



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**REKONSTRUKCE POŠKOZENÉ ČÁSTI OTISKU PRSTŮ
S VYUŽITÍM NEURONOVÝCH SÍTÍ**

RECONSTRUCTION OF DAMAGED PARTS OF FINGERPRINT IMAGE USING NEURAL NETS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

BORIS BOBOCKÝ

VEDOUcí PRÁCE

SUPERVISOR

Ing. ONDŘEJ KANICH, Ph.D.

BRNO 2023

Zadání bakalářské práce



148339

Ústav: Ústav inteligentních systémů (UITS)
Student: **Bobocký Boris**
Program: Informační technologie
Specializace: Informační technologie
Název: **Rekonstrukce poškozené části otisku prstů s využitím neuronových sítí**
Kategorie: Zpracování obrazu
Akademický rok: 2022/23

Zadání:

1. Prostudujte literaturu týkající se rozpoznávání podle otisků prstů, soustředte se na kroky zvyšující kvalitu otisku. Seznamte se s metodami pro rekonstrukci případně generování syntetických otisků prstů a detekci poškození.
2. Lokalizujte poškozené oblasti otisku prstu. Navrhněte metody pro rekonstrukci otisku prstu v lokalizovaných oblastech s využitím neuronových sítí.
3. Implementujte navržené algoritmy z předchozího bodu.
4. Otestujte algoritmy na vhodné databázi otisků prstů s různou úrovní poškození. Analyzujte přesnost provedené rekonstrukce a změnu kvality otisku prstu před a po rekonstrukci.
5. Dosažené výsledky shrňte a diskutujte. Uveďte možná rozšíření vašeho řešení.

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.
- Barotová, Š.: *Detector of Skin Diseases by Fingerprint Technology*, 2017. Bachelor's thesis. FIT BUT in Brno, Brno.
- Drahanský, M.: *Hand-Based Biometrics: Methods and technology*, IET 2018, p. 430, ISBN 978-1-78561-224-4.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2.

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

Vedoucí práce: **Kanich Ondřej, Ing., Ph.D.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1.11.2022
Termín pro odevzdání: 10.5.2023
Datum schválení: 3.11.2022

Abstrakt

V tejto práci navrhujem spôsob rekonštrukcie poškodených odtlačkov s využitím generatívnych protichodných sietí (GAN), implementovaných s Pythonom a knižnicou PyTorch. Vytrénoval som špecifický GAN model na približne dvadsaťtisícovom datase, vytvorenom nástrojom Anguli a ďalšími nástrojmi na simuláciu poškodenia. Tento prístup produkoval vynikajúce výsledky a mohol by mať široké uplatnenie v biometrických systémoch. Táto práca zdôrazňuje potenciál hlbokého učenia v oblastiach rekonštrukcie obrazu a biometrie.

Abstract

In this paper, I propose a method for reconstructing damaged fingerprints using generative adversarial networks (GANs), implemented with Python and the PyTorch library. I have trained a specific GAN model on a dataset of approximately twenty thousand prints, created with Anguli and other damage simulation tools. This approach produced excellent results and could have wide application in biometric systems. This work highlights the potential of deep learning in the fields of image reconstruction and biometrics.

Klíčové slová

rekonštrukcia, poškodené odtlačky, lokalizácia poškodenia, odtlačky, neuronové siete, GAN, rekonštrukcia obrazu

Keywords

reconstruction, damaged fingerprints, damage localization, fingerprints, neural networks, GAN, image reconstruction

Citácia

BOBOCKÝ, Boris. *Rekonstrukce poškozené části otisku prstů s využitím neuronových sítí*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ondřej Kanich, Ph.D.

Rekonstrukce poškozené části otisku prstů s využitím neuronových sítí

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána doktora Ondřeja Kanicha. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, ze ktorých som čerpal.

.....
Boris Bobocký
23. mája 2023

Podakovanie

Chcel by som poďakovať pánovi doktorovi Ondřeji Kanichovi za to, že viedol túto prácu a svojej rodine, partnerke a priateľom za podporu.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 2 |
| 2 | Odtlačky prstov | 3 |
| 2.1 | Zloženie odtlačkov | 3 |
| 2.2 | Rozpoznávanie podľa odtlačkov prstov | 5 |
| 2.3 | Prejav kožných ochorení na odtlačkoch prstov | 6 |
| 2.4 | Syntetické odtlačky prstov | 7 |
| 2.5 | Generácia poškodenia syntetických odtlačkov | 8 |
| 3 | Kvalita odtlačkov prstov | 10 |
| 3.1 | Evaluácia kvality | 11 |
| 3.2 | Postupy zvýšenia kvality obrazu | 12 |
| 3.3 | Lokalizácia poškodenia odtlačku | 16 |
| 4 | Neurónové siete | 19 |
| 4.1 | Konvolučné neurónové siete | 20 |
| 4.2 | Generatívne protichodné siete | 25 |
| 5 | Návrh a implementácia | 28 |
| 5.1 | Generátor a Diskriminátor | 29 |
| 5.2 | Tréning a optimalizácia | 30 |
| 5.3 | Finálny nástroj | 32 |
| 6 | Testovanie a analýza | 33 |
| 6.1 | Vizuálna analýza | 34 |
| 6.2 | Analýza kvality odtlačkov | 34 |
| 6.3 | Porovnanie presnosti rekonštrukcie | 36 |
| 7 | Záver | 38 |
| | Literatúra | 39 |

Kapitola 1

Úvod

System rozpoznávania odtlačkov prstov sa stal dôležitou súčasťou mnohých aplikácií, čo zdôrazňuje význam biometrickej identifikácie v dnešnom svete. Od autentifikácie v smartpho-
noch, forenzné vedy, či bezpečnostné systémy a imigračné kontroly, to všetko je postavené na jedinečných vzoroch, ktoré sa nachádzajú na odtlačkoch našich prstov. Avšak nízka kvalita obrazu alebo poškodenie odtlačku sú jedny z problémov, ktoré aj naďalej bránia optimálnej funkčnosti týchto systémov.

Potencionálnym riešením tohto problému môže byť využitie umelej inteligencie(AI). Jej rast a zvyšujúca popularita viedli k výraznému pokroku a pozoruhodnému potenciálu v rôznych oblastiach. Jedným z významných momentov v tejto oblasti je vznik generatívnych adverzných sietí (GAN), relatívne nového konceptu, ktorý je obrovským príslubom najmä v oblasti počítačového videnia. V roku 2014 ho vo svojej práci predstavil Ian Goodfellow a jeho kolegovia. [14]

GAN spôsobili revolúciu v spôsobe, akým stroje vnímajú a interpretujú vizuálne informácie, čo predstavuje nevídané možnosti pokroku v oblasti generovania, rozpoznávania a analýzy nie len obrazu. Aktuálne už existuje rozsiahle množstvo GAN architektúr ako napríklad CycleGAN, StyleGAN, pixelRNN. Tieto modely sú schopné vytvárať alebo modifikovať realistické obrazy alebo audio, zýšiť kvalitu fotiek, alebo generovať video.

Táto bakalárska práca predstavuje inovatívny prístup k riešeniu problému rekonštrukcie poškodených častí obrazov odtlačkov prstov pomocou strojového učenia, konkrétne kontextových generatívnych adverzných sietí (contextual GAN). V práci sa navrhuje využitie týchto sietí na zlepšenie kvality odtlačkov prstov, čo má vplyv na spoľahlivosť biometrických systémov.

Obsah práce je štruktúrovaný do niekoľkých kapitol, ktoré poskytujú úvod do odtlačkov prstov a ich úlohy v biometrickej identifikácii. Sú tu aj vysvetlené výzvy spojené s poškodenými obrazmi odtlačkov prstov a dôležitosť syntetických odtlačkov pri vývoji a testovaní biometrických systémov. Ďalej sa diskutuje o kvalite obrazu a faktoroch, ktoré ju ovplyvňujú, vrátane nástrojov na detekciu poškodenia. Ďalšia časť práce sa zameriava na metódy zlepšovania kvality obrazu odtlačkov prstov.

Druhá polovica práce sa venuje umelým neurónovým sieťam a architektúre generatívnych adverzných sietí (GAN). Je popísaný návrh na rekonštrukciu poškodených oblastí odtlačkov prstov pomocou kontextových GAN sietí. Sú uvedené podrobnosti implementácie, ako je príprava datasetu, zvolená knižnica pre tvorbu sietí, proces tréningu a ladenia. Na záver sú zhrnuté dosiahnuté výsledky a diskutované možné implementačné a vylepšovacie postupy.

Kapitola 2

Odtlačky prstov

Odtlačky prstov patria dnes medzi jedny z najpoužívanějších biometrických vlastností. Ich prvotné využitie na identifikáciu osôb predstavil vo svojej práci Henry Faulds už v roku 1880 [17]. Neskôr sa ich vlastnosť unikátnosti začala využívať vo forenznej analýze. K roku 2008 spracovala databáza FBI IAFIS priemerne 63 tisíc odtlačkov denne [20]. Dnes sa však vo veľkom využívajú na autorizáciu do telefónov, počítačov, digitálnych zámkov a mnoho ďalšieho. Z tohto dôvodu sa kladie čoraz väčší dôraz na spracovanie odtlačku a presnosť porovnávacích algoritmov.

2.1 Zloženie odtlačkov

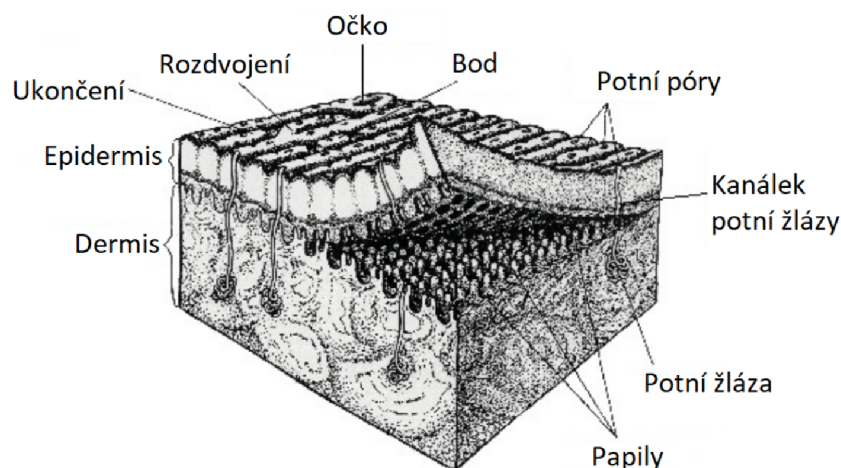
Odtlačky sa formujú do 6. mesiaca vývinu plodu a po zvyšok života ostávajú nemenné, to je jedna z kritických vlastností pre biometrický identifikátor.

Skladajú sa z dvoch hlavných vrstiev epidermis a dermis. Epidermis tvorí vonkajší vode odolný ochranný obal a je zodpovedný za obnovu kožných buniek. Keďže neobsahuje žiadne krvné cievy, získava živiny a kyslík difúziou z dermy. [23]

Dermis označuje hrubú vrstvu živých buniek pod epidermou, ktorá obsahuje krvné cievy, nervové zakončenia, potné žľazy a vlasové folikuly (nie však na odtlačkoch prstov). Dodáva pokožke rozťažnosť, pevnosť a pružnosť. [23]

Pre nás najpodstatnejšia časť sú papily nachádzajúce sa vo vrchnej vrstve dermis, ktoré tvoria papilárne línie. Výška papilárnych línií sa pohybuje v rozmedzí 0,1 - 0,4 mm a šírka v rozmedzí 0,2 - 0,5 mm. [11] Na povrchu brušiek prstov tvoria štruktúry, ktoré sú pre každý odtlačok unikátne.

Na obrázku 2.1 nižšie vidíme prierez kožou v oblasti bruška prstu. Vidíme na ňom dve hlavné vrstvy a medzi nimi ukotvené papily. Okrem nich môžeme vidieť potné žľazy, potné kanáliky, potné póry, ale aj základné markanty ako sú očko, bod, ukončenie alebo rozdvojenie, ktorým sa budeme podrobnejšie venovať v sekcii nižšie.

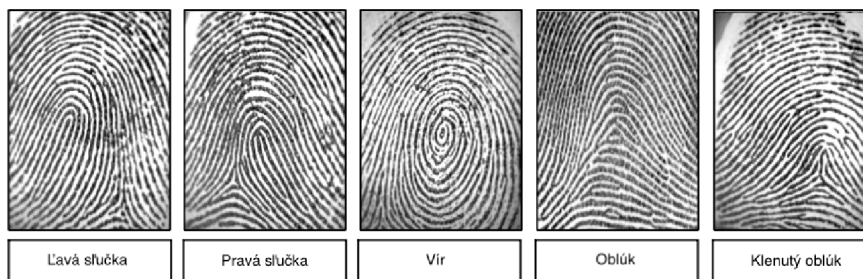


Obrázok 2.1: Prierez kožou zobrazujúci jednotlivé vrstvy a zložky. [11]

2.1.1 Triedy odtlačkov prstov

Každý deň sa zhromažďujú a ukladajú veľké objemy odtlačkov prstov v širokej škále aplikácií, najmä vo forenzných a vládnych aplikáciách. Digitálna identifikácia na základe odtlačkov vyžaduje porovnanie vstupného odtlačku prsta s veľkým počtom odtlačkov uložených v databáze. Aby sa skrátil čas vyhľadávania a výpočtová zložitosť, je potrebné odtlačky klasifikovať do tried na základe konzistentných znakov. [24]

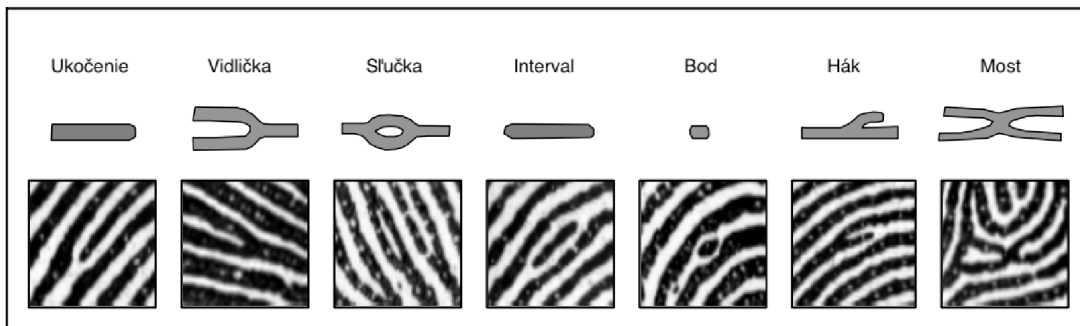
Známou klasifikáciu odtlačkov prstov navrhol Edward Henry, ktorá pozostáva z piatich hlavných tried: vír, ľavá slučka, pravá slučka, oblúk a klenutý oblúk. Príklad týchto tried vidíme na obrázku 2.2 [24]



Obrázok 2.2: Päť základných tried podľa Edwarda Henryho. [24]

2.1.2 Markanty

Po identifikácii triedy odtlačku môžeme prejsť k samotnému porovnávaniu. Väčšina dnešných systémov odtlačky porovnáva na základe útvarov tvorených hrebeňmi (vyvýšené časti papilárnych línií) a údoliami, ktoré nazývame markanty. V kontexte odtlačkov prstov označujú rôzne spôsoby, ako môžu byť hrebene nespojité. Sir Francis Galton (1822-1911) bol prvou osobou, ktorá kategorizovala markanty a zistila, že zostávajú nezmenené počas celého života jednotlivca. [24][31]



Obrázok 2.3: Sedem najčastejších markantov. [24]

Ako vidieť na obrázku 2.3, rozlišujeme mnoho druhov a v daktyloskopii sa využívajú všetky, no pri väčšine prístupových systémoch sa využívajú dva hlavné markanty, a to sú ukončenie a vidlička.

2.2 Rozpoznávanie podľa odtlačkov prstov

Skôr než sme schopní s odtlačkom porovnať, musíme surový obraz spracovať a previesť na čierno-bielý obraz obsahujúci len papilárne línie, kde každá línia má hrúbku presne jedného pixelu. Z takto spracovaného obrazu môžeme extrahovať markanty potrebné pre daný algoritmus. Podrobný postup je popísaný nižšie a vyobrazený na obrázku 2.4.

1. **Vstupný obraz.** Prvým krokom je zosnímanie fyzického odtlačku pomocou senzoru a jeho prevedenie do digitálnej formy. Senzorov poznáme viacero druhov a každý má svoje výhody aj nevýhody, líšia sa hlavne rozlíšením, snímacou plochou, geometrickou presnosťou, vzniknutým šumom či kvalitou obrazu. Medzi hlavné typy snímačov odtlačkov patria: optické, kapacitné, ultrazvukové a termické. [11]
2. **Pole orientácií.** Cieľom tohto kroku je získať pole orientácií hrebeňov odtlačku. Prvým krokom k získaniu tohto pola bude, že v každom bode odtlačku spočítame orientáciu hrebeňa papilárnej línie (orientácia môže nadobudnúť hodnotu 0 až 180°). V druhom kroku transformujeme tieto hodnoty do blokov, čím vznikne blokové pole orientácií. Finálnym krokom je nanosenie tohto pola späť na odtlačok prsta. [11]
3. **Extrahovanie línií.** V tomto bode spracovania môžu prebiehať ďalšie úpravy obrazu za cieľom zvýšiť finálnu presnosť porovnávacej operácie, týmto postupom sa budeme podrobne venovať v sekcii 3.4.

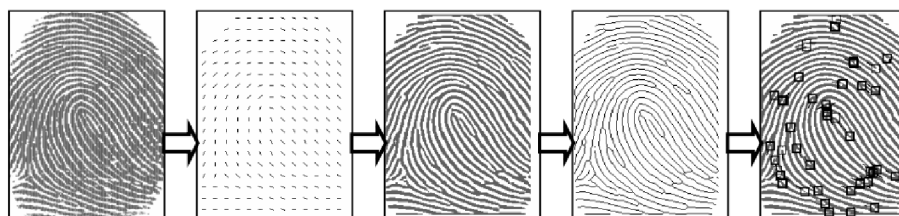
Cieľom kroku extrahovania línií je získať čiernobiely obraz obsahujúci len papilárne línie. Rôzne časti obrazu môžu disponovať odlišnými intenzitami šedej (dôvodom môže byť odlišnosť v jase či kontraste alebo šum zo snímača), preto nám jeden univerzálny prah na celý obraz nebude stačiť. Riešením je rátať prah lokálne, obvykle na poli o veľkosti niekoľkých pixelov. [31][24]

4. **Zúženie línií.** V poslednom kroku sme získali čierno-biely obraz odtlačku. Teraz je potrebné stenčiť čiary papilárnych línií na šírku jedného pixelu. Tento proces nie je príliš komplikovaný, avšak musí sa dbať na to, aby žiadne papilárne línie nezaniikli ani nezmenili svoj tvar.
5. **Extrahovanie markantov.** Posledným krokom je extrahovať z binárneho a zúženého obrazu markanty. Vo väčšine automatických systémoch nám stačí detektovať konkrétne dva markanty, sú to ukončenie a vidlička. Na ich extrakciu slúži vzorec 2.1, ktorý sa aplikuje na body patriace hrebeňom papilárnych línií, kde p_0, p_1, \dots, p_7 sú susedné pixely k pixelu \mathbf{p} a $\text{val}(\mathbf{p}) \in \{0, 1\}$ [24]

$$cn(p) = \frac{1}{2} \sum_{i=1..8} |\text{val}(p_{i \bmod 8}) - \text{val}(p_{i-1})| \quad (2.1)$$

Pre $cn(\mathbf{p})$ platí:

- $cn(\mathbf{p}) = 1$,
- $cn(\mathbf{p}) = 2$, bod sa nachádza v strede hrebeňa
- $cn(\mathbf{p}) = 3$, v bode vzniká rozdvojenie hrebeňov (markant vidličky)
- $cn(\mathbf{p}) > 3$, jedná sa o zložitejší markant

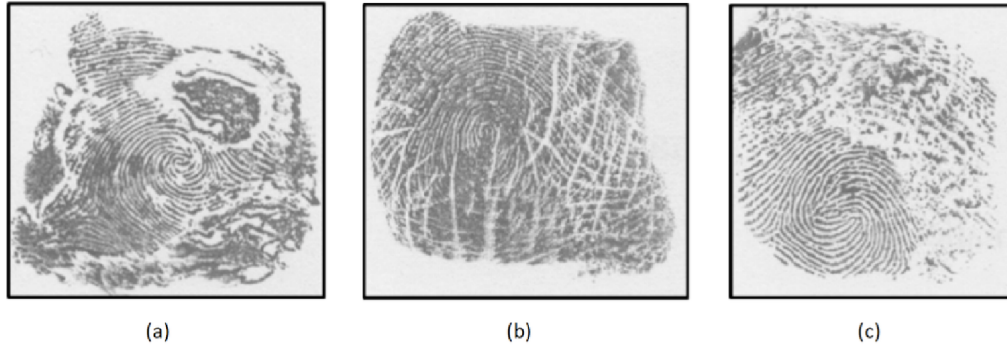


Obrázok 2.4: Jednotlivé fázy spracovania odtlačku prstu. [10]

2.3 Prejav kožných ochorení na odtlačkoch prstov

Kožné ochorenia môžu nepriaznivo ovplyvniť proces porovnávania odtlačkov. K dnešnému dňu vieme o desiatkach typov týchto ochorení. Počet ľudí trpiacimi týmito chorobami je nejasný, keďže sa nemusia vždy prejavíť príznaky a v určitých fázach prejavu nijak nezhoršujú životnú úroveň postihnutých jedincov. Medzi najčastejšie sa vyskytujúce choroby z pohľadu odtlačkov prstov patri hyperkeratotický ekzém, dyshidrotický ekzém, tinea manus, pyoderma, verruca vulgaris (pravé bradavice) a mnohé ďalšie. Tieto choroby môžu poškodiť štruktúru papilárnych línií a zmeniť farbu pokožky, čím sťažujú proces porovnávania.

Pokiaľ choroba poškodí aj nižšiu vrstvu kože (dermis), papilárne línie sa nezregenerujú do pôvodného stavu, ak sa zregenerujú vôbec. Príklad takto poškodených odtlačkov môžeme vidieť na obrázku 2.5. Táto práca sa bude zaoberať metódami, ako zvýšiť kvalitu obrazu takto poškodených odtlačkov a spôsobom ako poškodené oblasti zrekonštruovať. [8][22]



Obrázok 2.5: Obrázky znázorňujúce prejavy kožných ochorení. a) Psoriasis vulgaris, b) Kolagenóza, c) Hyperkeratotický ekzém. [9]

2.4 Syntetické odtlačky prstov

Pri vývoji, testovaní a evaluácii algoritmov na porovnávanie odtlačkov prstov je často potrebné mať veľké množstvo dát. To je však veľmi ťažké získať. Na príčine je viacero faktorov, ako vysoká cena existujúcich databáz, časová aj finančná náročnosť manuálneho zozbierania dostatočného počtu odtlačkov, ktoré navyše komplikujú zákony o ochrane osobných údajov. Ak by sa nám aj podarilo tieto prekážky prekonať, je následne potrebné tieto dáta popísať a kategorizovať. Väčšinu týchto problémov sme schopní obísť vytvorením realisticky vyzerajúcich syntetických odtlačkov. Výhodou týchto odtlačkov je, že sme schopní špecifikovať konkrétne vlastnosti, ktoré pre našu aplikáciu vyžadujeme ako napríklad generácia len konkrétneho prstu, triedy, kvality alebo poškodenia odtlačku. Realistický syntetický odtlačok by mal spĺňať tieto vlastnosti [24]:

- Rôzna hrúbka hrebeňov spôsobená vlhkosťou kože alebo intenzitou tlaku na skener.
- Malé fyzické poškodenia na bruškách prstov.
- Komplexné pozadie, ktoré vzniká pri snímaní prstu.
- Rôzne dotykové oblasti.

Populárnym prístupom ku generácii syntetických odtlačkov prstov je **SFinGe** [6] navrhnutá Cappellim, Maiom a Maltonim v roku 2001. Je dostupná aj ako aplikácia a schopná generovať rozsiahle databázy odtlačkov, rôzne zábery totožného odtlačku, realistické pozadie, rôzne typy a úrovne šumov a mnohé ďalšie prvky. Nástroje, ktoré v tejto práci budem používať a budú popísané nižšie, taktiež vychádzajú z tejto metódy. Proces generácie sa skladá z 10 častí, kde kroky 1-4 vytvoria hlavnú a nemennú kostru odtlačku a kroky 5-10 z nej vytvárajú rôzne variácie finálneho syntetického obrazu. Tieto kroky sú: [16]

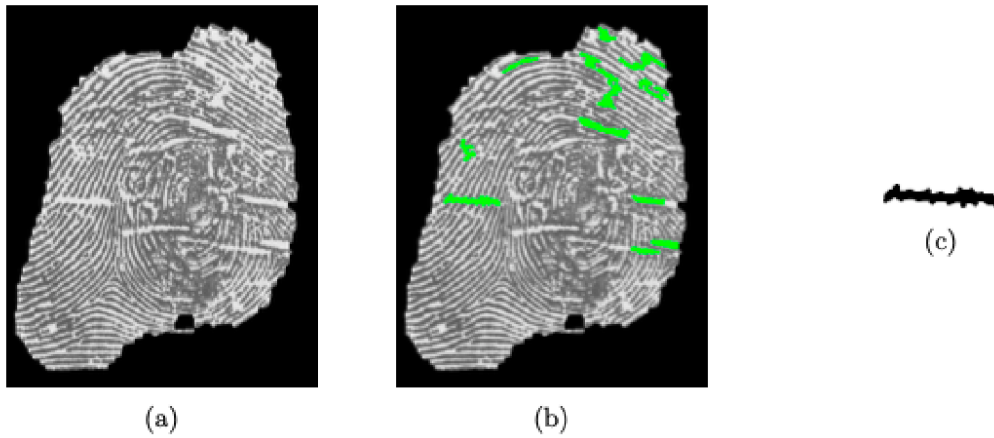
1. Určenie celkového tvaru odtlačku modelom založeným na eliptických segmentoch.

2. Získanie pola orientácií. To získame tak, že náhodne zvolíme triedu odtlačku, ktorá nám umožní zvoliť pozíciu singularít v odtlačku. S pomocou týchto znalostí sme schopní vygenerovať pole orientácií.
3. Získanie pola frekvencií. Generácia pola orientácií je založená na heuristike získanej z analýzy veľkého počtu reálnych odtlačkov.
4. Vytvorenie hlavnej kostry odtlačku. Z vyššie získaných vstupných dát sa vytvorí vzorová štruktúra papilárnych línií. Na generovanie sa používa iteratívna metóda využívajúca Gaborov filter, ktorý je aplikovaný na každý pixel. Obraz môže byť generovaný simultánne z viacerých bodov, čím vieme ovplyvniť výsledný počet markantov.
5. Pokiaľ máme hlavný odtlačok vytvorený aplikáciou krokov 5-10, môžeme generovať jeho variácie s ľubovoľnými vlastnosťami. Tieto kroky generujú:
 - Výber kontaktnej plochy so senzorom.
 - Zmena hrúbky papilárnych línií. Týmto krokom simulujeme rôzne úrovne vlhkosti a tlaku v odtlačku.
 - Skreslenie odtlačkov prstov. Tento krok sa využíva na simuláciu elastickej vlastnosti pokožky.
 - Šum a vykresľovanie. Simuluje zašumenie šedo-tónového obrazu, ktoré sa vyskytuje aj pri reálnych snímkoch odtlačku.
 - Posun a rotácia. V reálnom svete sú odtlačky len zriedka centrovane a rovné. Preto sa obrázok náhodne otočí a posunie.
 - Generovanie pozadia. Na základe toho z akého typu snímača chceme odtlačky simulovať, vytvoríme pozadie zodpovedajúce tomu, aké generuje tento snímač.

2.5 Generácia poškodenia syntetických odtlačkov

Pre návrh, testovanie a ladenie algoritmov slúžiacich na opravu poškodených odtlačkov je ideálne mať databázu s rôznymi stupňami prejavu choroby. Na generovanie databázy poškodených syntetických odtlačkov budem v tejto práci využívať nástroj z bakalárskej práce od Dávida Košťáka, ktorý umožňuje špecifikovať počet znakov poškodenia v odtlačku. Tento nástroj vkladá do syntetických odtlačkov vytvorených pomocou aplikácie Anguli reálne príznaky chorôb: atopický ekzém a psoriáza. Generátor poškodenia je naprogramovaný v jazyku C++ a využíva knižnicu Computer Vision library OpenCV verzia 3.2 (BSD licence). Tento nástroj implementuje algoritmus pracujúci nasledujúcim spôsobom [22]:

1. Pred spustením programu je potrebné pripraviť adresáre s reálnymi odtlačkami obsahujúce prejav choroby a syntetickými odtlačkami.
2. Program načíta reálne odtlačky s odpovedajúcim poškodením zo zložky. Následne lokalizuje poškodenia spôsobené zvolenou chorobou (rôzne choroby môžu spôsobiť rôzne poškodenia), tie sa po jednom z obrazu extrahujú a uložia do príslušnej zložky.
3. Pokiaľ zložka z extrahovanými prvkami nie je prázdna, program otvorí ďalšiu zložku, v ktorej by sa mali nachádzať nepoškodené syntetické odtlačky. Ak je táto podmienka splnená program, vloží prvky poškodenia na náhodne miesta v odtlačku a otočí ich náhodný uhol.



Obrázok 2.6: Obrázky znázorňujúce postup extrahovania poškodených plôch. a) Reálny poškodený odtlačok, b) detekcia poškodených častí, c) príklad extrahovanej plochy, ktorá sa uloží do adresáru. [22]

Anguli je voľne dostupná aplikácia na generovanie syntetických odtlačkov. Vychádza z metódy SFinGe a je implementovaná v jazyku C++. Má užívateľské rozhranie a je schopná napríklad generovať viac zobrazení jedného odtlačku, ukladať metadáta, ukladať odtlačky do adresárov po množstvách určených užívateľom.

Kapitola 3

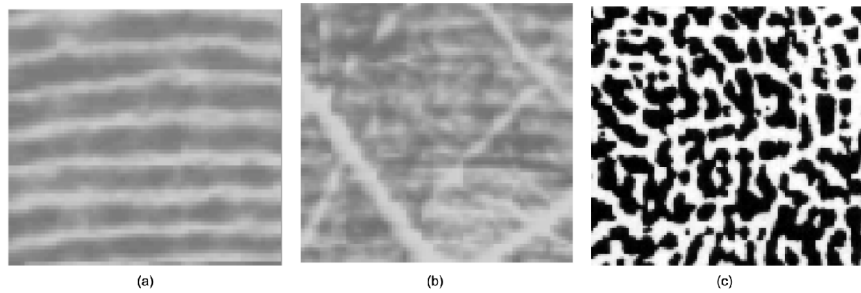
Kvalita odtlačkov prstov

Výkonnosť algoritmov na porovnávanie odtlačkov je z veľkej časti závislá na kvalite vstupného obrazu odtlačku. V ideálnom prípade sa hrebene a údolia na odtlačkoch prstov striedajú a prúdia lokálne konštantným smerom. [24] Kvalitu získaného obrazu však môžu znižovať viaceré faktory. Medzi hlavné kategórie, ktoré nepriaznivo ovplyvňujú kvalitu sú stav snímaného prstu užívateľa a okolité faktory, ktoré z veľkej časti závisia od typu skenera. Medzi najčastejšie príčiny zníženia kvality odtlačku zo strany užívateľa patria: mechanické poškodenie prstu (rezné rany, jazvy a pod.), kožné choroby, vek, ale aj prílišný tlak vyvíjaný na senzor či nečistoty.

Narozdiel od znakového hesla, algoritmy na zhodu odtlačkov nie sú stopercentné a každý negatívny faktor môže mať za následok nesprávne vyhodnotenie výsledku. Nemalé úsilie je preto vynaložené nie len na spôsoby ako detektovať nízku kvalitu obrazu, ale hlavne na postupy vedúce k zvýšeniu kvality. [30]

Vo všeobecnosti vieme kvalitu oblastí obrazu rozdeliť do troch kategórií na základe čitateľnosti štruktúr papilárnych línií, ako vidíme na obrázku 3.1 : [19]

- *Dobre ohraničená oblasť*, kde sú hrebene a údolia odlíšené od seba takým spôsobom, že algoritmus na extrakciu markantov vie správne operovať.
- *Obnoviteľné poškodené oblasti*, kde hrebene a údolia sú poškodené drobným množstvom zlomov, šmúh, šumu atď. Sú však stále čitateľné a zo susedných oblastí vieme získať dostačujúci počet informácií k ich pravdivej rekonštrukcii.
- *Neobnoviteľná poškodená oblasť*, kde hrebene a údolia sú poškodené tak veľkým množstvom hluku, že nie sú jasne definované a susedné oblasti neposkytujú dostatok informácií na ich rekonštrukciu.



Obrázok 3.1: a) Dobře ohraničená oblast b) Obnovitelná poškozená oblast c) Neobnovitelná poškozená oblast. [19]

3.1 Evaluácia kvality

Pod pojmom kvalita pri odtlačkoch prstov rozumieme hlavne ich spôsobilosť pre ďalšie spracovanie a porovnávanie. Odhad kvality môže byť globálny, kde je kvalita celého odtlačku vyjadrená jedinou hodnotou alebo lokálny kde, posudzujeme kvalitu jednotlivých segmentov získaného obrazu. Lokálny odhad kvality je pre nás oveľa výhodnejší, pretože nám poskytuje užitočnejšiu a podrobnejšiu inFormáciu o jednotlivých častiach odtlačku. Napríklad vieme rôznym segmentom priradiť rôzne váhy, čo môže pomôcť pri ďalšom spracovaní. Navyše sme schopní veľmi jednoducho s lokálnych hodnôt kvality vyrátať globálnu. Dôvody pre výpočet odhadu kvality:

- Odmietnutie ďalšieho spracovania obrazu nízkej kvality a výber najvhodnejšieho .
- Priradenie váh segmentom s nižšou či vyššou kvalitou, či ich úplné odstránenie z ďalšieho spracovania.
- Výber vhodných metód pre spracovanie odtlačku.

3.1.1 Lokálny a globálny odhad kvality

Populárnym nástrojom na odhad kvality býval NFIQ (NIST Fingerprint Image Quality). NFIQ definuje kvalitu ako predpoveď výkonu porovnávača: kvalitné odtlačky prstov pravdepodobne prinesú vysoké skóre zhody. Výstupom hodnotenia je hodnota na stupnici od 0 do 5, kde 0 znamená, že získaný obraz neobsahuje žiadnu užitočnú inFormáciu. V roku 2011 začal NIST v spolupráci s mnohými ďalšími vyvíjať NFIQ 2, ktorý sa dodnes využíva a aktualizuje. Hlavným rozdielom mimo efektívnosť algoritmu je oveľa vyššie rozlíšenie výsledného skóre na stupnici od 0 po 100, ktoré sa vypočíta na základe 69 vlastností odtlačku. Patria tam napríklad počet a kvalita markantov, variácie frekvencie papilárnych línií v obraze a podobne. [15]

Väčšina metód pre odhadovanie lokálnej kvality (po blokoch) vychádza zo spoľahlivosti miestnej orientácie papilárnych línií. Spôsoby výpočtu spoľahlivosti lokálnej orientácie sú rôzne, ale obvykle sa získava koherenciou k susedným orientáciám v danom segmente. Jedným z problémov tohto prístupu sú segmenty v blízkostiach singularít, ktoré majú obvykle vysoký uhol zakrivenia a tým pádom nízku úroveň koherencie. Na prekonanie tohto problému sa využívajú perátory lineárnej a parabolickej symetrie. Medzi ďalšie parametre využívané k odhadu lokálnej kvality patria [24]: Frekvencia, hrúbka hrebeňa a hrúbka od hrebeňa k doline, pri čom odchýlky od typického rozsahu pre tieto hodnoty môžu predpovedať nízku

kvalitu daného segmentu. Štatistiky odvodené z intenzít pixelov ako sú priemer, rozptyl, kontrast, magnitúda gradientu, vlastnosti histogramu.

3.2 Postupy zvýšenia kvality obrazu

Ako som už spomínal, vyššia kvalita vstupného obrazu predpovedá vyššiu úspešnosť operácií prevádzaných nad odtlačkom. Väčšina aktuálnych snímačov odtlačkov vyžaduje určité predspracovanie obrazu pred tým, ako prejde ku krokom potrebným k porovnávaniu. Hlavné príčiny kvôli ktorým je toto predspracovanie nutné, sú kontrast obrazu, stav prstu užívateľa a iné okolité faktory. Cieľom je získať čo najvyšší počet kvalitných segmentov, markantov a minimalizovať nežiaduce prvky v získanom obraze. V tejto sekcii sa zameriam na metódy a princípy, ktoré nám to umožnia.

3.2.1 Vylepšenie po pixeloch

Pri pixelovom vylepšení vychádzame len z predošlej hodnoty pixelu a pár globálnych parametrov. Obvykle sa tejto metódy využíva za cieľom vytvoriť dostatočný kontrast medzi hrebeňmi a údoliami. To sme schopní pomocou jednoduchého prahovania, naťahovania alebo normalizácie. Tieto techniky sami osebe nie sú dostačujúce, no sú vo veľkom využívané ako vstupný krok pri sofistikovanejších algoritmoch. Nevýhodou tohto kroku môže byť zvýšenie šumu pozadia a tým zhoršenie našej schopnosti ho rozlíšiť. Tento problém je čiastočne riešiteľný pokiaľ operácie ako je prahovanie, naťahovanie a pod. prevádzame po segmentoch s lokálnymi hodnotami. Napriek týmto úskaliam je vylepšenie kontrastu nevyhnutné pre bližšiu analýzu hrebeňov. [16][24]



Obrázok 3.2: Vpravo vidíme výsledok normalizácie. [24]

3.2.2 Kontextové filtre

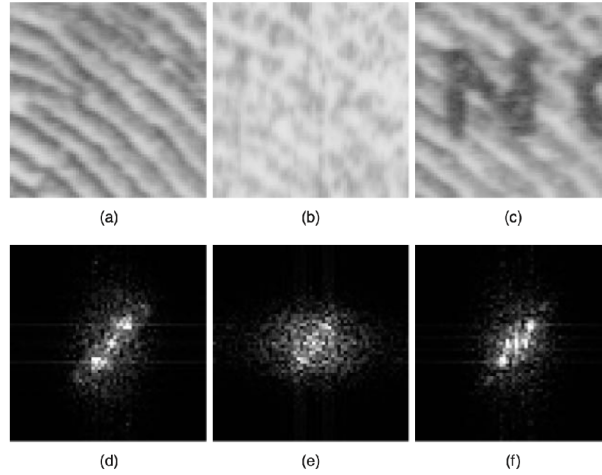
Kontextové filtrovanie nám umožňuje rozdeliť obraz na množinu segmentov, kde na každý segment vieme aplikovať iný filter z pred pripravenej skupiny. Filter sa vyberá na základe lokálneho kontextu. Pri odtlačkoch to vo väčšine prípadov predstavuje orientácia a frekvencia hrebeňov. Šírka hrebeňov je ovplyvnená faktormi ako lokácia, stav prstu či jeho veľkosť. To má za následok odlišné hodnoty frekvencií v odtlačku. Táto vlasnosť je preto dôležitá pri návrhu kontextových filtrov na zlepšenie obrazu. O niektorých metódach vychádzajúcich z tohto princípu si povieme nižšie. [16]

Metódy založené na vzťahu hrebeňov k Furierovej doméne

Pri odtlačkoch je možné využiť toho, že hrebeňový vzor (šírka a frekvencia hrebeňov) je v lokálnom merítku cyklický a sme preto schopní ho popísať Furierovou doménou. Vďaka tejto

vlastnosti vieme navrhnuť dva odlišne filtre - frekvenčný filter a smerový filter. Použitím týchto dvoch filtrov sa obraz odfiltruje a najvhodnejší výsledok sa zvolí na vylepšenie obrazu.

V 2D Furierovom spektre máme možnosť vidieť hrebeňový vzor ako dva vrcholy. Lokácia týchto dvoch vrcholov nám popisuje orientáciu a frekvenciu. Frekvencia je určená vzájomnou vzdialenosťou dvoch vrcholov a orientácia kolmicou spájajúcou tieto vrcholy. [18]

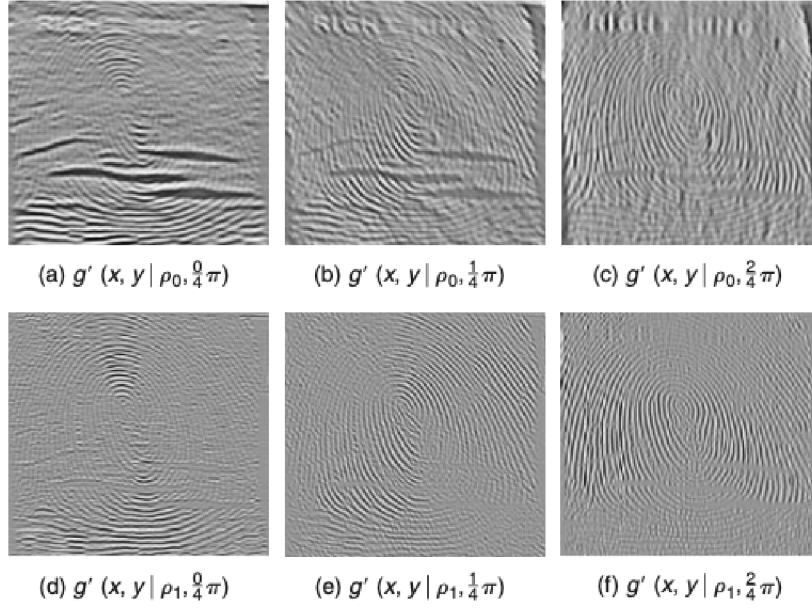


Obrázok 3.3: Na vrchných obrázkoch máme vstupný obraz a pod nimi vidíme ich reprezentáciu na Furierovom spektre. Môžeme vidieť, že pri obrázku (a) s vyššou kvalitou sme schopní lokalizovať vrcholy, zatiaľ čo (b) a (c) produkujú nerozlišiteľné, rozmazané výsledky. [18]

Navrhnutý $H(\rho, \phi)$ filter má nasledujúcu Formu:

$$\mathbf{H}(\mathbf{u}, \mathbf{v}) = \mathbf{H}(\rho, \phi) = \mathbf{H}_{\text{frekvencia}}(\rho) \cdot \mathbf{H}_{\text{smer}}(\phi) \quad (3.1)$$

kde (ρ, ϕ) označujú polárne súradnice vo Furierovej doméne a ortogonálne súradnice sú vyjadrené ako $(\mathbf{u}, \mathbf{v}) = (\rho \cos\phi, \rho \sin\phi)$. Filter H rozdelíme na dve zložky, ρ frekvenčný filter a ϕ smerový filter. Oddeliteľnosť týchto zložiek nám umožňuje nezávisle spracovanie frekvencií a orientácií pri samotnom procese vylepšenia obrazu. [18]



Obrázok 3.4: Filtrované obrázky pomocou $H(\rho, \phi)$. Kde ρ_0 je menšie ako ρ_1 . [24]

Vstupný obraz je filtrovaný každým filtrom zo setu $\{H(\rho, \phi | \rho_i, \phi_i)\}$. Pre každý filter sa vypočíta jeho lokálny výkon. Funkcia, ktorá ráta lokálne výkony môže vyzerat nasledovne:

$$\mathbf{p}(\mathbf{x}, \mathbf{y} | \rho_i, \phi_i) = \frac{1}{S} \sum_{\mathbf{b}=-\mathbf{h}_y}^{\mathbf{h}_y} \sum_{\mathbf{a}=-\mathbf{h}_x}^{\mathbf{h}_x} \mathbf{g}'(\mathbf{x} + \mathbf{a}, \mathbf{y} + \mathbf{b} | \rho_i, \phi_i)^2 \quad (3.2)$$

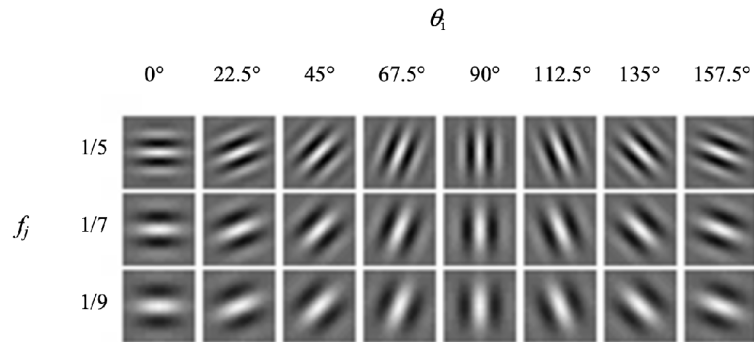
kde $p(x, y | \rho_i, \phi_i)$ reprezentuje lokálnu silu filtru, $\mathbf{g}'(\mathbf{x}, \mathbf{y} | \rho_i, \phi_i)$ je filtrovaný segment a $S = (2h_x + 1)(2h_y + 1)$. Z vyhodnotených filtrov sa na koniec takzvanou energetickou funkciou E vyberú tie najvhodnejšie. [18] Tento prístup je možné podľa potreby implementácie prispôbiť. Kvalitnejší výsledok vieme získať napríklad pomocou vytvorenia viac filtrov alebo presnejšou energetickou funkciou, tým však môžeme zvýšiť nároky na zdroje a čas.

Metóda založená na Gaborových filtroch

V roku 1998 bola navrhnutá efektívna metóda založená na Gaborových filtroch. Gaborové filtre využívajú frekvenčné aj priestorové vlastnosti, vďaka čomu sú vhodné pre aplikáciu na odtlačkoch. Párny symetrický dvojrozmerný Gaborov filter má nasledujúci tvar:

$$\mathbf{g}(x, y : \theta, f) = \exp \left\{ -\frac{1}{2} \left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2} \right] \right\} \cdot \cos(2\pi \cdot x_\theta) \quad (3.3)$$

kde θ predstavuje orientáciu filtra, $[x_\theta, y_\theta]$ sú súradnice $[x, y]$ po otočení kartézskych osí o uhol $90^\circ - \theta$. f je frekvencia sínusovej vlny a $\sigma_{x,y}$ je štandardná odchýlka. [24] Frekvencia filtru je určená čisto z lokálnej frekvencie línií a orientácia filtru je taktiež určená čisto z lokálnej orientácie. Voľba odchýliek $\sigma_{x,y}$ nie je však tak zrejímavá. Čím vyššie hodnoty majú, tým sú filtre odolnejšie voči šumu v obraze, môžu však vytvoriť falošné hrebene a údolia. Nízke hodnoty naopak nevytvárajú nežiadúce prvky, ale nie sú príliš efektívne v odstraňovaní šumu. Výber vhodných odchýliek je preto často kompromis.



Obrázok 3.5: Príklad 24 Gaborových filtrov s odchýlkou = 4. [24]

3.2.3 Rozklad do viacerých rozlíšení

Dekompozícia obrazu do rôznych frekvenčných pásiem alebo pod-obrazov nám umožní pozorovať rôzne šумы v rôznych mierkach. Drsný tok hrebeňa sme schopní odstrániť v nízkych až stredných frekvenciách, naopak vyššie nám zachovávajú jemnejšie detaily. Jednotlivé pásma sa nakoniec znovu skombinujú do výsledného obrazu. Z tohto princípu vychádzajú mnohé metódy napríklad [24]:

- Využitie variácie Laplaceovskej pyramídy na dekompozíciu originálneho obrazu na tri menšie, kde sa na každý z nich aplikuje určitá forma kontextového filtru.
- Reprezentovanie obrazu vo viacerých rozloženiach založených na vlnovom rozklade. Každý pod-obraz je filtrovaný textúrovým aj smerovým filtrom na to, aby sa potlačil spektrálny šum a uzavreli sa medzery spôsobené jazvami či záhybmi.

3.2.4 Detekcia a odstránenie záhybov

Prítomnosť záhybov v odtlačku môže negatívne ovplyvniť jeho spracovanie napríklad tým, že spôsobia falošné markanty ako je ukončenie. Viaceré z vyššie spomínaných postupov sú schopné drobné medzery medzi hrebeňmi z obrazu odstrániť. Pokiaľ sa jedná o väčšie poškodenie alebo je obraz nekvalitný, nemusia tieto postupy stačiť. Preto existujú techniky zamerané konkrétne na túto problematiku, ako sú [16] [24]:

- Detekčný prístup založený na analýze Houghovo transformačného priestoru odvodeného od zakončenia hrebeňov. Táto technika je pomerne jednoduchá a efektívna pri detekcii v segmentoch s vyšším šumom. [16]
- Detekcia bodov záhybov, porovnávaním medzi lokálnymi orientáciami rátaných v dvoch rôznych mierkach. Dôvodom je, že lokálna orientácia pri záhybe rátaná v malom merítku sa od hlavnej orientácie, ktorú vieme odhadnúť výrazne odchyľuje. [16]

3.2.5 Zhrnutie metód zvyšujúcich kvalitu

Metódy na zvýšenie kvality vstupného obrazu môžu byť aplikované v odlišných fázach spracovania odtlačku a ich výstup sa môže taktiež líšiť v závislosti od metódy napríklad obraz v šedých odtieňoch, či binárny obraz. Pri výbere vhodnej metódy na zlepšenie kvality (funkčnosti) obrazu odtlačku musíme myslieť na to, že vizuálne uspokojivý výsledok nemusí

znamenat vyššiu úspešnosť ďalších operácií. V určitých prípadoch môže zlý odhad lokálnej kvality, zle zvolená metóda alebo parametre filtru viesť práve k nežiaducej strate informácií alebo k všeobecnému zníženiu kvality obrazu. Jedným z mnohých faktorov, ktoré ovplyvňujú výber finálnej metódy či metód, sú výpočetná a časová náročnosť.

3.3 Lokalizácia poškodenia odtlačku

Pre úspešnú rekonštrukciu poškodeného odtlačku je dôležité správne lokalizovať a vymedziť poškodené miesta. Na lokalizáciu poškodení budem v tejto práci využívať aplikáciu z bakalárskej práce Štěpánky Barotovej. Jedná sa o aplikáciu s užívateľským rozhraním, ktorá je schopná lokalizovať poškodené oblasti v odtlačku a na základe typu poškodenia klasifikovať akou chorobou je odtlačok postihnutý. Tento proces je navyše vďaka užívateľskému prostrediu prehľadne vizualizovaný. Aplikácia sa skladá z dvoch častí: detektora a klasifikátora. Pre túto prácu budem používať hlavne funkciu detektora a jeho štruktúru si následne priblížime. [4]

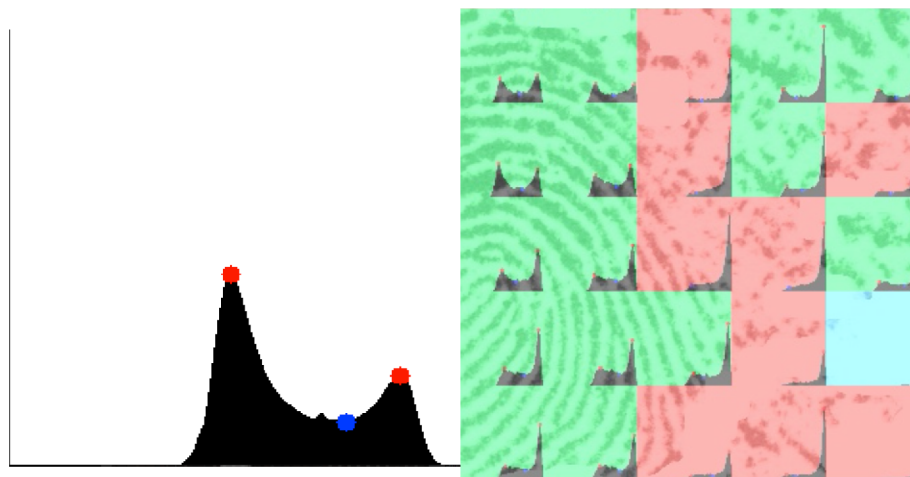
Výstup detektora je list extrahovaných poškodení a skladá sa zo šiestich pod-detektorov, jeden pre každý typ poškodenia.

- Biele škvrny
- Záhyby, ryhy
- „Gepardie“ škvrny
- Porušenie papilárnych línií
- Diskontinuita pola orientácií
- Nebimodálny histogram

Každý z nich obraz pred-spracuje a extrahuje žiadané časti. Pri extrahovaní sa využívajú tri metódy: Analýza Histogramu, Blokové Pole Orientácií a takzvaná „Flood Fill“ metóda, tá sa využíva pri prvých štyroch bodoch. [4]

3.3.1 Analýza histogramu

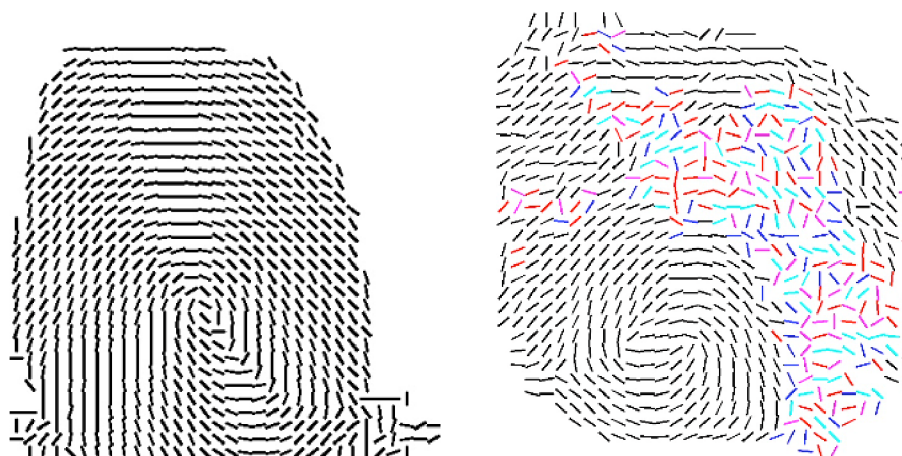
Táto metóda vychádza z predpokladu, že histogram pre nepoškodený a dobre zosnímaný obraz bude mať dva vrcholy približne rovnakej výšky spojené údolím. Takýto tvar nazývame bimodálny. Je to spôsobené tým, že v kvalitnom obraze sú hrebene a údolia rovnomerne zastúpené a majú približne rovnakú farbu. Implementovaný algoritmus získa histogram pre každý segment a pokiaľ nemá bimodálny tvar alebo nespĺňa určité parametre (napr. výška a vzdialenosť vrcholov), vieme s vysokou pravdepodobnosťou o danom sektore prehlásiť, že v ňom ide o poškodenie obrazu/odtlačku. [4]



Obrázok 3.6: Na obrázku vľavo je ideálny bimodálny histogram a v pravo sú histogramami segmentov v spracovávanom obraze. [4]

3.3.2 Blokové pole orientácií

Pole orientácií som už spomínal v sekcii 2.4 a je súčasťou všeobecného postupu pri spracovaní odtlačku. Výhodou tejto metódy je to, že sa ľahko napája na už existujúce riešenie. Ako náhle máme získané pole orientácií, detektor začne po riadkoch a po stĺpcoch porovnávať stupne orientácií v susedných blokoch. Ak je tento rozdiel väčší ako 45 stupňov, detektor blok označí ako poškodenie. Týmto spôsob, ale môžu byť ako poškodenie vyhodnotené náhodné jednorázové odchýlky alebo môžu byť ignorované miesta vo vnútri poškodenej oblasti. Tento problém je v tejto práci riešený kopírovaním vlastností bloku do susedných. [4]



Obrázok 3.7: Na obrázku vľavo je pole orientácií zdravého odtlačku a vpravo pole orientácií s už detekovaným poškodením. [4]

3.3.3 Flood Fill metóda

Jedná sa o známy algoritmus využívaný primárne pri grafike. Algoritmus má tri parametre štartovací pixel, cieľovú farbu a farbu, ktorá sa ma nahradiť. Pred spustením algoritmu je

však obraz potrebné pedspracovať a previesť na čierno biely. Náročnosť tohto kroku veľmi závisí na kvalite a poškodení odtlačku. Ak máme takto spracovaný obraz, algoritmus sa pozrie na okolité 4 pixely a nahradí farbu, ktorá sa má nahradiť cieľovou. Následne pomocou rekurzie alebo dátovej štruktúry (zásobník, fronta atď.) sa susedný pixel stane štartujúcim pixelom. V danej bakalárskej práci bola použitá variácia "Scanline Flood Fill" k zlepšeniu správy pamäti a zachovaní informácií o bodoch patriacim určitej ploche. Tento detektor detekuje veľké biele škvrny, hrubé biele ryhy, "gepardie"škvrny a porušenie papilárnych líní. [4]



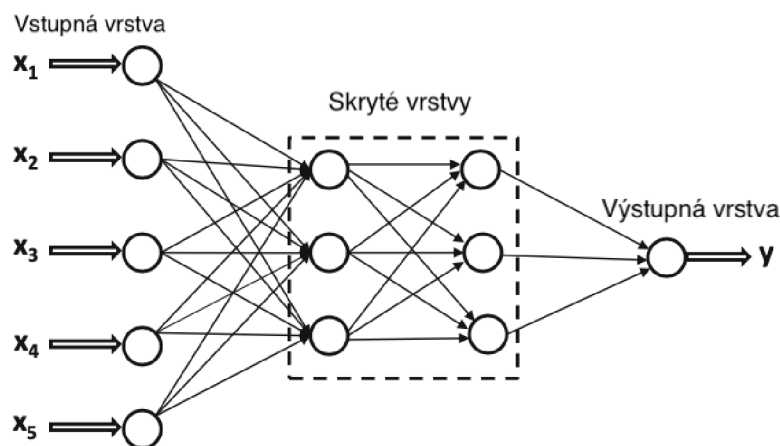
Obrázok 3.8: Extrahovanie rovných bielych rýh. [4]

Kapitola 4

Neurónové siete

Neurónové siete sú typom umelej inteligencie, ktorá sa snaží replikovať spôsob, akým funguje ľudský mozog. Skladajú sa zo vzájomne prepojených uzlov známych ako umelé neuróny, ktoré dokážu spracovávať a prenášať informácie. Umelé neuróny (perceptrony) sú vzájomne prepojené, pričom každé spojenie (vstup/výstup) neurónu má svoju vlastnú váhu. Na obrázku 4.1 môžeme vidieť vzorové schéma takto prepojenej siete. Umelá neurónová sieť počíta funkciu vstupov šírením vypočítaných hodnôt zo vstupných vrstiev siete do výstupných a váhy spojení využíva, ako parametre tohto šírenia. Pomocou úpravy týchto váh dokážeme v neurónovej sieti simulovať proces učenia. K tomu sú potrebné tréningové dáta obsahujúce vstupy a k nim očakávané výstupy. [1]

Okrem neurónov a váh obsahujú neurónové siete aj aktivačné funkcie, čo sú matematické funkcie, ktoré určujú, ako sa výstup neurónu transformuje predtým, ako sa preniesie do ďalšej vrstvy. Medzi aktivačné funkcie patria: sigmoidná funkcia, hyperbolický tangens, saturovaná lineárna funkcia, ostrá nelinearita (ReLU) a ďalšie. Na základe využitia architektúry sme schopní definovať viacero druhov neurónových sietí napríklad: dopredné neurónové siete, konvolučné neurónové siete, rekurentné neurónové siete, autokódery a generatívne protichodné siete. Niektorým z nich sa budem podrobnejšie venovať v sekcii nižšie a na konci sekcie 4.1 môžete pseudokód pre proces tréningovania neurónovej siete vidieť. [1][29]

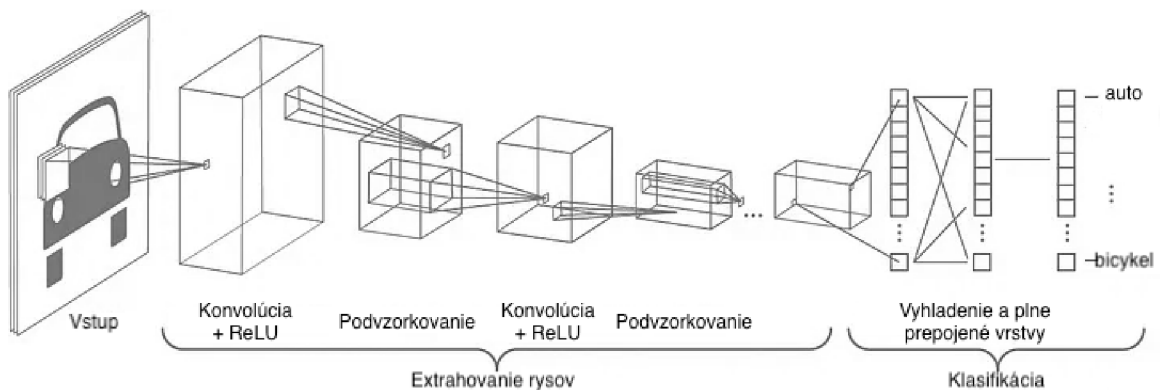


Obrázok 4.1: Príklad jednoduchšej neurónovej siete s dvoma skrytými vrstvami. [1]

4.1 Konvolučné neurónové siete

Konvolučné neurónové siete (CNN) sú špeciálnym typom hlbokoj neurónovej siete, ktorý je vhodný na spracovanie obrazu či videa. Tieto siete sa úspešne využívajú v širokej škále aplikácií ako napríklad medicínske diagnózy alebo detekcia a klasifikácia objektov v obraze. Vstupom obvykle býva trojdimenzionálne pole (výška, šírka a znaky), kde znaky môžu reprezentovať napríklad farbu v danom bode ako RGB. Základným princípom tejto siete je konvolúcia, ktorej sa budem bližšie venovať v sekcii nižšie. V jednoduchosti sa jedná o matematickú operáciu, ktorá sa využíva na vstupnej vrstve konvolučnej siete - aplikuje filtre na vytvorenie takzvaných rysových máp. Tieto mapy sa ďalej predávajú ako vstup do ďalších vrstiev, ktoré aplikujú ďalšie operácie, čím vzniká abstraktnejšia reprezentácia vstupu, z ktorej sme schopní vyvodit užitočné informácie.[1][12][4]

Architektúra konvolučných neurónových sieti sa skladá z kombinácie niekoľkých konvolučných, aktivačných a podvzorkovacích vrstiev (pooling layers), nasledovaných jednou alebo viacerými plne prepojenými vrstvami. Striedaním konvolúcie, aktivačnej funkcie a podvzorkovania vznikajú už spomínané rysové mapy. Vstupné a výstupné dimenzie sú špecifické vzhľadom k aplikácii. Medzi populárne architektúry patria LeNet, AlexNet, VGGNet, ResNet a InceptionNet. Líšia sa hlavne počtom vrstiev (hĺbkou) a parametrov. [2][12]



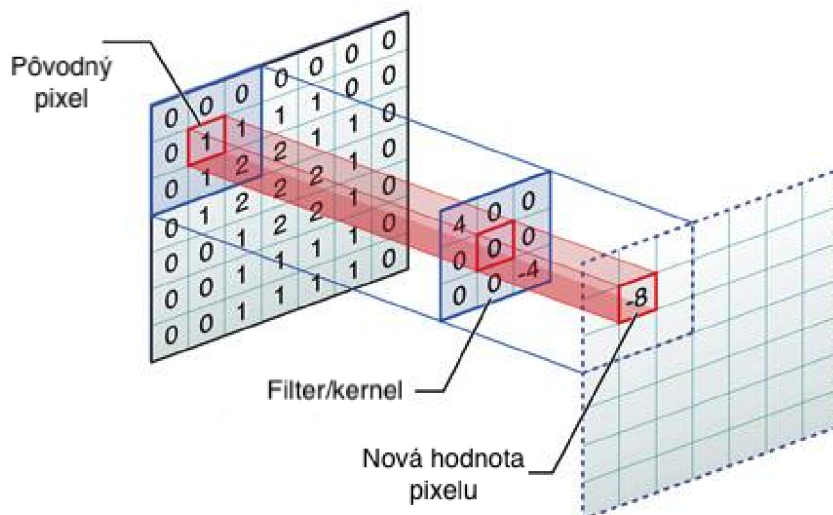
Obrázok 4.2: Znáznornenie siete s viacerými konvolučnými vrstvami, slúžiace na klasifikáciu dopravného prostriedku. [25]

4.1.1 Konvolúcia

Konvolúcia je často využívaná v spracovaní signálu, počítačovom videní a v strojovom učení. Spočíva v aplikovaní filtra (masky) na signál alebo obraz, za účelom transformácie alebo extrahovania. Výsledkom obvykle býva nový signál alebo obraz. Matematicky môžeme konvolúciu formulovať ako lineárnu operáciu na dvoch funkciách (f a g), ktorá produkuje tretiu funkciu ($f * g$). [5]

Pri spracovaní obrazu je konvolúcia proces transformácie obrazu aplikovaním kernelu (filtru) na každý pixel a jeho susedov naprieč celým obrazom. Kernelom/filtrom sa v tomto kontexte chápe matica reálnych hodnôt, ktorá sa postupne prikladá na obraz, kde sa každá pôvodná hodnota vynásobí prekrývajúcou a výsledky sa sčítajú a vznikne nová hodnota stredného pixelu. Tento postup sa opakuje pre každý pixel v obraze. Vzorec tejto operácie je nasledovný: [12] [2] [1]

$$y[m, n] = x[m, n] * h[m, n] = \sum_j \sum_i x[i, j] \cdot g[m - i, n - j] \quad (4.1)$$



Obrázok 4.3: Príklad jednoduchej konvolúcie s filtrom o veľkosti 3x3 pixely. [5]

Teraz, keď sme si popísali základné princípy konvolúcie, môžeme prejsť na hlavné dôvody a výhody jej použitia v neuronových sieťach. Narozdiel od predošlých prístupov využívajúcich prevažne násobenie matice maticou parametrov, má využitie konvolúcie tieto vlastnosti: riedke interakcie, zdieľanie parametrov a ekvivalentné reprezentácie.

Riedka interakcia označuje vlastnosť konvolučných sietí, pri ktorých každý neurón na výstupe konvolučnej vrstvy je spojený len s malou podmnožinou vstupných dát. Problémom násobenia matíc bolo predovšetkým to, že každý vstupný neurón bol prepojený s každým výstupným neuróm, čo malo za následok veľký počet parametrov a vyššiu výpočetnú náročnosť. Narozdiel od toho konvolučná vrstva posúva kernel menšej veľkosti ako je vstup po jeho ploche, vďaka čomu je schopná odhaliť lokálne rysy ako sú rohy, hrany a iné textúry. [12]

Vďaka menšiemu filtru sa dostávame k druhej dôležitej vlastnosti konvolúcie, a to je zdieľanie parametrov. Znamená, že sme schopní aplikovať rovnaké parametre (váhy) na každú časť vstupného obrazu a sieť sa tak nemú učiť nové parametre pre každý pixel. Myšlienka za tým je, že pokiaľ má niečo zmysel detektovať v jednej časti obrazu pravdepodobne sa oplatí to detektovať všade. Zdieľaním parametrov sme schopní znížiť výpočetnú náročnosť, paralelizovať konvolúciu, čím sa stane model kompaktnejším a rýchlejšie trénovateľným. Navyše veľa moderných (TensorFlow, PyTorch) nástrojov je optimalizovaných na využitie zdieľania parametrov. [2][1]

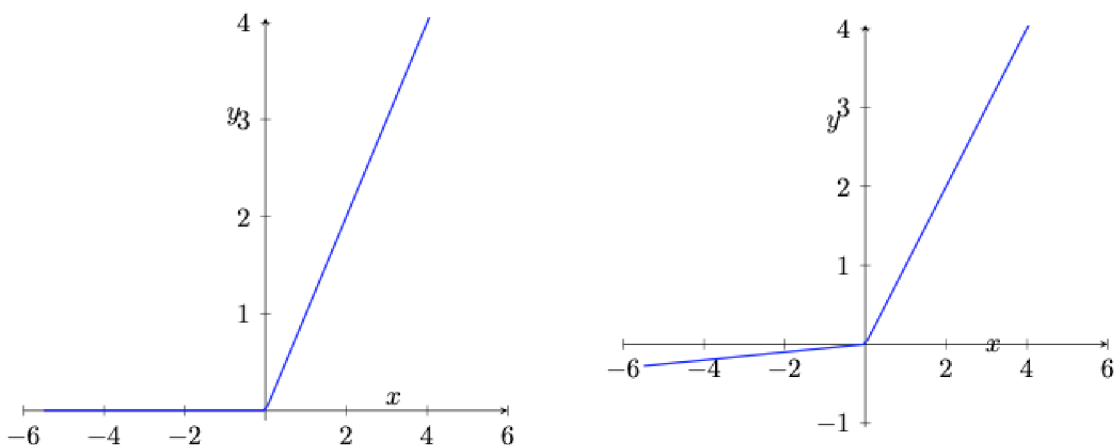
Poslednou z hlavných výhod konvolúcie je ekvivalentná reprezentácia. Znamená to, že ak sa zmení vstup, tak výstup sa zmení rovnakým spôsobom. Matematicky povedané funkcia $f(x)$ je ekvivalentná k funkcii $g(x)$ ak $f(g(x)) = g(f(x))$. Táto vlastnosť je pre nás veľmi dôležitá pretože prvky, ktoré v obraze detekujeme môžu byť posunuté, otočené alebo rozdielne veľké a vďaka nej sú konvolučné siete schopné sa týmito problémami vysporiadať, čím sa stávajú robustnejšie. [12][2]

4.1.2 Aktivačné funkcie

Veľkou súčasťou neurónových sietí je takzvaná aktivačná vrstva aplikujúca aktivačnú funkciu na výstup predchádzajúcej vrstvy. Jej účelom je vniesť nelinearitu do výstupu neurónovej vrstvy a umožniť tak sieti vytvárať nelineárne spojenia medzi vstupom a výstupom, vďaka čomu je schopná odhaliť a reprezentovať oveľa komplexnejšie vzorce v dátach. Voľba vhodnej aktivačnej funkcie má preto výrazný vplyv na efektívnosť a výstup siete. Medzi základne aktivačné funkcie patria: identita, sigmoid, hyperbolický tangens, signum, softmax, ReLU a ďalšie.[2] [1]

ReLU

ReLU (Rectified Linear Unit - usmernená lineárna jednotka) patrí medzi najvyužívanejšie aktivačné funkcie využívané v konvolučných neurónových sieťach. Jedná sa o jednoduchú funkciu, ktorá vracia nulu pokiaľ pre negatívny vstup a pozitívnu hodnotu ponechá nezmenenú. Matematicky ju môžeme zapísať nasledovne: $ReLU(x) = \max(0, x)$.



Obrázok 4.4: Priebek základnej funkcie $ReLU(x)$ a priebek presakujúceho $ReLU$ $LeakyReLU(x)$. [3]

Hlavnou výhodou pre použitia ReLU je jej výpočetná nenáročnosť, keďže sa jedná len o jednoduché prahovanie. Ďalšou výhodou ReLU funkcie je, že je schopná potlačiť problém miznúceho gradientu, čo je pomerne častý problém pri hlbokých neurónových sieťach. Gradient, ktorý sa využíva na aktualizáciu váh sa svojou propagáciou cez mnohé vrstvy neustále znižuje, čím sa spomaľuje proces učenia v sieti. ReLU tento problém rieši vďaka tomu, že jeho pre pozitívne vstupné hodnoty majú vždy konštantný gradient rovný 1. Vďaka tomu sme schopní v priebehu propagácie udržať väčšie gradienty a zefektívniť proces učenia. [2][3]

Problém pri používaní klasického ReLU je, že môže vytvárať takzvané mŕtve neuróny. Táto situácia nastáva pokiaľ je veľká časť vstupu negatívna. V takom prípade bude výstupná hodnota neurónu 0 a ďalej už nijak neprispieje do ďalšieho spracovania vstupu a výsledku siete. Riešením tohto problému sú rôzne variácie ReLU ako napríklad presakujúce $ReLU$ (leaky $ReLU$), náhodné presakujúce $ReLU$, parametrické $ReLU$ a exponenciálne lineárne jednotky (ELU).[2] [1]

Presakujúca ReLU sa od štandardnej líši tým že aj pre negatívni vstup vracia malé pozitívne hodnoty. Dosahuje sa tak násobením negatívnej hodnoty vstupu parametrom α , ktorý je v rozsahu (0,1). Obvykle sa volí hodnota $\alpha = 0,01$, v niektorých prípadoch môžu vyššie hodnoty produkovať lepšie výsledky. Ďalšou možnosťou je správať sa k α ako k premennej, ktorá sa môže odvodiť z tréningových dát. Takúto aktivačnú funkciu nazývame parametrická ReLU. [1] [3]

4.1.3 Loss funkcia a optimalizácia

Loss funkcia alebo inak známa ako cieľová alebo nákladová funkcia (cost function) sú kľúčovou súčasťou neurónových sietí. Ich úlohou je zistiť rozdiel medzi očakávaným výstupom a predikciou siete. Optimalizačné algoritmy ako stochastický gradientový zostup, adaptívna metóda momentového odhadu (Adam), či koreňová kvadratická propagácia strednej hodnoty (RMSProp) sa snažia hodnotu loss funkcie minimalizovať úpravou parametrov modelu, a tým zlepšiť jeho celkový výkon. Výber loss funkcie sa môže odvíjať od riešeného problému alebo žiadaného správania siete. [1][2] Medzi najpoužívanejšie v rámci CNN patria:

- **Křížová-entropia:** Štandardná loss funkcia pre úlohy klasifikácie viacerých tried, ktorá ráta rozdiel medzi predpovedaným a skutočným pravdepodobnostným rozdeľením.
- **Binárna křížová-entropia:** Variant křížovej-entropickej straty pre úlohy binárnej klasifikácie, ktorý vypočíta logaritmickú stratu pre každú triedu samostatne a potom výsledky kombinuje.
- **Stredná kvadratická chyba:** Používa sa v regresných úlohách, meria priemerný kvadratický rozdiel medzi predpovedanými a skutočnými hodnotami.

$$a) L = -\frac{1}{n} \sum_{i=1}^n y_i \log \hat{y}_i \quad (4.2)$$

$$b) L = -\frac{1}{n} \sum_{i=1}^n (y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)) \quad (4.3)$$

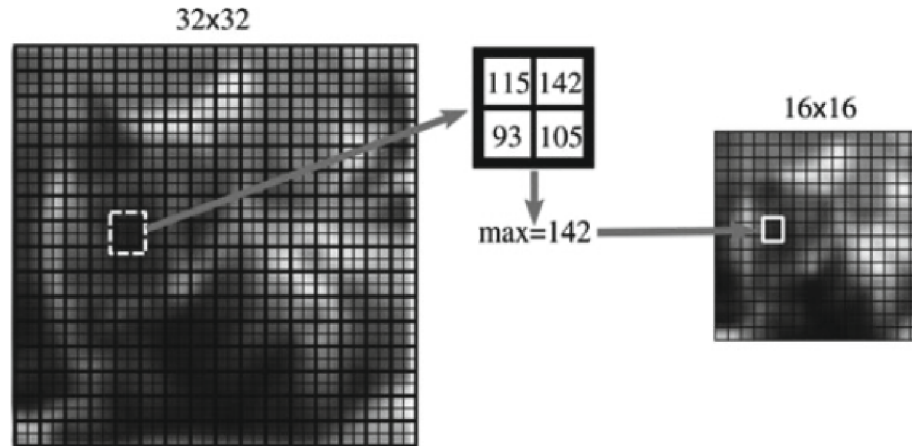
$$c) L = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (4.4)$$

Vyššie vidíme vzorce pre: a) Křížovú-entropiu b) Binárnu křížovú-entropiu c) Strednú kvadratickú chybu, kde n je počet vzorkov v datasete, y je reálna hodnota a \hat{y} je sieťou predikovaná hodnota.

4.1.4 Podvzorkovanie a veľkosť posunu

Teraz, keď sme si priblížili konvolučné a aktivačné vrstvy, môžeme prejsť na poslednú časť špecifickú pre konvolučné siete, a to sú podvzorkovacie vrstvy (v angličtine "pooling layers"). Konvulčnou vrstvou vytvorené rysové mapy zachytávajú presnú polohu detekovaných rysov a aj mierny posun. Rotácia a zmena vstupného obrazu môže viesť k odlišnému výsledku. Riešením je prídanie podvzorkovacej vrstvy, ktorá rozdelí vstup do niekoľkých $x * x$ regiónov, na ktoré sa aplikuje pooling funkcia, napríklad: max pooling alebo priemerný

pooling. Výsledok je nie len robustnejší, ale navyiac má menšie priestorové rozmery pri zachovaní dôležitých informácií, a tým robí sieť výpočetne menej náročnú. Rozmery výstupu pooling-u ovplyvňuje veľkosť zmenšovaného regiónu $R_x * R_x$ a veľkosť posunu P_x obvykle sa volí posun $P_x > 1$, takže výstup bude mať rozmery $(W_x - R_x) / P_x + 1$, kde W_x reprezentuje šírku alebo výšku. Treba dodať, že pooling prebieha separátne pre každú rysovú mapu. [1][3]



Obrázok 4.5: Príklad max pooling-u o rozmeroch $2 * 2$ s posunom 2. [2]

Proces tréningovania neurónovej siete

- 1: Definuj architektúru (počet a typy vrstiev, veľkosť filtrov atď.)
 - 2: Inicializuj hyperparametre (rýchlosť učenia, veľkosť várky, počet epoch)
 - 3: Inicializuj váhy and bias pre každú vrstvu
 - 4: Načítaj a rozdel dáta na tréningové a validačné
 - 5: Rozdeľ dáta do dávok
 - 6: **for** každú epochu **do**
 - 7: **for** každú dávku v tréningovej sade **do**
 - 8: Posuň vstup každou vrstvou siete
 - 9: Aplikuj aktivačnú funkciu, ak je potreba, aplikuj podvzorkovanie(pooling)
 - 10: Vypočítaj výstupné pravdepodobnosti pomocou softmax funkcie (pre klasifikačné úlohy)
 - 11: Vypočítaj stratu pomocou vhodnej loss funkcie
 - 12: Vypočítaj gradienty so vzťahom k hodnote loss pre každú váhu a bias
 - 13: Použi spätnú propagáciu na propagovanie gradientov naprieč sieťou
 - 14: Aplikuj zvolený optimalizačný algoritmus na aktualizovanie váh a biasov
 - 15: **end for**
 - 16: Opakuj kroky 7-11 pre validačnú sadu
 - 17: Vypočítaj presnosť modelu
 - 18: Ulož model s najlepším validačným výkonom
 - 19: **end for**
-

4.2 Generatívne protichodné siete

Generatívne protichodné siete po anglicky GANs (Generative Adversarial Networks) boli prvý krát predstavené Ianom Goodfellowom a jeho kolegami v roku 2014. Jedná sa o špecifický typ architektúry neurónových sietí, ktorý nám prevratným spôsobom umožnil generovať realistické texty, obrazy, audio či video. Vo všeobecnosti sa skladajú z dvoch častí: generujúca sieť, produkujúca syntetické dáta a diskriminačná sieť, ktorá sa snaží rozlíšiť medzi syntetickými a reálnymi dátami. Tieto dve siete sa simultánne učia v procese, ktorý sa nazýva protichodný tréning. Počas tohto procesu sa generátor snaží vytvoriť dáta, ktoré dokážu oklamať diskriminátor a ten sa naopak zlepšuje v schopnosti rozoznať pravé dáta od falošných. Po dostatočnom tréningu získame generátor, ktorý generuje dáta nerozoznateľné od reálnych. Vďaka tomu majú takéto siete širokú škálu aplikácií od počítačového videnia, zábavy až po umenie. [14][7]

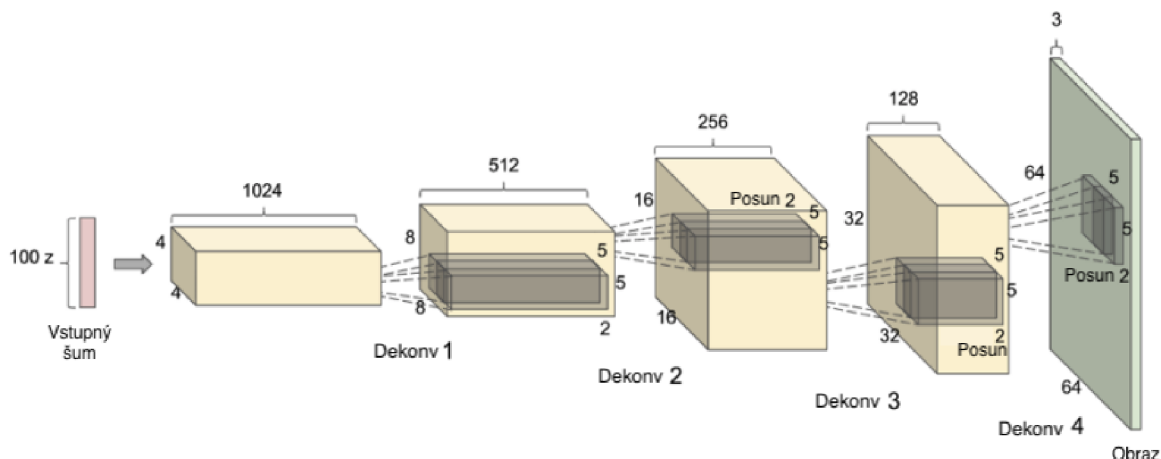
V posledných rokoch sa v rámci GANs objavil nový typ siete s názvom "podmienený GAN" (conditional GAN alebo skrátene cGAN), ktorý prináša ďalší rozmer do generatívnych procesov. Narozdiel od pôvodných GANs, cGANs umožňujú riadenie generovaných výstupov prostredníctvom podmienok, čo nám umožňuje vytvárať špecifické a personalizované dáta na základe zadaných parametrov. [27] Práve tento typ siete využívam vo svojej práci.

4.2.1 Generátor a diskriminátor

Generátor v GAN generuje syntetické dáta vďaka učeniu sa zložitých vzorov a vlastností daného typu dát. Obvykle sa jedná o hlbokú konvulučnú neurónovú sieť s cieľom oklamať diskriminátor generovaním realistických dát. V mojom prípade by sa mal rastúci výkon generátoru prejavovať presnejšou rekonštrukciou poškodených oblastí, a tým zvýšiť presnosť a efektívnosť porovnávacích systémov.

Architektúra generátoru sa skladá z: vstup, konvulučné, dekonvulučné, normalizačné a aktivačné vrstvy. Vstupná vrstva prijíma náhodný šum vo forme vektoru. Tento vektor slúži ako počiatkový bod, ktorý sa následne transformuje na syntetický obraz. Obvykle sa jedná o náhodný šum, ale v závislosti od aplikácie môže byť použitý už existujúci obraz, ktorý chceme modifikovať alebo pridaná podmienka. Proces transformácie zahŕňa sériu operácií zväčšenia a konvolúcie, ktoré drasticky menia rozmery a obsah dát. [12]

Dekonvulučná vrstva (vrstva transponovanej konvolúcie) slúži na nadzorkovanie rysových máp, čím efektívne zväčšuje rozmery generovaného obrazu pri zachovaní naučených rysov, vďaka čomu sme schopní rekonštruovať zložité štruktúry, kritické pre realističnosť výsledku. Normalizačná vrstva sa stará o stabilitu a efektívnosť tréningového procesu tým, že riadi aktiváciu v sieti. Medzi najpoužívanejšie normalizácie patrí dávková a inštančná. Tieto techniky pomáhajú riešiť problémy ako sú miznúce, explodujúce gradienty a vnútorná kovariantná zmena. Aktivačná vrstva spomínaná vyššie v podsekcii 4.1.2 má za úlohu vniesť do celého procesu nelinearitu. Vo vnútorných vrstvách sa často používa klasické a tečúce ReLU, narozdiel od toho na výstupe sa obvykle objavuje hyperbolický tangens alebo sigmoid, aby sa zaistil žiadúci rozsah výsledných hodnôt. Cieľ generátora môžeme slovne popísať ako rozpoznanie a následná replikácia komplexných rysov cieľových dát. [27]



Obrázok 4.6: Náčrt architektúry generátora používaného v hlbokých konvolučných proti-
chodných sieťach DCGAN. [13]

Diskriminátor je jeden z dvoch hlavných komponentov GAN, vo väčšine prípadov ide o klasický binárny klasifikátor, ktorý má za úlohu určiť autenticitu dát. V kontexte spracovania obrazu sa obvykle jedná o CNN (konvolučná neurónová sieť) architektúru. Vstupom sú buď reálne dáta alebo dáta produkované generátorom. Ako som už spomínal v sekcii 4.1, sieť začína konvolučnou vrstvou, ktorá sa snaží identifikovať jednotlivé rysy. Za ňou nasleduje postupnosť ďalších vrstiev zachytávajúce stále komplexnejšie vzorce a je ukončená postupnosťou plne prepojených vrstiev s jednou výstupnou hodnotou. Táto hodnota určuje pravdepodobnosť autenticity dát a slúži ako spätná väzba pre generátor aj diskriminátor. Cieľom diskriminátora a celej architektúry GAN je tréning generátora. Pri poklese úspešnosti diskriminátora klasifikovať dáta ako falošné či pravé na 50% ,sa generátor považuje za plne funkčný a diskriminátor je už nepotrebný (môže sa využiť pri ďalšom tréningu a ušetriť tak potrebné náklady). [3][14] Nižšie môžete vidieť proces tréningovania GAN siete popísaný pseudokódom.

GAN trénovací cyklus

- 1: Inicializuj generátor (G) a diskriminátor (D) neurónových sietí.
 - 2: Definuj hyperparametre: rýchlosť učenia, veľkosť dávky, počet epoch atď.
 - 3: Definuj loss funkciu a optimalizačné algoritmy pre G a D .
 - 4: **for** každá epocha **do**
 - 5: Zamiešaj vzorky skutočných dát.
 - 6: **for** každá dávka v skutočných dátach **do**
 - 7: Vzorkuj dávku náhodných vektorov šumu z , z distribúcie šumu.
 - 8: Generujte falošné vzorky pomocou $G(z)$.
 - 9: Zlúč skutočné a falošné vzorky a vytvor zmiešanú dávku.
 - 10: Prirad' vzorkám štítky: 1 pre skutočné a 0 pre falošné.
 - 11: Trénuj diskriminátor D , na zmiešanej dávke a aktualizuj jeho váhy na základe loss funkcie.
 - 12: Vzorkuj novú dávku náhodných vektorov šumu z , z distribúcie šumu.
 - 13: Aktualizuj váhy generátora minimalizáciou loss funkcie, pomocou gradientov z výstupu diskriminátora na falošných vzorkách.
 - 14: **end for**
 - 15: **end for**
 - 16: Vyhodnoť výkon GAN pomocou vhodných metrík a vizualizuj generované vzorky.
 - 17: Opakuj kroky 4-14, kým nedosiahneš požadovaný počet epoch alebo konvergenciu.
-

Kapitola 5

Návrh a implementácia

Táto kapitola pojednáva o návrhu a implementácii systému na rekonštrukciu poškodených odtlačkov. Vzhľadom na požiadavky na transformáciu obrazu som sa rozhodol použiť neurónovú sieť typu kontextová GAN (contextual GAN). Konkrétne som sa zameral na generátor typu kóder-dekóder. Pôvodne som plánoval vyrezať oblasť okolo poškodenia, poslať ju do generátoru (neurónovej siete) a výstup následne vložiť späť do pôvodného obrazu. Avšak, v prípade rozsiahlych poškodení, ako na obrázku 5.1, sa oblasti poškodenia prekrývajú a ich spätné vloženie do obrazu je problematické. Okrem toho pri posielaní obrazov do siete, je vhodné, aby mali rovnaké rozmery. To je pomerne komplikované, ak máme poškodenia s rôznymi rozmermi.

Preto som sa rozhodol použiť bitovú mapu poškodenia a do neurónovej siete poslať celý odtlačok prsta. Následne som na základe bitmapy vyrezal plochy z výstupu siete, ktoré boli vložené do pôvodného odtlačku prsta. Tento prístup umožňuje vyriešiť problémy s rôznymi rozmermi poškodenia pri zachovaní pôvodných zdravých papilárnych línií. Navyše je tento spôsob výpočtetne menej náročný, keďže na rekonštrukciu celého odtlačku stačí jeden priechod neurónovou sieťou a vyhneme sa konfliktom pri spätnom spájaní obrazu.

V nasledujúcich sekciách popisujem implementáciu contextual GAN. Celý systém sa skladá z dvoch neurónových sietí: generátora založeného na U-Net architektúre a diskriminátora odvodeného z modelu PatchGAN, ktorý sa používa na vyhodnotenie kvality generovaných odtlačkov. Implementácia bola vykonaná v jazyku Python s využitím knižnice pre neurónové siete PyTorch a ďalších nástrojov na generovanie dátovej sady. Optimalizácia a výpočet straty loss bol pre oba modely vyhodnocovaný na pôvodnom obraze, do ktorého boli vložené zmeny.



Obrázok 5.1: Na obrázku je ukážka rozsiahle poškodeného odtlačku a v ňom zvýraznených poškodených plôch

5.1 Generátor a Diskriminátor

Architektúra triedy *Generator* je odvodená od U-Net modelu, populárnej kóder-dekóder siete pri úlohách dopĺňaniach obrazov. U-Net sa skladá z dvoch hlavných častí: proces znižovania rozlíšenia(kódovanie) a proces zvyšovania rozlíšenia(dekódovanie). Tieto dve časti sú prepojené takzvaným skip prepojením, umožňujúcim uchovanie informácií z pôvodného obrazu, a tým aj presnejšiu rekonštrukciu poškodených odtlačkov. Celá architektúra sa v princípe skladá z dvoch typov blokov, ktorých tvorba je implementovaná triedami: [28]

- ***UNetDown()***: táto funkcia vytvára 1 blok obstarávací kódovanie, obsahuje: konvolučnú vrstvu, voliteľnú dávkovú normalizáciu, aktivačnú funkciu LeakyReLU a možnosť využitia dropoutu.
- ***UNetUp()***: táto funkcia vytvára 1 blok obstarávací dekódovanie, obsahuje: transponovanú konvolučnú vrstvu, voliteľnú dávkovú normalizáciu, aktivačnú funkciu LeakyReLU a možnosť využitia dropoutu. Navyše v metóde *forward()* sa pomocou funkcie *torch.cat()* spája výstup kóderu a dekóderu, čím sa vytvára skip spojenie.

Parameter dropout zabraňuje pretrénovaniu siete tým, že "vypne" pri iterácii určitý počet neurónov, dropout 0,5 = päťdesiat percent. Sieť je tak menej závislá na špecifických neurónoch a viac robustnejšia. Pri evaluácii či používaní vytrénovanej siete sa dropout technika nepoužíva. V procese tréningu obsahoval generátor 42 vrstiev(vrátane vrstiev obsluhujúcich dropout, normalizáciu, pooling atď.) a takmer 30 miliónov trénovateľných parametrov. Nižšie je ukážka kódu inicializujúca generátor z výslednej implementácie.

```
def __init__(self, input_shape):
    super(Generator, self).__init__()
    channels, _, _ = input_shape
    self.down1 = UNetDown(channels, 64, normalize=False)
    self.down2 = UNetDown(64, 128)
    self.down3 = UNetDown(128, 256, dropout=0.5)
    self.down4 = UNetDown(256, 512, dropout=0.5)
    self.down5 = UNetDown(512, 512, dropout=0.5)
    self.down6 = UNetDown(512, 512, dropout=0.5)

    self.up1 = UNetUp(512, 512, dropout=0.5)
    self.up2 = UNetUp(1024, 512, dropout=0.5)
    self.up3 = UNetUp(1024, 256, dropout=0.5)
    self.up4 = UNetUp(512, 128)
    self.up5 = UNetUp(256, 64)

    final = [nn.Upsample(scale_factor=2),
             nn.Conv2d(128, channels, 3, 1, 1),
             nn.Tanh()]
    self.final = nn.Sequential(*final)
```

Diskriminátor odvodený z PatchGAN architektúry je navrhnutý tak, aby rozlíšil či sú jednotlivé časti obrazu (patche) skutočné alebo falošné. Trieda diskriminátora dedí z PyTorch *nn.Module*. Prijíma *input_shape* ako argument a vracia výstup v tvare *(1, počet segmentov na výšku, počet segmentov na šírku)*. Každý blok diskriminátora, obsahuje konvolučnú vrstvu, voliteľnú vrstvu pre normalizáciu a aktivačnú vrstvu LeakyReLU. Nakoniec siete sa pridá posledná konvolučná vrstva, ktorej výsledkom je jedna hodnota-skóre pre každý blok. Takto vytvorené vrstvy sa spoja pomocou funkcie *nn.Sequential()* a za použitia funkcie *forward()* sa obraz prenesie cez model a vráti sa výstup diskriminátora. Diskriminátor má celkovo 12 vrstiev a približne 1,5 milióna trénovateľných parametrov. [21][26]

5.2 Tréning a optimalizácia

Všeobecný proces tréningu GAN a k tomu potrebných komponentov som popísal už vyššie, v tejto sekcii sa budem venovať postupom konkrétnym pre moju implementáciu. Na prípravu potrebného datasetu bol využitý nástroj Anguli generátor syntetických odtlačkov a generátory poškodenia od Davida Košťáka a Vanessi Jórioovej. Pomocou týchto nástrojov bola vytvorená databáza pozostávajúca z takmer 20 tisíc trojíc: poškodených, originálnych odtlačkov a bitmap popisujúcich plochu poškodenia, obsahuje poškodenia simulujúce atopický ekzém, jazvy, rozličné záhyby a vlasy na ploche senzoru. Následne boli všetky obrázky prevedené do šedej škály pre úsporu výpočetnej náročnosti.

Na načítanie dát počas tréningu bola využitá trieda *Dataloader* z knižnice PyTorch, a ako parameter prijímala triedu *ImageDataset(Dataset)*, ktorá pri zavolaní metódy *getitem(self, index)* vráti tensor obsahujúci trojicu: originálny obrázok, poškodený obrázok

a príslušnú bitmapu, všetko v rozmeroch 512x512. Dataloader navyše dáta transformuje, vyplní do potrebného rozmeru, rozdelí do várok a tie následne zamieša.

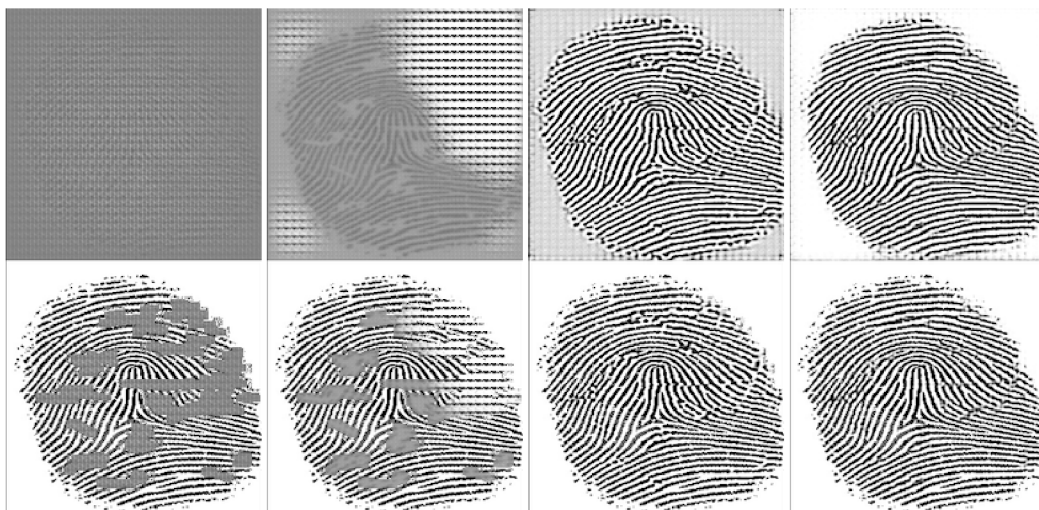


Obrázok 5.2: Na obrázku znázornená jednotka dát potrebných k tréningu siete: originálny odtlačok, poškodený odtlačok a bitmapa poškodenej a prilahlej oblasti

Algoritmus zvolený na optimalizáciu generátora aj diskriminátora bol *Adam* z hodnotou rýchlosti učenia 0,0002 a beta hodnotami 0,5 a 0,99. Ako funkcia na výpočet loss funkcie bola zvolená binárna krížová entropia *BCEWithLogitsLoss()* pre oba modely. Neskôr v tréningu však produkovala pre generátor veľké hodnoty aj napriek vizuálne lepším výsledkom a nebol som schopný s ňou dosiahnuť správne detaily. Preto som pre generátor zaviedol parameter *alpha*, ktorý určoval pomer v akom sa sčítali hodnoty z *BCEWithLogitsLoss()* a novozavedenej strednej kvadratickej chyby *MSELoss()*. Čím ďalej som sa v tréningu dostal, tým menej sa brala do úvahy BCE a prikladal sa väčší dôraz na MSE. V posledných epochách tréningu bola BCE pre generátor úplne odstránená. A výsledné hodnoty MSE sa pohybovali v rozmedzí stotín. [26]

Tréning zabral niekoľko desiatok epoch rozdelených do niekoľkých tréningov, pričom z každého tréningu som uložil diskriminátor a generátor s najnižšou stratou. Oba modely boli vyhodnocované na výsledkoch generátora aplikovaných do pôvodného obrazu.

Jedným z problémov bol príliš "silný" diskriminátor. Diskriminátor sa stal natoľko dobrým vo vyhodnocovaní vstupov, že generátor nebol schopný robiť zmyselné zmeny. Hodnota loss funkcie diskriminátora dosahovala hodnoty okolo 0,3 a tréning tak kolaboval. Ideálne hodnoty loss pre diskriminátor GAN by sa mali pohybovať okolo 0,5 pre správne fungovanie sietí. Tento problém som vyriešil pomocou vyhladenia označení (label smoothing). Pri vyhladzovaní som tvrdé hodnoty 0 a 1 (pre falošné a pravé) nahradil mäkkými hodnotami 0,2 a 0,8. Cieľom vyhladzovania označení je znížiť dôveru modelu v jeho predikcie a zabrániť preučeniu (overfitting). Použitím mäkkých označení je diskriminátor povzbudený k učeniu sa jemnejšej hranice medzi skutočnými a falošnými dátami namiesto závislosti na ostrých hraniciach medzi nimi.



Obrázok 5.3: Na obrázkoch je ukážka rôznych výstupov počas tréningu generátora. V hornej časti obrázku sú výstupy generátora a v spodnej časti je pôvodný poškodený odtlačok, v ktorom sú vložené len vybrané plochy

Počas analýzy priebežných výsledkov som si všimol, že pred výrezom a aplikáciou na pôvodný obraz, generátor produkuje veľmi realistické výsledky. Na základe toho som sa rozhodol vyskúšať alternatívny prístup k tréningu generátora. Namiesto pôvodnej metódy, ktorá zahŕňa detekciu poškodenia a aplikáciu výsledkov generátora na pôvodný obraz, som generátor trénoval a vyhodnocoval bez prevodu výstupu späť na poškodený obraz.

Tento nový prístup priniesol svoje výhody aj nevýhody. Na jednej strane umožnil väčšiu flexibilitu pri generovaní výsledkov, keďže žiaden detektor poškodenia nepotrebuje, nie je obmedzený jeho schopnosťami. Na druhej strane bez explicitnej detekcie poškodenia, a tým zachovania nepoškodených častí pôvodného obrazu, nie sme schopní zaručiť, či generátor nepridal, nezmenil alebo neodstránil pôvodné markanty.

Oba prístupy produkujú takmer totožné výsledky a podrobným porovnaniam sa budem venovať v nasledujúcej kapitole. Napriek tomu považujem tento pokus za úspešný a určite má zmysel ho v budúcnosti rozvinúť. Výber vhodného postupu záleží na špecifických potrebách a prioritách konkrétnej aplikácie.

5.3 Finálny nástroj

Pre používanie finálneho produktu som vytvoril skript v pythone s názvom *reconstruct.py*. Skript spracuje argumenty z príkazovej riadky a načíta vytrénovaný generátor, ktorý bol exportovaný do súboru *generator.pth*. Takto pripravený načíta poškodené odtlačky a bitmapy zo zložky špecifikovanej užívateľom pomocou argumentov *-damaged_f*, *-bitmap_f*. Pokiaľ nie je parameter *-bitmap_f* prítomný, skript vracia priamo výstup generátora bez prenosu na pôvodný obraz. Následne poškodené odtlačky prejdú neurónovou sieťou a výstup sa aplikuje podľa príslušnej bitmapy, ak je prítomná. Výsledok je uložený buď samostatne alebo ako pár pôvodný a zrekonštruovaný odtlačok. Túto funkciu môžete zapnúť pomocou argumentu *-pair*. Príklad spustenia:

```
python3 reconstruct.py -damaged_f /damaged -bitmap_f /bitmap -pair
```

Kapitola 6

Testovanie a analýza

Táto kapitola sa zaoberá praktickým hodnotením metód rekonštrukcie odtlačkov prstov, ktoré som vyvinul. Prvý krok zahŕňa vytvorenie komplexného testovacieho datasetu, ktorý je rozhodujúcim prvkom pri hodnotení týchto metód.

Ďalej predstavím nástroj na testovanie kvality, ktorý je nevyhnutný na zabezpečenie presnosti a spoľahlivosti procesu rekonštrukcie. Následne prevediem podrobnú analýzu výsledkov so zameraním na porovnanie rekonštruovaných odtlačkov prstov s ich pôvodným nepoškodeným stavom. Cieľom tejto kapitoly nie je len overenie metód, ale aj získanie poznatkov, ktoré by mohli usmerniť budúci výskum v tejto oblasti.

Na testovanie dát bolo vytvorených 400 odtlačkov, 100 z každého typu poškodenia (atopický ekzém, záhyby, vlasy, jazvy) následne sú na odtlačok aplikované oba z implementovaných prístupov, čím mi vznikne štvorica: originálny, poškodený, zrekonštruovaný a zrekonštruovaný odtlačok bez použitia bitmapy.



Obrázok 6.1: Príklad poškodených a k nim pomocou bitmapy zrekonštruovaných odtlačkov zľava doprava: atopický ekzém, záhyby, vlasy, jazvy

6.1 Vizuálna analýza

Vizuálna analýza rekonštruovaných obrazov zohráva rozhodujúcu úlohu pri hodnotení efektívnosti techník rekonštrukcie obrazu, a to počas tréningu sietí a aj po ňom.

Na prvý pohľad obe metódy produkujú realistické a správne výsledky. Vo väčšine výsledkov nie sme schopní identifikovať, kde sa nachádzali poškodené miesta a vzor papilárnych sietí vyzerá prirodzene. Problém vzniká pri rozsiahlych poškodeniach ako sú jazvy, ktoré vytvárajú veľké a prázdne plochy. V týchto miestach sú rekonštruované oblasti mierne rozmazané a nie úplne jasné ako môžeme vidieť na obrázku 6.2, riešením by mohol byť ďalší tréning alebo aplikovanie klasickejších metód, ako napríklad prahovanie. Tento problém produkovali oba prístupy.

Ďalším problémom je, že pri prístupe s detekciou poškodenia, sme v niektorých prípadoch vizuálne detektovali miesta, kde boli aplikované záplaty. To značí, že rekonštrukcia nebola úplne presná a mohla viesť k vzniku nových markantov.

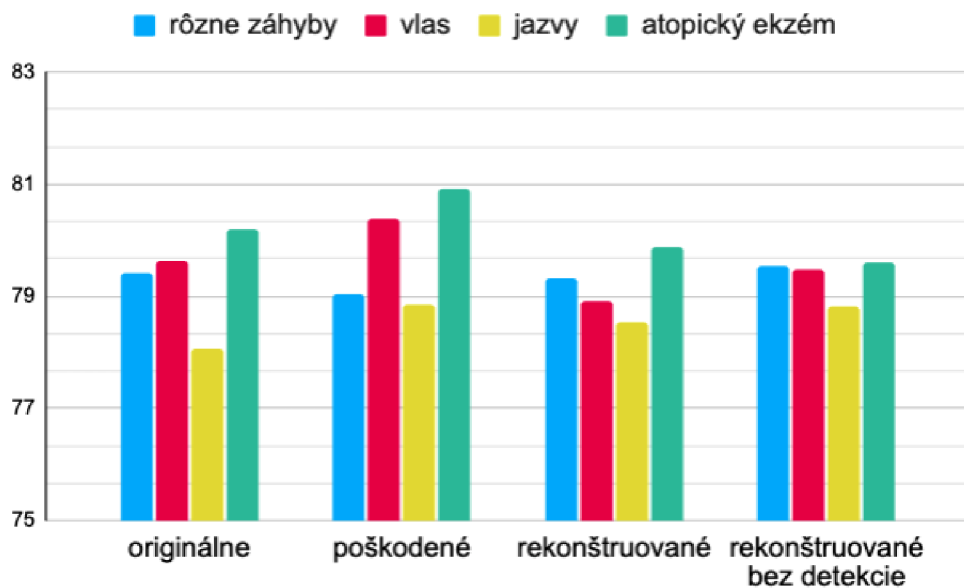
Napriek tomu, vizuálna analýza rekonštruovaných obrazov ukázala sľubné výsledky oboch metód pri rekonštrukcii poškodených častí odtlačkov. To však samo o sebe nie je dostačujúce, a preto v ďalších sekciách používam objektívne metriky, ako je skóre kvality, počet markantov, percento zhody, aby som získal komplexnejšie hodnotenie výkonnosti metód rekonštrukcie.



Obrázok 6.2: Príklad problematických rekonštrukcií.

6.2 Analýza kvality odtlačkov

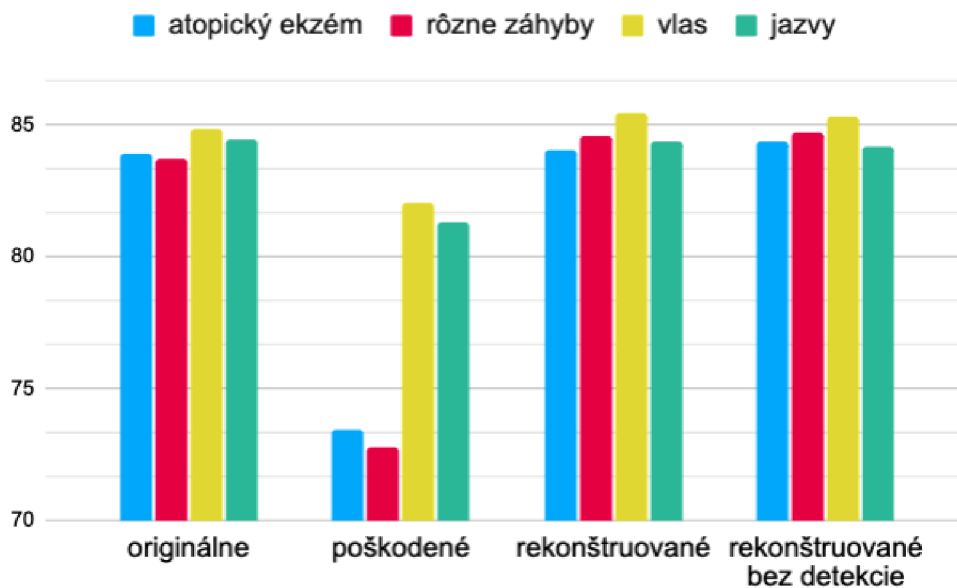
Na evaluáciu kvality bol použitý nástroj FiQiVi. FiQiVi je softvér na hodnotenie kvality odtlačkov prstov, využíva kombináciu lokálnych a globálnych vlastností. Softvér navyše vytvára mapu kvality, ktorá vizuálne znázorňuje definíciu kvality v malých častiach odtlačku prsta. Taktiež zahŕňa implementáciu NFIQ1 a NFIQ2, dvoch zavedených modelov hodnotenia kvality, čo ďalej rozširuje jeho možnosti.



Obrázok 6.3: Priemerné skóre NFIQ2 podľa poškodenia.

Skóre z nástroja NFIQ2 som musel zamietnuť. Výsledky tohto nástroja boli nekonzistentné a nebol schopný správne posúdiť odtlačky. Paradoxné bolo, že skóre pre poškodené odtlačky bolo v priemere mierne vyššie ako pre originálne odtlačky.

Existuje niekoľko možných dôvodov tohto javu. Jedným z nich je, že poškodenia v odtlačkoch vytvárajú viac markantov, čo spôsobuje vyššie skóre kvality. Ďalším možným dôvodom je, že nástroj NFIQ2 vníma chýbajúce plochy ako pozadie, čo môže ovplyvniť hodnotenie kvality odtlačkov. Táto problematika si určite vyžaduje ďalšie podrobnejšie skúmanie, avšak v rámci tejto práce sa jej nevenujem. Preto som sa rozhodol použiť nástroj FiQiVi, ktorý poskytuje spoľahlivejšie výsledky a zabezpečuje konzistentné hodnotenie bez ohľadu na stav odtlačkov prstov. Následujúce výsledky pochádzajú z nástroja FiQiVi.



Obrázok 6.4: Priemerné skóre FiQiVi podľa poškodenia.

Z grafu vyššie vidíme, že oba prístupy majú schopnosť nielen zlepšiť kvalitu poškodeného odtlačku, ale dokonca ju zvýšiť v porovnaní s pôvodným odtlačkom. Možným vysvetlením tohto je, že Anguli generuje odtlačky, ktoré nie sú dokonalé a môžu obsahovať mierne množstvo poškodenia. Výsledky oboch prístupov produkujú odtlačky takmer rovnakej kvality, preto som sa rozhodol preskúmať presnosť, s akou tieto prístupy rekonštruujú odtlačky.

6.3 Porovnanie presnosti rekonštrukcie

Presnosť rekonštrukcie odtlačkov je kľúčovým faktorom pri hodnotení účinnosti týchto prístupov a môže mať veľký vplyv na funkčnosť porovnávacích systémov. Preto som sa venoval preskúmaniu presnosti, s akou tieto prístupy dokážu odtlačky rekonštruovať. Využil som globálne aj lokálne kritériá na porovnanie rekonštruovaných odtlačkov s originálnymi.

Tabuľka nižšie prezentuje percentuálne zhody medzi originálnym odtlačkom a výstupmi, ktoré boli generované dvoma odlišnými prístupmi. Na prvý pohľad môžeme konštatovať, že zhoda je v oboch prípadoch pomerne vysoká. Ale pri bližšom skúmaní je možné si všimnúť, že prístup, pri ktorom najskôr identifikujeme poškodenie a potom na tento poškodený obraz aplikujeme len záplaty, produkoval mierne vyššie výsledky. Hoci sa môže zdať, že rozdiel je len niekoľko percent, v skutočnosti tento rozdiel môže mať významný vplyv. Môže viesť k zániku, vzniku alebo zmene niekoľkých markantov, čo by malo byť zohľadnené pri hodnotení výsledkov.

| Metóda | Celkový priemer | Atopický ekzém | Rôzne záhyby | Vlas | Jazvy |
|----------------------------------|-----------------|----------------|--------------|--------|--------|
| Originálny/opravený | 97,081 | 97,400 | 95,082 | 96,687 | 99,156 |
| Originálny/opravený bez detekcie | 96,161 | 96,239 | 94,659 | 96,087 | 97,659 |

Tabuľka 6.1: Tabuľka poskytuje podrobný prehľad percentuálnej zhody medzi originálnym odtlačkom a výstupmi generovanými dvoma použitými prístupmi v kontexte rôznych typov poškodení.

Po výsledkoch percentuálnej zhody som sa rozhodol bližšie preskúmať rozdiely na úrovni markantov. Tabuľka 6.2 prezentuje rozdiely v počte markantov medzi dvoma metódami rekonštrukcií odtlačkov pre rôzne typy poškodení. Vysoký rozdiel v počte markantov môže signalizovať nižšiu presnosť rekonštrukcie.

Pri atopickom ekzéme, rôznych záhybov a vlase metóda detekcie a záplat generuje viac markantov oproti metóde priameho generovania bez detekcie, čo môže byť dôsledkom toho, že v miestach vloženia záplaty vzniknú nežiadúce markanty, napríklad ukončenia. Naopak, pri jazvách je situácia opačná, kde metóda detekcie a záplat produkuje menej markantov, čo naznačuje presnejšiu rekonštrukciu.

| Metóda | Atopický ekzém | Rôzne záhyby | Vlas | Jazvy |
|----------------------------------|----------------|--------------|-------|-------|
| Originálny/opravený | 23,86 | 23,47 | 24,77 | 10,55 |
| Originálny/opravený bez detekcie | 21,32 | 21,16 | 16,84 | 20,49 |

Tabuľka 6.2: Tabuľka ukazuje rozdiel v počte markantov medzi originálnym odtlačkom a použitými prístupmi pri jednotlivých poškodeniach

Na základe týchto výsledkov je dôležité zvážiť typ poškodenia a príslušné priority pri výbere a implementácii metód opravy odtlačkov.

Kapitola 7

Záver

Cieľom tejto práce bolo zrekonštruovať poškodené časti obrázkov odtlačkov prstov pomocou neurónových sietí. Na tento problém som si zvolil kontextové GAN. Aby bolo možné trénovať generatívne modely, je vhodné mať veľké množstvo tréningových dát. Na tento účel som použil generátor odtlačkov prstov Anguli a dva ďalšie nástroje na simuláciu poškodenia a vytvoril dataset pozostávajúci z približne 20 tisíc odtlačkov prstov.

Na tréningovanie siete boli navrhnuté a implementované dva prístupy. Prvý z nich vyžaduje detekciu poškodenia a pomocou nej premieta výstup generátora späť na poškodený odtlačok. Druhý nevyžaduje žiadnu detekciu, prijíma poškodený odtlačok a generuje priamo jeho rekonštrukciu.

Výsledky boli slubné a ukázali, že oba prístupy dokážu presne rekonštruovať umelé odtlačky prstov. Použitie kontextových GAN mi umožnilo generovať obrazy, ktoré boli nielen realistické, ale zachovávali aj pôvodné vlastnosti odtlačku prsta.

Aj keď architektúra neurónových sietí vyvinutých v tejto práci nie je optimálna z hľadiska efektivity a využitia pamäte, optimalizácia týchto aspektov nebola primárnym cieľom tejto práce. Hlavným cieľom bolo dokázať možnosť použitia neurónových sietí na rekonštrukciu obrazu odtlačkov prstov. Napriek tomu je potrebné poznamenať, že ďalším výskumom a zlepšením by sa architektúra siete mohla zredukovať tak, aby bola rýchlejšia a pamäťovo efektívnejšia bez toho, aby sa znížila kvalita výsledkov.

Hoci som sa zamerlal na syntetické odtlačky prstov, je vysoko pravdepodobné, že môj prístup by sa dal po dodatočnom tréningu na vhodných dátach aplikovať na skutočné odtlačky prstov. Moja práca poskytuje základ pre budúci výskum v tejto oblasti. Okrem toho môže mať táto práca praktické využitie pri forenznom vyšetrení a v iných oblastiach, kde sa vyžadujú vysokokvalitné obrazy odtlačkov prstov. Po ďalšom vývoji by môj prístup mohol pomôcť ľuďom postihnutým rôznymi ochoreniami využívať biometrické systémy a uľahčiť tak ich každodenný život.

Moje zistenia by mohli byť užitočným východiskom pre budúci výskum, ktorý sa zaoberá o rekonštrukciu obrazov odtlačkov prstov. Existuje niekoľko smerov ďalšieho skúmania vrátane skúmania limitov môjho prístupu a optimalizácie architektúry modelu a parametrov tréningovania. Celkovo moja práca poukazuje na potenciál využitia kontextových GAN na rekonštrukciu poškodených častí obrazov odtlačkov prstov a splňuje všetky body zadania.

Literatúra

- [1] AGGARWAL, C. C. *Neural networks and deep learning*. Cham: Springer, [2018]. ISBN 978-3-319-94462-3.
- [2] AGHDAM, H. H. a HERAVI, E. J. *Guide to Convolutional Neural Networks*. 1. vyd. Springer International: Springer Cham, 2017. ISBN 978-3-319-57549-0.
- [3] BAK, A. *Simulation of Skin Diseases Effect Using GAN*. Brno, 2021. Master's thesis. Brno University of Technology, Faculty of Information Technology.
- [4] BAROTOVÁ Štěpánka. *Detektor kožních onemocnění u technologie otisků prstů*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav inteligentních systémů. Dostupné z: <http://hdl.handle.net/11012/69692>.
- [5] BASAVARAJAIAH, M. *6 basic things to know about Convolution*. Citované: 6 marec 2023. Dostupné z: <https://medium.com/@bdhuma/6-basic-things-to-know-about-convolution-daef5e1bc411>.
- [6] BOLOGNA, D.-U. of. *Fingerprint Generation, SFinGe*. Dostupné z: <http://biolab.csr.unibo.it/research.asporganize=Activities&select=&selObj=12&pathSubj=111%7C%7C12&#>.
- [7] BROWNLEE, J. *A Gentle Introduction to Generative Adversarial Networks (GANs)*. 2019. Citované: 21 marec 2023. Dostupné z: <https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/>.
- [8] DRAHANSKY, M., DOLEZEL, M., URBANEK, J., BREZINOVA, E. a KIM, T. hoon. Influence of Skin Diseases on Fingerprint Recognition. *Journal of Biomedicine and Biotechnology*. 2012, zv. 2012. DOI: 10.1155/2012/626148. ISSN 1110-7243. Dostupné z: <http://www.hindawi.com/journals/bmri/2012/626148/>.
- [9] DRAHANSKÝ, M. *Hand-Based Biometrics: Methods and Technology*. The Institution of Engineering and Technology, 2018. ISBN 978-1-78561-224-4. Dostupné z: <https://www.fit.vut.cz/research/publication/11556>.
- [10] DRAHANSKÝ, M. a ORSÁG, F. Fingerprints and Speech Recognition as parts of the biometry. In: *Proceedings of 36th International Conference MOSIS '02*. 2002. ISBN 80-85988-71-2. Dostupné z: <https://www.fit.vut.cz/research/publication/6903>.
- [11] DRAHANSKÝ, M. a ORSÁG, F. *Biometrie*. 1.vydanie. Brno: Computer Press a. s., 2011. ISBN 978-80-254-8979-6.

- [12] GOODFELLOW, I., BENGIO, Y. a COURVILLE, A. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [13] GOODFELLOW, I. J. *NIPS 2016 Tutorial: Generative Adversarial Networks*. 2017. Dostupné z: <http://arxiv.org/abs/1701.00160>.
- [14] GOODFELLOW, I. J., POUGET ABADIE, J., MIRZA, M., XU, B., WARDE FARLEY, D. et al. *Generative Adversarial Networks*. 2014. Dostupné z: <https://arxiv.org/abs/1406.2661>.
- [15] GREG FIUMARA, A. F. *NFIQ 2 github readme*. NIST, 2011. Citované: 12 november 2022. Dostupné z: <https://github.com/usnistgov/NFIQ2/blob/master/README.md>.
- [16] HARA, M. *Fingerprint Image Enhancement*. Boston, MA: Springer US, 2009. 474–482 s. ISBN 978-0-387-73003-5. Dostupné z: https://doi.org/10.1007/978-0-387-73003-5_49.
- [17] HENRY FAULDS. *Henry Faulds: the Invention of a Fingerprinter* [<https://galton.org/fingerprints/faulds.htm>]. [Accessed: May 10, 2023].
- [18] HONG, L. a JAIN, A. *Fingerprint Enhancement*. New York, NY: Springer New York, 2004. 127–143 s. ISBN 978-0-387-21685-0. Dostupné z: https://doi.org/10.1007/0-387-21685-5_7.
- [19] HONG, L., WAN, Y. a JAIN, A. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1998, zv. 20, č. 8, s. 777–789. DOI: 10.1109/34.709565.
- [20] INVESTIGATION, F. B. for. *Privacy Impact Assessment for the Fingerprint Identification* [online]. FBI services informaion management, máj 2008. Citované: 9 október 2022. Dostupné z: <https://www.fbi.gov/services/information-management/foipa/privacy-impact-assessments/firs-iafis>.
- [21] ISOLA, P., ZHU, J.-Y., ZHOU, T. a EFROS, A. A. *Image-To-Image Translation With Conditional Adversarial Networks*. July 2017.
- [22] KOŠŤÁK, D. *Generování projevu kožního onemocnění do syntetických otisků prstů z generátoru Anguli*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav inteligentních systémů. Dostupné z: <http://hdl.handle.net/11012/85198>.
- [23] LAKNA. *Difference Between Dermis and Epidermis* [online]. Pediaa.Com, december 2017. Citované: 9 október 2022. Dostupné z: <https://pediaa.com/difference-between-dermis-and-epidermis/>.
- [24] MALTONI, D., MAIO, D., JAIN, A. K. a PRABHAKAR, S. *Handbook of Fingerprint Recognition*. 2nd ed. 2009. London: Springer London, 2009. ISBN 978-1-8488-2254-2.
- [25] PRABHU. *Understanding of Convolutional Neural Network (CNN) — Deep Learning*. Citované: 6 marec 2023. Dostupné z: <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>.

- [26] PYTORCH CONTRIBUTORS. *PyTorch Documentation*. 2023. Citované: 3 apríl 2023. Dostupné z: <https://pytorch.org/docs/stable/index.html>.
- [27] RADFORD, A., METZ, L. a CHINTALA, S. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. Dostupné z: <https://arxiv.org/abs/1511.06434>.
- [28] RONNEBERGER, O., FISCHER, P. a BROX, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Cham: Springer International Publishing, 2015. ISBN 978-3-319-24574-4.
- [29] VOLNÁ, E., KOTYRBA, M., JANOŠEK, M. a KOCIAN, V. *Umělá inteligence: rozpoznávání vzorů v dynamických datech*. 1.vydanie. Praha: BEN - technická literatura, 2014. ISBN 978-807-3004-972.
- [30] YUN, E.-K. a CHO, S.-B. Adaptive fingerprint image enhancement with fingerprint image quality analysis. *Image and vision computing*. AMSTERDAM: Elsevier B.V. 2006, zv. 24, č. 1, s. 101–110. ISSN 0262-8856.
- [31] ZAERI, N. *Biometrics: Minutiae-based Fingerprint Extraction and Recognition*. 1.vydanie. Londýn: IntechOpen, 2011. ISBN 953-307-618-6.