wavelet rodin. Přepínač `-t` je určený pro nastavení dané vlnkové rodiny a typu vlnky. Všechny podporované vlnky implementovaného programu `type` a zároveň i všechny podporované vlnky knihovnou PyWavelets pro 2D diskrétní vlnkovou transformaci jsou uvedeny na konci tohoto manuálu.

Pořadí argumentů není volitelné.

Režim metody zobrazení jednotlivých technik

Cílem tohoto režimu uživateli snadno předložit různé využití techniky při analýze. Program je možné spustit následovně:

```
-show lbp [-s otsu|gauss|mean] -img <path>
```

Uživatel proměnnou `path` nahradí cestou k obrazu otisku prstu, který chce analyzovat. Výsledkem je zobrazení otisku prstu zpracovaného pomocí algoritmu LBP a histogram, který byl důležitou vlastností v případě extrakce vektoru pro analýzu.

```
-show sobel [-s otsu|gauss|mean] -img <path>
```

Tento příkaz umožní zobrazit okno se zpracovaným otiskem prstu pomocí Sobelova operátoru na základě osy x a y a výsledku aplikace Laplaceova operátoru.

```
-show wavelet -t <type> [-s otsu|gauss|mean] -img <path>
```

Provádí vlnkovou transformaci. Výsledkem je horizontální, vertikální a diagonální detail a jejich aproximace.

```
-show seg -s otsu|gauss|mean -img <path>
```

Zobrazí segmentovaný obraz s využitím daného typu prahování.

Segmentace je v tomto režimu povinná pouze v posledním případě.

Pořadí argumentů není volitelné.

Zobrazení nápovědy pro program

Následující argumenty umožní zobrazení jednoduché nápovědy pro uživatele:

```
-help
```

Zobrazí nápovědu pro celý program.

```
-test help
```

Zobrazí nápovědu pro argumenty týkající se Režimu extrakce vektorů a klasifikace.

```
-show help
```

Zobrazí nápovědu pro argumenty týkající se Režimu metody zobrazení jednotlivých technik.

Některé z příkladů spuštění

```
python3.6 bt.py -test lbp -s gauss -clasif ann -img r
```

Provede klasifikaci na základě vektoru založeném na lokálním binárním vzoru s využitím Gaussian prahování a finálního trénování a testování umělé neuronové sítě. Jsou využity jen snímky nasvícené červenou barvou světla.

```
python3.6 bt.py -test wavelet -t bior1.5 -s otsu -clasif svm -img b
```

Vlnka bior1.5 z rodiny biortogonálních spline vlnek je použita pro provedení vlnkové transformace pouze pro dataset obsahující otisky prstů nasvícené modrým světlem. Předzpracování je provedeno s Otsu prahováním. Pro klasifikaci je využito SVM.

```
python3.6 bt.py -test sobel -clasif ann -img all
```

Extrakce vektorů je provedena s využitím Sobelova a Laplaceova operátoru. Pro trénování a testování pomocí umělé neuronové sítě je využitý celý získaný dataset nasvícený různými barvami viditelného světla.

```
python3.6 bt.py -show seg -s otsu -img ./dataset/allTest/live99.jpg
```

Zobrazení postupu segmentace obrazu otisku prstu s využitím Otsu prahování. Uživateli je zobrazen původní normalizovaný obraz, výsledek aplikace prahování, extrahovaná maska využívající morfologických operací a výsledný segmentovaný otisk prstu.

```
python3.6 bt.py -show sobel -s otsu -img ./dataset/allTrain/fake69.jpg
```

Daný falešný otisk prstu je segmentován s využitím Otsu prahování a uživateli je zobrazen původní normalizovaný obraz, výsledek aplikace Laplaceova operátoru a Sobelova operátoru pro osy x a y .

Podporované vlnky

Knihovna PyWavelets podporuje následující vlnkové rodiny a jejich příslušné vlnky pro diskrétní vlnkovou transformaci:

- biortogonální spline vlnky (*Biorthogonal Spline Wavelets*) – bior1.1, **bior1.3**, **bior1.5**, bior2.2, **bior2.4**, bior2.6, bior2.8, bior3.1, bior3.3, bior3.5, bior3.7, bior3.9, bior4.4, bior5.5, bior6.8,
- coiflety (*Coiflets*) – coif1, coif2, coif3, coif4, coif5, coif6, coif7, coif8, coif9, coif10, coif11, coif12, coif13, coif14, coif15, coif16, coif17,
- vlnky Daubechies (*Daubechies wavelets*) – db1, **db2**, db3, **db4**, db5, db6, db7, db8, db9, db10, db11, db12, db13, db14, db15, db16, db17, db18, db19, db20, db21, db22, db23, db24, db25, db26, db27, db28, db29, db30, db31, db32, db33, db34, db35, db36, db37, db38,
- diskrétní Meyerova vlnka (*Discrete Meyer Wavelet*) – dmey,
- Haarova vlnka (*Haar Wavelet*) – haar,

- reverzní biortogonální spline vlnky (*Reverse Biorthogonal Spline Wavelets*) – rbio1.1, rbio1.3, rbio1.5, rbio2.2, rbio2.4, rbio2.6, rbio2.8, **rbio3.1**, rbio3.3, rbio3.5, rbio3.7, rbio3.9, rbio4.4, rbio5.5, rbio6.8,
- symlety (*Symlets*) – sym2, sym3, sym4, sym5, sym6, sym7, sym8, sym9, sym10, sym11, sym12, sym13, sym14, sym15, sym16, sym17, sym18, sym19, sym20.

Tučně jsou vyznačeny vlnky, se kterými bylo v průběhu práce experimentováno.

Postup instalace potřebných knihoven

Předpokládejme, že máme v systému nainstalovaný Python 3.6. Dostupnost můžeme zkontrolovat pomocí příkazu `python3.6 --version`.

Dále potřebujeme nainstalovat další používané knihovny. Pro instalaci využijeme příkazu `pip3`, který musíme ovšem nainstalovat. Instalace je provedena příkazem:

```
sudo apt install python3-pip
```

Instalace použitých knihoven probíhá následovně. Další knihovny např. NumPy nebo Matplotlib, by se již měly nainstalovat zároveň automaticky.

```
pip3 install opencv-python
pip3 install scikit-image
pip3 install scikit-learn
pip3 install PyWavelets
```

Příloha C

Výsledky analýzy datasetů otisků prstů nasvícených stejnou barvou světla

Segmentace	Klasifikace	Modré světlo	Zelené světlo	Červené světlo
Otsu	ANN	88.888889	96.428571	92.592593
	SVM	92.592593	82.142857	81.481481
	DTs	88.888889	92.857143	92.592593
Gaussian	ANN	96.296296	96.428571	100.000000
	SVM	92.592593	57.142857	100.000000
	DTs	88.888889	71.428571	100.000000
Mean	ANN	96.296296	92.857143	100.000000
	SVM	92.592593	71.428571	100.000000
	DTs	85.185185	85.714286	100.000000

Tabulka C.1: Srovnání přesnosti v % pro metodu LBP

Segmentace	Klasifikace	Modré světlo	Zelené světlo	Červené světlo
Otsu	ANN	96.296296	92.857143	100.000000
	SVM	92.592593	78.571429	85.185185
	DTs	81.481481	96.428571	88.888889

Tabulka C.2: Srovnání přesnosti v % pro metodu využívající Sobelův a Laplaceův operátor

Segmentace	Klasifikace	Modré světlo	Zelené světlo	Červené světlo
Otsu	ANN	92.592593	100.000000	88.888889
	SVM	96.296296	85.714286	81.481481
	DTs	92.592593	75.000000	88.888889
Gaussian	ANN	88.888889	89.285714	88.888889
	SVM	85.185185	71.428571	92.592593
	DTs	85.185185	78.571429	74.074074
Mean	ANN	85.185185	67.857143	96.296296
	SVM	88.888889	71.428571	88.888889
	DTs	77.777778	75.000000	96.296296

Tabulka C.3: Srovnání přesnosti v % pro metodu bior1.3

Segmentace	Klasifikace	Modré světlo	Zelené světlo	Červené světlo
Otsu	ANN	92.592593	92.857143	96.296296
	SVM	88.888889	85.714286	85.185185
	DTs	77.777778	82.142857	81.481481
Gaussian	ANN	74.074074	85.714286	100.000000
	SVM	85.185185	89.285714	81.481481
	DTs	62.962963	78.571429	74.074074
Mean	ANN	74.074074	71.428571	92.592593
	SVM	92.592593	78.571429	81.481481
	DTs	74.074074	82.142857	81.481481

Tabulka C.4: Srovnání přesnosti v % pro metodu db2

Segmentace	Klasifikace	Modré světlo	Zelené světlo	Červené světlo
Otsu	ANN	96.296296	92.857143	96.296296
	SVM	92.592593	82.142857	85.185185
	DTs	92.592593	85.714286	88.888889
Gaussian	ANN	92.592593	71.428571	92.592593
	SVM	81.481481	78.571429	88.888889
	DTs	66.666667	92.857143	92.592593
Mean	ANN	92.592593	78.571429	96.296296
	SVM	85.185185	75.000000	96.296296
	DTs	77.777778	60.714286	96.296296

Tabulka C.5: Srovnání přesnosti v % pro metodu rbio3.1

Segmentace	Klasifikace	Modré světlo	Zelené světlo	Červené světlo
Otsu	ANN	96.296296	92.857143	92.592593
	SVM	88.888889	85.714286	88.888889
	DTs	74.074074	71.428571	74.074074
Gaussian	ANN	92.592593	67.857143	100.000000
	SVM	81.481481	60.714286	85.185185
	DTs	81.481481	67.857143	100.000000
Mean	ANN	92.592593	78.571429	96.296296
	SVM	92.592593	78.571429	92.592593
	DTs	88.888889	60.714286	96.296296

Tabulka C.6: Srovnání přesnosti v % pro metodu bior2.4

Segmentace	Klasifikace	Modré světlo	Zelené světlo	Červené světlo
Otsu	ANN	92.592593	96.428571	92.592593
	SVM	96.296296	89.285714	81.481481
	DTs	85.185185	89.285714	92.592593
Gaussian	ANN	85.185185	78.571429	88.888889
	SVM	85.185185	71.428571	92.592593
	DTs	77.777778	71.428571	66.666667
Mean	ANN	81.481481	75.000000	96.296296
	SVM	85.185185	75.000000	88.888889
	DTs	88.888889	82.142857	92.592593

Tabulka C.7: Srovnání přesnosti v % pro metodu bior1.5

Segmentace	Klasifikace	Modré světlo	Zelené světlo	Červené světlo
Otsu	ANN	100.000000	92.857143	92.592593
	SVM	88.888889	85.714286	74.074074
	DTs	88.888889	78.571429	92.592593
Gaussian	ANN	70.370370	89.285714	85.185185
	SVM	70.370370	71.428571	85.185185
	DTs	70.370370	67.857143	81.481481
Mean	ANN	100.000000	85.714286	100.000000
	SVM	88.888889	89.285714	74.074074
	DTs	96.296296	78.571429	96.296296

Tabulka C.8: Srovnání přesnosti v % pro metodu db4

Příloha D

Výsledky analýzy datasetu obsahující všechny získané snímky otisků prstů

Segmentace	Klasifikace	LBP	Sobelův a Laplaceův operátor
Otsu	ANN	93.589744	97.435897
	SVM	88.461538	84.615385
	DTs	93.589744	96.153846
Gaussian	ANN	93.589744	
	SVM	83.333333	
	DTs	87.179487	
Mean	ANN	94.871795	
	SVM	80.769231	
	DTs	88.461538	

Tabulka D.1: Srovnání přesnosti v % pro metody LBP a Sobelův a Laplaceův operátor

Segmentace	Klasifikace	bior1.3	db2
Otsu	ANN	89.743590	91.025641
	SVM	89.743590	88.461538
	DTs	85.897436	83.333333
Gaussian	ANN	87.179487	78.205128
	SVM	88.461538	75.641026
	DTs	82.051282	74.358974
Mean	ANN	88.461538	84.615385
	SVM	80.769231	83.333333
	DTs	84.615385	76.923077

Tabulka D.2: Srovnání přesnosti v % pro metody bior1.3 a db2

Segmentace	Klasifikace	rbio3.1	bior2.4
Otsu	ANN	88.461538	93.589744
	SVM	85.897436	88.461538
	DTs	87.179487	83.333333
Gaussian	ANN	87.179487	84.615385
	SVM	85.897436	80.769231
	DTs	85.897436	76.923077
Mean	ANN	89.743590	89.743590
	SVM	85.897436	88.461538
	DTs	85.897436	80.769231

Tabulka D.3: Srovnání přesnosti v % pro metody rbio3.1 a bior2.4

Segmentace	Klasifikace	bior1.5	db4
Otsu	ANN	89.743590	85.897436
	SVM	87.179487	84.615385
	DTs	85.897436	78.205128
Gaussian	ANN	82.051282	88.461538
	SVM	87.179487	85.897436
	DTs	76.923077	78.205128
Mean	ANN	87.179487	87.179487
	SVM	79.487179	87.179487
	DTs	82.051282	85.897436

Tabulka D.4: Srovnání přesnosti v % pro metody bior1.5 a db4