VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

DETEKCE A KLASIFIKACE POŠKOZENÍ OTISKU PRSTU S VYUŽITÍM NEURONOVÝCH SÍTÍ

DETECTION AND CLASSIFICATION OF DAMAGE IN FINGERPRINT IMAGES USING NEURAL NETS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **MILAN ŠALKO**
AUTHOR

VEDOUCÍ PRÁCE **Ing. ONDŘEJ KANICH, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Šalko Milan**
Program: Informační technologie
Název: **Detekce a klasifikace poškození otisku prstu s využitím neuronových sítí**
Detection and Classification of Damage in Fingerprint Images Using Neural Nets
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte literaturu týkající se biometrického rozpoznávání podle otisků prstů s důrazem na poškozené či nekvalitní snímky z důvodu onemocnění na prstu. Seznamte se z možnostmi detekce a klasifikace s využitím neuronových sítí.
2. Navrhněte algoritmus který posoudí, jaká část otisku je poškozená. V místech s poškozením pak algoritmus klasifikuje dané onemocnění (bradavici a dishydrózu).
3. Implementujte navržený algoritmus z předchozího bodu.
4. Otestujte algoritmus na dostupných databázích otisků prstů (poškozených i nepoškozených).
5. Dosažené výsledky shrňte a diskutujte. Uveďte možná vylepšení a rozšíření Vašeho řešení.

Literatura:

- Drahanský, M.: *Hand-Based Biometrics: Methods and technology*, IET 2018, pages 430, ISBN 978-1-78561-224-4.
- 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.
- Maltoni, D., Maio, D., Jain, A.K. and Prabhakar, S.: *Handbook of Fingerprint Recognition*. Springer, 2009, pages 512. ISBN 978-1-8488-2254-2.

Pro udělení zápočtu za první semestr 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. listopadu 2019
Datum odevzdání: 28. května 2020
Datum schválení: 31. října 2019

Abstrakt

Cieľom tejto práce je naštudovať a navrhnúť algoritmus pre detekciu poškodení spôsobenými kožnými ochoreniami na odtlačku prsta, menovite sú týmito ochoreniami bradavice a dyshidróza. Pre detekciu prejavov ochorení bola využitá konvolučná neurónová sieť založená na frameworku Keras. Táto sieť posúdi aká časť odtlačku prsta je poškodená a v poškodených oblastiach klasifikuje dané ochorenie. Pri trénovaní siete boli využité syntetické odtlačky doplnené reálnymi odtlačkami prstov.

Abstract

The aim of this bachelor thesis is to study and design algorithm for detection of fingerprint damage caused by skin disease, specifically by wart and dyshidrosis. Symptome detection was implemented by convolutional neural network based on Keras framework. This network determine, which part of finger is damaged and in these areas will classify the disease. Combination of synthetic and real fingerprints was used to train the neural network.

Kľúčové slová

konvolučné neurónové siete, odtlačky prsta, bradavice, dyshidróza, klasifikácia poškodenia, lokalizácia poškodenia

Keywords

convolutional neural networks, fingerprints, warts, dyshidrosis, damage classification, damage location

Citácia

ŠALKO, Milan. *Detekce a klasifikace poškození otisku prstu s využitím neuronových sítí.* Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ondřej Kanich, Ph.D.

Detekce a klasifikace poškození otisku prstu s využitím neuronových sítí

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Ondreja Kanicha Ph.D. Uviedol som všetky literálne pramene, publikácie a dalšie zdroje z ktorých som čerpal.

.....
Milan Šalko
28. mája 2020

Poďakovanie

Chcel by som veľmi pekne poďakovať môjmu vedúcemu práce doktorovi Ondřejovi Kanichovi, za jeho rady, ústretový prístup, trpezlivé odpovede a v neposlednom rade za vedenie tejto práce.

Obsah

Kapitola 1

Úvod

Keď sa spomenú odtlačky prstov väčšinu ľudí napadne, využitie v kriminalistike. No používanie odtlačkov preniká stále viac do každodenných oblastí našich životov. Dnes už skoro každý mobilný telefón dokáže snímať odtlačky prstov, pomocou ktorých sa nielen odomykajú naše zariadenia, ale aj sa nimi prihlasuje do rôznych aplikácií, či sa potvrdzujú platby v internetovom bankovníctve. Za zmienku stojí aj možnosť nahradenia prístupových kariet do rôznych budov. Je pravdepodobné, že časom pribudnú ďalšie možnosti využitia.

Problémy s využitím odtlačkov prstov nastavajú pri poškodení odtlačku prsta, spôsobeným znečistením alebo kožným ochorením. Tieto poškodenia môžu spôsobiť falošnú detekciu markantov čo je následne problém pri identifikácií osoby. Práca je zameraná na detekciu a klasifikáciu poškodení spôsobených ochoreniami zasahujúcich odtlačky prstov. Sledovanými ochoreniami sú bradavice a dyshidróza. Ak sieť zaznamená poškodenie odtlačku takéto miesto vyznačí a záznam o jeho pozícií uloží do súboru.

Pre tento proces bude využívaná konvolučná neurónová sieť, ktorá bude vytvorená vo frameworku TensorFlow. V dnešnej dobe sú neuronové siete obrovským trendom nielen na poli rozpoznávania ale vďaka čoraz väčšiemu uplatneniu v praxi prenikajú stále viac do našich každodenných životov. V súčastnosti sa len ťažko dá nestretnúť s jej využitím, skvelým príkladom sú internetové vyhľadávače alebo sociálne siete kde neurónová sieť preberá niektoré úlohy. Ďalšími oblasťami kde sa s nimi môžme stretnúť sú možnosti viesť autonómne vozidlá, vyhľadávanie obrázkov či identifikácia osôb na záznamoch kamier.

Cieľom tejto práce je návrh a implementácia algoritmu za pomoci neurónovej siete. Táto sieť posúdi aká časť odtlačku prsta je poškodená a v poškodených oblastiach klasifikuje dané ochorenie (bradavicu a dyshidrózu). V kapitole 2 je popísaná štruktúra kože, papilírne línie a samotné ochorenia (bradavica a dyshidróza). Biometria a spracovanie odtlačkov prstov je témou 3 kapitoly. Poslednou teoretickou kapitolou je kapitola 4, ktorá popisuje neurónové siete a špeciálne je zameraná na konvolučné neuronové siete a ich učenie. V kapitole 5 je popísaný návrh riešenia práce. Samotná implementácia riešenia sa nachádza v kapitole 6. Poslednou témou je testovanie a popis výsledkov, ktoré sú popísané v kapitole 7.

Kapitola 2

Popis kože a vybraných chorôb

V tejto kapitole sú predstavené funkcie kože, jej štruktúra a z akých vrstiev sa koža skladá. Ďalej bude popísané, čo sú papilárne línie, kedy nastáva ich vývoj a čo ovplyvňuje ich výsledný vzor. V druhej časti tejto kapitoly 2.3 budú popísané samotné ochorenia a tiež, akú časť populácie zasahujú. Popísaný bude aj spôsob, ako sa sa prejavuje ich poškodenie na samotných odtlačkoch prstov.

2.1 Popis kože

Koža je dôležitý orgán ľudského tela pokrývajúci jeho povrch. Plní niekoľko funkcií, medzi najvýznamnejšie patrí ochranná funkcia pred vonkajšími vplyvmi. Zároveň sú v nej uložené aj receptory tlaku, tepla a bolesti, vďaka ktorým sme v kontakte s vonkajším svetom. Ďalšou, nie menej podstatnou funkciou, je termoregulácia a vylučovanie odpadových látok. [12]

Dermatológia je oborom, ktorý popisuje štruktúru kože, jej deriváty a choroby, ktoré ju postihujú. Plocha kože sa pohybuje od 1,5 do 1,8 m^2, priemerná hmotnosť je 4,5 kg. Jej hrúbka sa môže byť od 0,5 do 1,5 mm. Najtenšia je v oblastí očí, najhrubšia na dlaniach a chodidlách. Koža pozostáva z troch hlavných častí [12]:

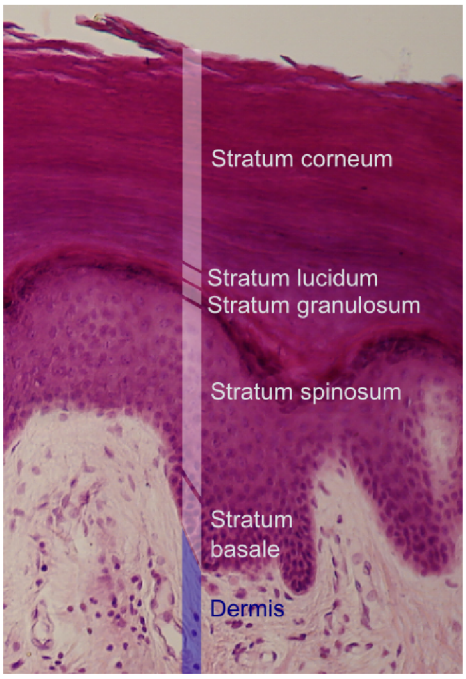
1. Pokožka – (*epiderma*) je vrchná vrstva kože.
2. Zamša – na pokožku naväzuje zamša (*dermis*).
3. Podkožie – zamša ďalej leží na podkoží (*hypodermis*).

2.1.1 Epiderma

Najvrchnejšia vrstva kože je epiderma, tvoria ju hlavne vrstvy dlaždicovo usporiadaných buniek. Tieto vrstvy sú rozlíšiteľné pod mikroskopom na 5 hlavných vrstiev, znázornených na obrázku 2.1. Do najspodnejšej vrstvy epidermy prenikajú výbežky dermy, ktoré sú zodpovedné za výsledný vzor papilárnych línií odtlačku prsta. [12]

Vrstva **stratum basale** je najspodnejšou časťou pokožky, obsahuje jednu vrstvu kubicko-cylindrických buniek. Tieto bunky sú pripojené k bazálnej membráne, ktorá oddeľuje epidermu od dermy. Táto membrána je zvlnená. Vrstva má za úlohu deliť sa a obnovovať vrchné častí pokožky. **Stratum spinosum** sa nachádza nad vrstvou *stratum basale*. Táto vrstva je hrubá 5-10 buniek. Podieľa sa na fyzickej odolnosti pokožky vďaka keratínovým filamentom. Vrstva **stratum granulosum** sa skladá zo živých buniek keratinocitov, ktoré sú mierne

sploštené. Produkujú veľké množstvo keratínu, ktorý sa ukladá v zrnách, podľa čoho táto vrstva dostala názov. Bunková membrána sa zosiluje a bunky vylučujú epiderálne lipidy, ktoré sú prepojené pomocou proteínov, čím vytvárajú bariéru epidermy. Vrstva **stratum lucidum** sa nachádza len na dlaniach a chodidlách. Skladá sa z 2-3 vrstiev priesvitných keratinocitov. Podľa prítomnosti tejto vrstvy rozlišujeme tzv. hrubú a tenkú pokožku. **Stratum corneum** je najvrchnejšia vrstva pokožky. Pozostáva z 15-20 vrstiev sploštených a zrohovatených buniek. Tieto bunky sa nazývajú korneocyty. Tieto bunky sú považované za mŕtve, nakoľko pri bunkovej diferenciácií zmizli všetky bunkové organely. Táto vrstva je najvyvinutejšia na miestach kože, kde je vysoký tlak, napríklad na dlaniach a chodidlách.



Obr. 2.1: Štruktúra epidermy. Zdroj: [15].

2.1.2 Derma a hyperderma

Derma sa nachádza sa pod epidermou, oddelená je bazálnou membránou. Hlavnou zložkou tejto vrstvy je kolagén, tvorí 70 % obsahu. Kolagénové vlákna dodávajú derme odolnosť voči napätiu a ťahu, zatiaľ čo elastínové vlákna dodávajú koži pružnosť. V tejto vrstve je množstvo nervových zakončení (Ruffiniho, Meissnerove telieska), tiež sa tu nachádzajú krvné vlásočnice, no najpodstatnejšími kožnými orgánmi pre daktyloskopiu sú potné žľazy, ktoré ústia na vyvýšeninách odtlačkov prstov (papilárne línie). Dermu rozdeľujeme na [31]:

1. **Stratum papillare** – v podobe papíl vybieha do epidermy, ktorú vrásni a tým vznikajú papilarne línie (pozri. 2.1.3), obsahuje riedke kolagénové väzivo. Sú v nej prítomné rôzne nervové zakončenia.

2. **Stratum reticulare** – obsahuje silnú vrstvu kolenového a elastínového väziva, nachádza sa v nej menej buniek.

Hyperderma je najspodnejšou časťou kože. Slúži ako ochrana proti nárazom a je v nej uložená energia v podobe tukových lalôčikov.

2.1.3 Papilárne línie

Papilárne línie sa vyskytujú na rukách a nohách. Dosahujú výšku od 0,1 do 0,4 mm a hrúbku 0,2 do 0,5 mm. Z funkčného hľadiska sa vyvinuli pre zníženie kĺzania pri uchopení predmetov, papilárne línie tiež zvyšujú citlivosť vnemov na bruškách prstov. Na ich vrcholoch sa nachádzajú vyústenia potných žliaz, ktoré je možné pozorovať pri vyššej kvalite snímku ako malé bodky. Papilárne línie vytvárajú na bruškách prstov odtlačky.

Tvar a smer papilarnych línií čiastočne vychádza z genetickej informácie, takže sa nejedná o náhodne vzory. Istá podobnosť počtu papilárnych línií, ich hĺbky či usporiadania, môže byť sledovaná medzi deťmi a rodičmi. Podľa [20] je najväčšia podobnosť pozorovaná u identických dvojčat, no ich odtlačky aj tak nie sú identické. Na unikátnosť odtlačkov vplývajú aj zmeny v mikro-prostredí počas vývoja špičiek prstov, na ktoré pôsobí obrovské množstvo faktorov. Pravé tieto malé zmeny vytvárajú unikátnosť odtlačkov prstov. [20]

Papilárne línie sa začínajú formovať počas 3-4 mesiaca vývoja plodu. Počas fázy vývoja dochádza k množeniu buniek, kedy sa vytvárajú malé výbežky nazývané papily. Ako už bolo spomenuté v kapitole 2.1.2, tieto papily deformujú pokožku nad sebou, čím vznikajú papilárne línie. Plne vyvinuté sú v siedmom mesiaci tehotenstva. Počas celého života sú tieto línie stále rovnaké. Presnejšie povedané, ako telo rastie, papily zväčšujú svoju plochu, no ich vzor zostava zachovaný. Papilárne línie nie je možné zmeniť alebo odstrániť. Jediným spôsobom je odstránenie zárodočnej vrstvy pokožky. [20]

2.2 Syntetické odtlačky prsta

Množstvo riešení pracujúcich s odtlačkami prstov tak ako aj táto práca potrebuje pre svoj vývoj dáta. Problémom sú však veľkosti databáz a tak sa často vývojové tímy či už v priemysle alebo vo výskume musia zaobísť s malými databázami. Aj keby bola vytvorená databáza s reálnymi odtlačkami jej zdieľane medzi rôznymi tímami by bolo problematické kvôli ochrane osobných údajov osob.

A tak sa ako potencionálna alternatíva ku zbieraniu masívnych databáz ukazuje generovať syntetické odtlačky, ktoré budú spĺňať alebo sa aspoň blížiť parametrami k tým reálnym. S týmito nástrojmi je možne vytvárať online odtlačky (získane zo skenerov) ako aj tie offline (využívajúce klasickú atramentovú metódu). Pri tejto simulácií musí byť kladený dôraz na intra a inter triednu premenlivosť. To znamená v databáze by mali byť rôzne odtlačky prstov (intertriedna premenlivosť) ako aj rôzne verzie jedného odtlačku (intratriedna premenlivosť) líšiace sa napríklad v sile odtlačenia jednotlivých časti či iných premenlivých faktoroch.

Syntetické odtlačky sú vytvárane z takzvaného hlavného odtlačku čo by sme mohli považovať za dokonalý odtlačok. Pre vytvorenie hlavného odtlačku potrebujeme tri veci. Ako prve je to samotná plocha odtlačku. Ďalej je to vytvorenie smerového poľa kde si určíme singulárne body a pripadne ďalšie parametre odtlačku. Poslednou súčasťou je hustota papilárnych línií. Po spojení týchto troch vecí dostaneme hlavný odtlačok. Po jeho vygenerovaní je možne z tohto odtlačku vytvárať ďalšie odtlačky ktorých výsledný stav je ovplyvnený len par parametrami. Ako je veľkosť odtlačenej plochy, nasmerovanie, či sila a plocha poškodenia.

2.3 Popis vybraných chorôb

Kožné choroby ovplyvňujú nielen zdravotný stav dotyčnej osoby, ale aj jej možnosť pracovať s niektorými technológiami. Niektoré poškodenia sú pomerne ľahko odstrániteľné – napríklad bradavice. No sú aj také, ktoré bruška prstov postihujú v takej miere, že oprava týchto odtlačkov nie je možná. V práci som sa zameral na dve ochorenia - dyshidrózu a bradavice. V ďalších podkapitolách budú tieto choroby bližšie predstavené.

2.3.1 Bradavice (verrucae vulgares)

Jedna sa o jedno z najrozšírenejších ochorení kože, spôsobované ľudským papilomavirusom, a to jeho typmi 1, 2, 4 a 7. Do organizmu sa dostávajú najčastejšie kožnými poraneniami. Inkubačná doba sa pohybuje od 4 týždňov do 8-20 mesiacov [17]. Hlavným prejavom je hyperkeratóza – jedná sa o zhrubnutie vrchnej vrstvy epidermy (*stratum corneum*). [32]

V populácií najviac postihuje deti, ale vyskytujú sa aj u dospelých. Najčastejšie sa vyskytuje na rukách a nohách, ale môže sa objaviť kdekoľvek na tele. Obvykle sa jedná o zrohovatené tvrdé výrastky kože kruhového tvaru o veľkosti okolo 5 mm. Niektoré výrastky však môžu dosahovať až okolo 1 cm. Občas sa okolo „materskej“ bradavice vyskytne druhotný výsev tzv. „dcérskych bradavíc“. Ďalším znakom môžu byť tmavé bodky na povrchu tohto útvaru, jedná sa o trombotizované kapiláry. [17]

Liečenie bradavíc nie je nijak zvlášť jednoduché, obvykle sa na ich odstraňovanie používa tekutý dusík, sneh z CO_2, kyselina salicová alebo iné leptadlá. V poslednej dobe sa používa aj terapia laserom [17]. Po odstránení materskej bradavice dcérske bradavice často zmiznú samé.

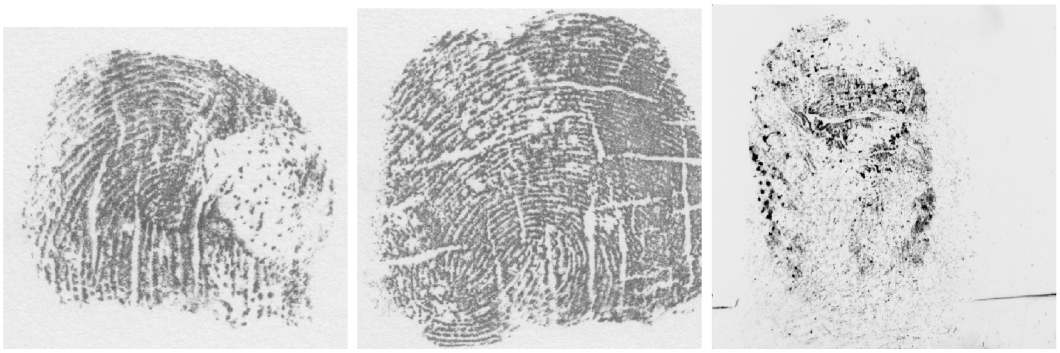
Na odtlačkoch prstov sa bradavice prejavujú ako biele, skoro okrúhle plochy s čiernymi bodkami, viď obrázok 2.2. Pri menších bradaviciach sa bodky vnútri bradavice nemusia objaviť vôbec. Ďalším častým znakom bradavíc je pomerne zreteľný tmavý okraj okolo bradavice. Poškodenie touto chorobou nie je až také závažné, nakoľko nie je zasiahnutá tak veľká oblasť, ako pri dyshidróze. Papilárne línie sú na zvyšku prsta dobre viditeľné [11].

Obr. 2.2: Prejavy bradavíc na odtlačkoch prstov. Zdroj: Interná databáza skupiny STRaDe.

2.3.2 Dyshidróza (dyshidrotic dermatitis)

Dyshidróza (*dyshidrotic dermatitis*), občas nazývaná aj *pompoholyx* [12]. Presný pôvod ochorenia nie je úplne známy, ale je pravdepodobné, že ochorenie vzniká v dôsledku styku s kontaktným alergénom.

Toto ochorenie postihuje ruky a nohy, na bokoch prstov a dlaniach sa objavujú svrbiace pľuzgieriky, ktoré sa môžu spájať až do búl, tieto pľuzgieriky potom praskajú, vlhnú a vysychajú. Koža na mnohých miestach po vyschnutí praská a objavujú sa hlboké bolestivé praskliny. Sprievodným javom môže byť zápal týchto pľuzgierikov. U niektorých pacientov sa toto ochorenie prejavuje každé leto 3-4 týždne, u iných sa jedná o trvalý problém. Časť pacientov je alergická na lieky či kozmetiku a vyhýbanie sa danému alergénu rieši problém s ochorením. [12]



Obr. 2.3: Prejavy dyshidrózy na odtlačkoch prstov. Zdroj: Interná databáza skupiny STRaDe.

Ako môžme vidieť na obrázku 2.3 sú odtlačky prstov postihnuté týmto ochorením, poškodené typicky hrubými bielymi čiarami ktoré pretínajú papilárne línie. Tieto čiary sa objavujú prakticky na celom odtlačku prsta a majú rôzne smery a dĺžku. Ďalším znakom sú biele oblasti. Ich veľkosť a rozsah závisí od závažnosti poškodenia. V miernejších prípadoch sú časti odtlačku zasiahnuté malými bielymi škvrnami. V týchto prípadoch je stále možné pomerne dobre pozorovať papilárne línie. Tieto oblasti škvŕn začínajú obvykle na krajoch a postupujú smerom do centra odtlačku. Pri vážnejších prípadoch sú tieto odtlačky zasiahnuté veľkými bielymi plochami ktoré nemajú presne definovateľný tvar. Tieto plochy občas môžu obsahovať menšie tmavé škvrny. Ochorenie zasahuje prakticky celu oblasť odtlačku a tieto odtlačky sa stávajú nepoužiteľnými. [11]

Kapitola 3

Biometria a spracovanie odtlačkov prstov

V nasledujúcich podkapitolách bude popísaná biometria a aké sú jej princípy. Popísaný bude aj význam odtlačku prsta v biometrií a rozdeľovanie odtlačkov prstov. V ďalšej časti kapitoly si predstavíme predspracovanie odtlačkov a často používané techniky.

3.1 Biometria

V dnešnej dobe je potreba zabezpečeného prístupu (k dátam, do budov, ...) oveľa dôležitejšia ako predtým. Častým spôsobom je zabezpečenie pomocou hesla (niečo, čo poznám) alebo prístupových kariet (niečo, čo mám). Heslo však môže človek zabudnúť a prístupová karta môže byť ukradnutá, prípadne stratená. Tieto problémy rieši biometrické rozpoznávanie, skrátene biometria, pretože využíva k identifikácií prvky, ktoré dotyčná osoba nemôže stratiť a sú len veľmi ťažko zneužiteľné. [20]

Biometria (*z grec. slova bios = život a metron = merať*) využíva jednu zo základných metód identifikácie – niečo, čím užívateľ je. Využívajú sa pritom rozlíšiteľné anatomické a behaviorálne vlastnosti jedinca. Tieto vlastnosti sa nazývajú biometrické identifikátory. Ich hlavnou výhodou je, že ich nie je možné stratiť, zabudnúť, či len tak zneužiť ďalšou osobou. Zvlášť v kontexte tejto práce je potrebné pripomenúť, že niektoré biometrické identifikátory môžu prezrádzať veľa o zdravotnom stave osoby. Pred využitím akéhokoľvek identifikátoru je nutné overiť, či identifikátor spĺňa týchto 8 základných vlastností [10] [18] [20]:

1. Univerzálnosť – každý by mal mať daný identifikátor.
2. Unikátnosť – žiadne dve osoby by nemali mať daný identifikátor rovnaký.
3. Trvalosť – identifikátor by sa nemal meniť počas života.
4. Stálosť – identifikátor by nemal degradovať.
5. Prijateľnosť – identifikátor by mal byť prijatý v spoločnosti.
6. Falšovateľnosť – aké ťažké je identifikátor sfalšovať.
7. Cena – Koľko stojí biometrický systém.
8. Prevádzka – aká je náročná prevádzka pre daný biometricky systém.

V biometrií existuje niekoľko dôležitých konceptov. Jedným z nich je inter a intra triedna premenlivosť. Inter triedna premenlivosť ukazuje, aký veľký je rozdiel vrámci identifikátov medzi rozdielnymi triedami (osobami). Intra triedna premenlivosť nám udáva, aký veľký rozdiel je medzi identifikátormi v jednej triede (jedinec). [18]

3.1.1 Odtlačok prsta v biometrií

Odtlačky prstov sú jednou z najstarších biometrických identifikátorov, ktoré ľudstvo používa už viac než 3000 rokov podľa zápisov v dochovaných čínskych textoch [30]. Mnoho ľudí je presvedčených, že odtlačky prstov sú unikátne, no v skutočnosti sa nejedná o fakt, ale len o výsledky empirického pozorovania za niekoľko storočí. Túto skutočnosť prvý krát popísal sir Francis Galton, ktorý spočítal, že pravdepodobnosť výskytu 2 rovnakých odtlačkov prstov je 1 ku 64 miliardám. [16]

Odtlačky prsta sú jedinečnou črtou pokožky vďaka unikátnemu usporiadaniu papilárnych línií. Hlavnou výhodou používania odtlačkov prstov pre identifikáciu je ich unikátnosť, stálosť a cena. Toto dokazuje aj ich široké využitie v praxi, konkrétne rozšírené používanie v kriminalistike, pri zabezpečení budov či pri odomykaní mobilných zariadení. Aj vďaka stále väčšiemu využitiu odtlačkov prstov v bežnom živote rastie akceptovateľnosť ich používania, problémom však stále zostáva tvorba databáz a nedôvera ľudí pri ukladaní ich biometrických údajov. [20] [18]

3.1.2 Rozdelenie odtlačkov prstov

Porovnávanie odtlačkov prstov nie je tak jednoduchá úloha, ako by sa mohlo zdať. Preto sú odtlačky rozdelené do niekoľkých tried podľa spoločných znakov. Porovnanie, či daný odtlačok prsta obsahuje daný znak, nám dáva efektívny nástroj pre rýchle prehľadávanie databáz, nakoľko nie je potrebné prehľadávať celú databázu. [20]

Papilárne línie sú uložené vedľa seba, no občas vytvárajú oblasti, kde končia, alebo sa stáčajú. Tieto oblasti voláme singulárne body a rozdeľujeme ich na tri typy: *slučka, delta a vír*. Niektoré zdroje typ *vír* neuvádzajú keďže je ho možné popísať ako dve slučky otočené proti sebe.

Pomocou singulárnych bodov môžme popísať 4 triedy, ktoré slúžia ako základ pre triedenie odtlačkov prstov. Týmito triedami sú: *slučka, vír, oblúk a klenutý oblúk*. Toto rozdelenie tiež nazývame Galton-Henryho rozdelením. [20]

Ďalšími markantmi, ktoré nám pomáhajú pri identifikácií, sú *Galtonové detaily* [20], pomenované na počesť sira Francisa Galtona. Galton popísal viac ako 100 týchto markantov. Najčastejšími sú však koniec línie, rozdvojenie, oko, interval, bod, ostroha a prekríženie. Tieto znaky sa štatisticky vyskytujú v priemere 0,49 $znakov/mm^2$ v singulárnych častiach a 0,18 $znakov/mm^2$ mimo nich. [20]

Na lokálnej úrovni môžeme sledovať drobné detaily ako sú rozmery papilárnych línií, jednotlivé rozloženia pórov, drobné poškodenia a prerušenia. Každá línia je pokrytá po celej svojej dĺžke pórmi. Tieto póry dosahujú veľkosť $60 - 250mm$. Už pri zistení pozície 20–40 pórov by bolo možné identifikovať osobu. Problémom je však nutnosť použitia skeneru, ktorý by dosahoval aspoň 1000 dpi (počet bodov na palec z angl. *dots per inch*), aby boli tieto vrstvy viditeľné. [20]

3.2 Spracovanie odtlačkov prstov

Schopnosť biometrických systémov rozpoznať dané vzory je často závislá od vstupných dát. Inak tomu nie je ani pri odtlačkoch prstov. Vstupné obrázky sú často priemernej alebo slabšej kvality. Na kvalitu obrázka vplýva množstvo faktorov, najčastejšími sú zvýšený tlak na plochu senzoru, drobné zárezy či vplyv vlhkosti. Častými poškodeniami odtlačkov sú [20]:

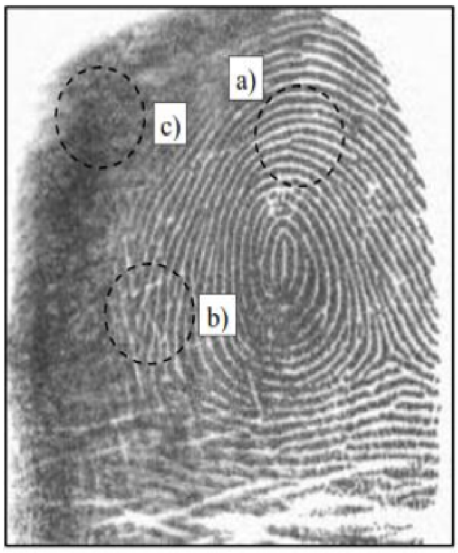
(a) Necelistvé línie, ktoré obsahujú drobné prerušenia,

(b) paralelne prebiehajúce línie ktoré nie sú zjavne oddelené,

(c) rôzne druhy zárezov a zošúchaných línií.

Tieto poškodenia spôsobujú problémy pri hľadaní markantov. Niektoré markanty nemusia byť kvôli oblasti s vyšším poškodením vôbec zaznamenané, prípadne môže byť zle zaznamenaná ich orientácia alebo pozícia. Tiež môže dôjsť k falošnému zaznamenaniu markantov. Odtlačky však často nie sú poškodené celé a zasiahnuté sú len niektoré oblasti (viď obrázok 3.1). Tieto oblasti sa rozdeľujú do troch kategórií [20]:

- Oblasti bez poškodenia – línie sú dobre viditeľné a sú zjavne oddelené. Markanty sú jasne viditeľné. [20]
- Obnoviteľné oblasti – línie síce obsahujú prerušenia, no oblasť okolo nich nám dobre ukazuje skutočný smer. [20]
- Neobnoviteľné oblasti – línie nie sú vôbec viditeľné kvôli šumu a množstvu poškodení. Rovnako nám ani ich okolie neposkytuje dostatok informácií pre obnovenie. [20]



Obr. 3.1: Odtlačok prsta s rôznymi oblasťami kvality. a) oblasť bez poškodenia b) obnoviteľná oblasť c) neobnoviteľná oblasť. Zdroj: [20]

3.2.1 Predspracovanie

Predspracovanie je dôležitou časťou pri práci s odtlačkami. Poznáme niekoľko spôsobov, ako odtlačky predspracovávať a ako ich vylepšovať. V tejto podkapitole si v krátkosti predstavíme bežne techniky predspracovania obrazu. V ďalšej časti si popíšeme algoritmy, ktoré boli použité v práci.

Adaptívne prahovanie

Nasledujúca podkapitola vychádza z [25] a [9]. Segmentácia prahovaním je jedna z najpoužívanejších a najjednoduchších metód pre spracovanie obrazu. Vychádza z predpokladu, že hľadané objekty majú iný jas ako pozadie na ktorom sa nachádzajú. Prahovanie je skvelým algoritmom, pokiaľ sa objekty na obrázku nedotýkajú. Jej hlavnou výhodou je časová a pamäťová nenáročnosť.

Prahovanie je transformácia vstupného obrazu f na výstupný binarizovaný obraz g. Túto transformáciu môžeme popísať ako:

$$g(i,j) = \begin{cases} 1 & pre \quad f(i,j) \geq T \\ 0 & pre \quad f(i,j) < T \end{cases} \tag{3.1}$$

Pričom platí, že T je hodnota prahu, $g(i,j) = 0$ sú oblasti pozadia, $g(i,j) = 1$ sú oblasti objektov. Keď sme popísali ako funguje základné prahovanie, môžme popísať to adaptívne, ktoré nachádza využitie pri nerovnomerne osvetlených oblastiach na obrázku, ako sú tiene, či príliš silné osvetlenie. V prípade odtlačkov prstov to môžu byť oblasti so zvýšeným prítlakom. Oproti bežnému prahovaniu sa pri adaptívnom mení hodnota prahu počas prechádzania obrazom. V zásade existuje niekoľko spôsobov, ako to dosiahnuť. Prvým spôsobom je modelovanie pozadia v obraze popísanej v [8]. Tým druhým je rátanie prahu len z malého okolia aktuálne rátaného pixelu. Posledným je rozdelenie obrazu na menšie podobrázky a vypočítanie prahu pre každý z nich. Je zrejmé, že posledná metóda naráža na problém, že v každom segmente môže byť vyrátaný iný prah, čo môže robiť problémy na hraniciach týchto segmentov. Rozdiely v adaptívnom a bežnom prahovaní si môžme všimnúť na obrázku 3.2, kde si adaptívne prahovanie oveľa lepšie poradilo s tmavými oblasťami, ako to bežné.

Obr. 3.2: Ukážka rozdielov pri použití bežného a adaptívneho prahovania (zľava doprava): originálny obrázok, klasické prahovanie, adaptívne prahovanie.

Hľadanie obrysov odtlačku

Ďalším algoritmom,ktorý bol využitý pri predspracovaní v práci, je algoritmus pre vyhľadávanie obrysov v odtlačku vytvorený S. Suzuki a K. Abe. Popis tohto algoritmu bude vychádzať z [26].

Vstupom funkcie je binarizovaný obrázok. Algoritmus začne prechádzať obrázkom pokiaľ nenarazí na pixel objektu, ktorý má hodnotu 1. Tento bod je počiatočným bodom hranice. Tejto hranici priradíme jednoznačné identifikačné číslo hranice. Pre zistenie či novovzniknutá hranica má rodiča postupujeme nasledovne. Počas prechádzania obrázkom si v pamäti udržujeme aj čislo poslednej vonkajšej hranice oblasti pripadne diery, na ktorú sme narazili naposledy. Táto hranica by mala byť rodičovská hranica novej hranice alebo by mala byť hranicou, ktorá je rovnako postavená k novonájdenej a má rovnakého rodiča.

Po najdení prvého bodu pokračuje v sledovaní okraja. Týmto pixelom je priraďované číslo objektu. Charakteristickou črtou tohto algoritmu sú dve podmienky.

1. Ak je aktuálne sledovaná hranica na rozhraní 1 0 potom zmeň označovanie pixelov na predchádzajúce číslo.

2. Inak, nastav hodnotu pixelu (p,q) na aktuálny identifikátor objektu pokiaľ nieje už tato hodnota nebola nastavená.

Podmienky 1 a 2 zabraňujú, aby sa pixel (p,q), ktorý už bol označený ako súčasť istého objektu, stal súčasťou objektu iného. Po označení kompletne celej hranice začne algoritmus s prechádzaním obrázku, až pokiaľ nedosiahne spodný pravý okraj.

Zlepšovanie kvality odtlačkov prstov

Po odstránení pozadia je ďalším krokom odstránenie šumu a iných rušivých elementov. Toto zlepšenie môžeme dosiahnuť aplikovaním rôznych filtrov. Hlavnou úlohou tohto kroku je teda zlepšenie čitateľnosti jednotlivých papilárnych línií. Medzi najčastejšie používané patria rôzne druhy filtrovania, ako sú kontextuálne či viacrozmerné filtrovanie. Ďalšími zaujímavými technikami sú pixel-wise techniky, kde nová hodnota pixelu nezávisí od okolitých pixelov, ako sú úpravy histogramu, čí kontrast streching, ktoré sú efektívnym počiatočným krokom pri zložitejších metódach. Ako posledné je dobre sa zmieniť o technikách pre odstraňovanie vrások. Tieto techniky sú schopné zoceliť menšie vrásky, ktoré by inak viedli k falošnému rozpoznaniu pri výpočte smerových hodnôt. [20]

Kapitola 4

Klasifikácia pomocou neurónových sietí

Neurónová sieť je abstraktný počítačový model ľudského mozgu. V ľudskom mozgu je okolo 10^{11} neurónov, ktoré tvoria odhadom 10^{15} prepojení. Táto biologická neurónová sieť je považovaná za zdroj inteligencie, čo zahŕňa poznávanie, vnímanie a učenie. Keď neurónovú sieť zobrazíme ako graf, neuróny môžeme prezentovať ako uzly a prepojenia neurónov ako hrany. [21]

V tejto kapitole si popíšeme čo je to základná jednotka neurón a najjednoduchšia forma neurónovej siete perceptrón. Ďalej v podkapitole 4.2 si predstavíme najpoužívanejšie aktivačné funkcie. V ďalšej podkapitole 4.3 bude popísaná loss funkcia, ktorá popisuje úspešnosť siete a je dôležitá pri učení. Následovať budú optimalizátory neuronových sieti v časti 4.4. Po prejdení týchto pojmov budú nasledovať viacvrstvové a konvolučné siete popísane v podkapitolách 4.5 a 4.6. Poslednými podkapitolami bude učenie neurónových sieti v časti 4.8. A na zaver kapitoly si predstavíme model VGG16 z ktorého bude naša sieť vychádzať.

4.1 Popis neurónu a perceptrónu

Neurón je základnou jednotkou neurónovej siete. Každý neurón má vstupy $x_0, ..., x_m$. Tieto vstupy sú abstrakciou prepojení v skutočnom neuróne. Každý tento vstup je potom vynásobený váhou w_i. Tento váhový koeficient je obdobou skutočnej hrúbky biologických synapsií (prepojení) medzi jednotlivými neurónmi. Vystúp operácie $x_i w_i$ je potom ďalej použitý v tele neurónu. V tele neurónu sa potom výsledky násobení jednotlivých koeficientov sčítajú. V matematickom zápise je možné potom zapísať toto násobenie jednotlivých koeficientov ako skalárny súčin dvoch vektorov $x = (x_0, x_1, ..., x_m)$ a $w = (w_0, w_1, ..., w_m)$. Výsledok podľa bežne používanej notácie nazveme net (viď rovnica 4.1). [14] [21]

$$net = w \cdot x \tag{4.1}$$

K výsledku tejto operácie net sa pripočíta bias b. Bias zvyšuje alebo znižuje vstup aktivačnej funkcie v závislosti od jeho hodnoty. Popis aktivačnej funkcie je popísaný v nasledujúcej podkapitole 4.2. Samotný neurón môžeme teda popísať ako (viď rovnica 4.2) výsledok aktivačnej funkcie po dosadení skalárneho súčinu net v súčine s použitým biasom b [14]:

$$y = \varphi(net + b) \tag{4.2}$$

Perceptrón je najjednoduchšou formou neurónovej siete, používa sa pre klasifikáciu vzorov, ktoré sú lineárne odlíšiteľné. V základe pozostáva z jedného neurónu s nastaviteľnými váhami a biasom. Perceptrón s jedným neurónom je limitovaný na klasifikovanie iba do dvoch tried (hypotéz). Príkladom môže byť Rosenblattov perceptrón, ktorý je vytvorený z jedného neurónu. V podkapitole 4.1 je uvedené, že takýto neurón pozostáva zo skalárneho súčinu a aktivačnej funkcie zastúpeného aktivačnou funkciou. Cieľom perceptrónu je teda správne zaradenie vonkajších vstupov stimulátorov $x_0, x_1, ..., x_n$ do jednej z dvoch tried C_1 alebo C_2. [14]

4.2 Aktivačná funkcia

Aktivačná funkcia, označená ako $\varphi(v)$, definuje vystúp neurónu. Existuje niekoľko typov aktivačných funkcií, no najčastejšie používanými sú podľa [14]:

- **Prahová funkcia** – často aj označovaná ako *Heavisidová funkcia*, určuje výslednú hodnotu ako:

$$\varphi(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \tag{4.3}$$

- **Sigmoid** – je najčastejšie používanou aktivačnou funkciou pre binárne problémy. Je definovaný ako neklesajúca funkcia, ktorá vykazuje dobrú rovnováhu medzi lineárnym a nelineárnym správaním. Príkladom sigmoidu môže byť *logistická funkcia* definovaná ako:

$$\varphi(v) = \frac{1}{1 + exp(-av)} \tag{4.4}$$

- **Softmax** – časté využitie nachádza v problémoch, kde je potrebné klasifikovať viac tried. Každej triede na výstupe priradí pravdepodobnosť, že sa jedná o danú triedu. Súčet týchto pravdepodobností bude 1. Pre určenie triedy sa nakoniec vyberie tá s najvyššou pravdepodobnosťou. Matematicky môžme túto funkciu zapísať ako:

$$\sigma(z)_j = \frac{e^{Z_j}}{\sum_{k=1}^{K} e^{Z_k}} \tag{4.5}$$

- **ReLU** – jedná sa o najčastejšie používanú aktivačnú funkciu pre neuronové siete špeciálne pre tie konvolučné. Ako výstup funkcie vracia vstup pokiaľ je väčšiu rovný 0. Pre záporne hodnoty vracia nulu. Jej funkciu môžme zapísať ako:

$$f(x) = max(0, x) \tag{4.6}$$

  Pričom jej hlavnou výhodou je jednoduchosť. Využíva pomerne jednoduchú matematiku čo ju robí nenáročnou na výpočet a tým aj urýchľuje samotne učenie. [4]

4.3 Loss funkcia

Táto podkapitola vychádza zo zdroja [5]. Stratová funkcia anglicky *loss function* je v kontexte optimalizácie funkcia, ktorá popisuje vzdialenosť medzi aktuálnym odhadom siete a skutočným riešením. V prípade strojového učenia je snaha nájsť jej lokálne minimum. Dôležitou úlohou stratovej funkcie je verná interpretácia všetkých vstupných parametrov, tak aby zlepšenie hodnoty na výstupe tejto funkcie odrážalo reálne zlepšenie modelu. V jednoduchosti výsledná hodnota stratovej funkcie odráža jedným číslom všetky parametre siete. Dôležité je vybrať správnu funkciu tak, aby odrážala riešenie nášho problému. V prípade zle zvolenej funkcie, môže byť aj skvelý model nepoužiteľný. V prípade loss funkcií rozdeľujeme problém klasifikácie na dva druhy.

- **Binárna klasifikácia** - ako už názov napovedá, sieť má za úlohu zaradiť vzorku do jednej z dvoch tried. Úlohou siete je teda určiť pravdepodobnosť, že daná vzorka patri do prvej triedy. Preto prvej triede priradíme celé číslo 1, zatiaľ čo druhej priradíme 0. V tomto prípade je vhodné použiť na výstupnom uzle ako aktivačnú funkciu sigmoid. A ako loss funkciu využiť binárnu krížovú entropiu.
- **Multi-triedna klasifikácia** - sieť musí vzorku zaradiť do jednej z viac ako dvoch tried. V tomto prípade už sa nezaobídeme len s 1 a 0, preto aktivačná funkcia musí byť schopná na výstupe dávať viacero hodnôt, preto využijeme funkciu softmax. Ako loss funkciu využijeme kategorickú krížovú entropiu.

Keď sme si predstavili problem klasifikacie uvdedieme si aj využívane loss funkcie ktore riešia tieto druhy problemov. Samozrejme existuje viacero loss funkcií tu uvadzame tie ktore boli využite v praci.

- **Binárna krížova entropia** meria, ako ďaleko od skutočnej hodnoty (ktorá je buď 0 alebo 1) je predpoveď pre každú z tried, a potom priemeruje tieto chyby podľa tried, aby sa získala konečnú hodnoty straty. [5]
- **Kategorická krížova entropia** porovnáva vektor predpovedí (výstupy aktivačných funkcií jedna pravdepodobnosť na každu triedu) so správnym výstupom, pričom pravdepodobnosti v spravnom výstupe sú uložené vo vektore pričom len pravdepodobnosť spravnej triedy je nastavená na 1 a pre ostatne triedy 0. Platí že čím bližšie su hodnoty vo vystupnom vektore k tomuto vystupnemu vektoru tym nižšia je hodnotda loss funkcie. [5]

4.4 Optimalizátory

Cieľom optimalizácie je nájdenie takých hodnôt, pri ktorých loss funkcia nadobúda najnižšiu pripadne najvyššiu hodnotu. V prípade učenia neurónových sieti hľadáme minimum účelovej funkcie, ktoré je rovné riešeniu problému. Zjednodušene optimalizátor upravuje váhy modelu, aby dosiahol optimálne riešenie problému.

Optimalizátory teda určujú rýchlosť učenia siete a presnosť výsledkov. Vyber optimalizátora je dôležitou súčasťou tvorby siete. V nasledujúcej podkapitole si preto predstavíme niekoľko z nich.

4.4.1 Gradient descent

Gradientny zostup (z angl. gradient descent) je základným optimalizačným algoritmom pre nájdenie minima na danom intervale funkcie. Funkcia je založená na pozorovaní, ze ak viacvstupná funkcia je definovaná v susedstve bodu w a je diferencovateľná. Potom funkcia $F(x)$ klesá najrýchlejšie smerom, ktorým je pre w gradient najnižší. Potom môžme napísať že:

$$w_{n+1} = w_n - \gamma \nabla F(w_n) \tag{4.7}$$

Pričom platí, že w_n je aktuálna pozícia a γ je veľkosť kroku a $\nabla F(w_n)$ je smer s najnižším gradientom. Pri ďalších krokoch nám vznikne sekvencia o ktorej platí, že klesá. Týmito postupnými krokmi dosiahneme lokálne minimum. Hlavnou nevýhodou je pomerne pomalý postup, nakoľko pre každú aktualizáciu váh musíme spočítať gradient pre celý súbor dát, čo je náročné nielen pre výpočtovú ale aj pamäťovú zložitosť. [3] [23]

4.4.2 Stochastic gradient descent

V skratke SGD je iteratívny optimalizačný algoritmus vychádzajúci z gradientneho zostupu predstavenom v predchádzajúcej podkapitole. Patrí medzi najpoužívanejšie. Jeho hlavnou výhodou je lepšia konvergencia k minimu. Namiesto reálneho gradientu ktorý je vypočítaný z celého súboru dát, stochastický odhadne ďalšiu optimalizáciu tým, že počíta gradient pre každý aktuálny prvok. Táto upráva má veľký význam pri väčších súboroch dát, kedy je ušetrený pomerne veľký výkon za cenu mierne nižšej konvergencie. Ďalšou výhodou je samotná rýchlosť učenia. Na druhú stranu nevýhodou môže byť veľká fluktuácia pri konvergencii, nakoľko výpočet gradientu závisí len od aktuálneho prvku.[23]

Tento problém však rieši vylepšenie v podobe mini-batch gradient descent, ktorý oproti klasickému SGD počíta gradient na malej podmnožine. Hlavnými výhodami je práve menšia fluktuácia konvergencie a skutočnosť, že pracujeme s maticou čo môže byť veľmi dobre optimalizovatelné pre grafické karty. [23]

4.5 Viacvrstvové neuronové siete

V predchádzajúcej kapitole 4.1 bola predstavená jednovrstvová neurónová sieť. Rovnako sme si popísali aj jej obmedzenia klasifikácie. V tejto kapitole si popíšeme viacvrstvové neuronové siete (ďalej len siete). Ich základnou vlastnosťou je, ako už názov napovedá, viacero vrstiev. Vrstvy môžno rozdeliť na vstupno-výstupné a skryté vrstvy. Vstupno-výstupné vrstvy prepájajú skryté vrstvy so vstupno-výstupnými uzlami. Skryté vrstvy sú oddelené od vstupno-výstupných uzlov. [14]

Aktivačná funkcia každého neurónu musí byť diferencovateľná. Táto vlastnosť je dôležitá kvôli stratégií spätnej propagácie (backpropogation optimization strategy) v sieti pri výpočte gradientu chyby s ohľadom na nastavené váhy [14] [28]. Tieto siete sa vyznačujú tiež vysokou konektivitou, ktorá je určená synaptickými váhami siete [14].

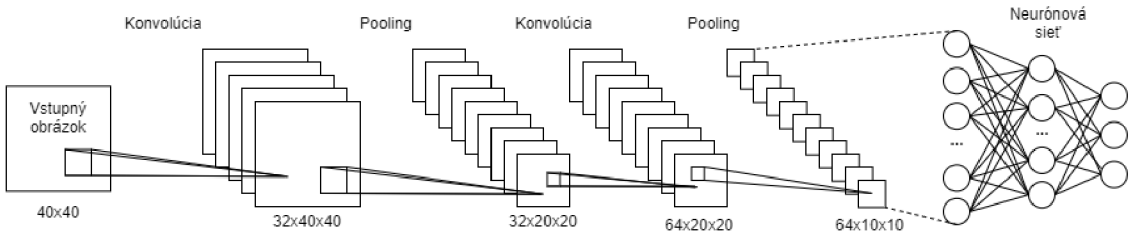
Charakteristiky spomenuté vyššie sú tiež dôvodom nedostatkov v našich vedomostiach o správaní týchto sietí. Využívanie skrytých vrstiev nám zhoršuje vizualizáciu učiaceho procesu siete. Učiaci proces určuje, ktoré znaky zo vstupných uzlov majú byť skrytými vrstvami využívané. Zložitosť pochopenia procesu učenia spôsobuje aj vysoká konektivita neurónov. Táto konektivita zväčšuje priestor, v ktorom vyhľadávame funkciu. Táto funkcia potom znázorňuje vstupný vzor.

4.6 Konvolučné neuronové siete

Konvolučné siete (v skratke CNN - z angl. Convolutional Neural Network) sú špeciálne typy neurónových sietí určených pre spracovávanie dát usporiadaných v 2D mriežke. Názov konvolučná neurónová sieť je odvodená od používanej matematickej operácie konvolúcie. Tento typ sietí má vysoké uplatnenie v rozpoznávaní objektov a textu, či pri spracovaní prirodzeného jazyka. Jednou z veľkých výhod týchto sietí sú menšie požiadavky na preprocesing oproti bežným neurnovým sieťam. [13]

Pri bežných neurónových sieťach sa využíva plne prepojená prvá vrstva. Pri menšom množstve dát to nie je problém, no pri veľkých obrázkoch by bežná neurónová sieť s niekoľkými desiatkami či stovkami neurónov v skrytej vrstve vyžadovala obrovské množstvo prepojení, čo podstatne navyšuje pamäťovú zložitosť siete. Táto skutočnosť dosť obmedzuje využiteľnosť týchto sieti v zariadeniach s nižšou kapacitou pamäte a výkonu. Rovnako pri takto veľkej sieti je potrebné obrovské množstvo trénovacích dát kvôli pokrytiu všetkých možných stavov. Hlavnou nevýhodou bežných sieti pri spracovávaní obrazu je ich vysoká závislosť na vstupoch, preto nie sú odolné voči šumu a posunom (napr. rukou písané číslo má rôzne sklony). [19]

Konvolučné neuronové siete kombinujú 3 návrhové myšlienky, ktoré zabezpečujú nemennosť voči posunom, zmene veľkosti a šumu: *zdieľanie váh, podvzorkovanie obrazu a získavanie príznakov z recepčných polí.* Využitie recepčných polí nám umožňuje získavať základné časti ako je veľkosť uhlov a ich orientácia, rohy, konce... Tieto časti sú potom v ďalších vrstvách skladané do väčších znakov. Ak bude základný vzor (napríklad hrana) detegovaný v jednej časti obrázku, s veľkou pravdepodobnosťou bude tento vzor detegovaný aj v inej časti. Táto skutočnosť nám umožňuje využívať rovnaké váhy na rôznych častiach obrázku. Neuróny sú usporiadané do rovín a všetky zdieľajú rovnakú množinu váh. To znamená, že každý z nich vykonáva tú istú operáciu nad inou časťou obrázku. Výstupy týchto neurónov sú organizované do *máp vlastností.* Samotná vrstva konvolučnej siete sa skladá z viacerých týchto máp, ktoré vznikajú použitím rôznych váh. [19]



Obr. 4.1: Architektúra konvolučnej neurónovej siete.

Po nájdení vlastnosti sa samotné umiestnenie stáva nepodstatným, nakoľko sa toto umiestnenie bude líšiť v rôznych prípadoch. Dôležitá je ale poloha voči ostatným nájdeným vlastnostiam nie poloha samotná (pretože vlastnosť nebude vždy na rovnakých pozíciach). Spôsob ako zvýšiť dôležitosť pozícií medzi vlastnosťami, je zmenšenie rozlíšenia (sub-sampling) a lokálne spriemerovanie danej mapy. Spojenie týchto operácií nazývame „pooling“. Touto operáciou sa znižuje vplyv posunov a šumu na výstup. Podľa obrázku 4.1 je možné vidieť, že prvá poolingová vrstva zmenšuje veľkosť tým, že zoberie maticu z predchádzajúcej mapy o veľkosti 2x2 a pomocou predurčených váh vypočíta novú hodnotu,

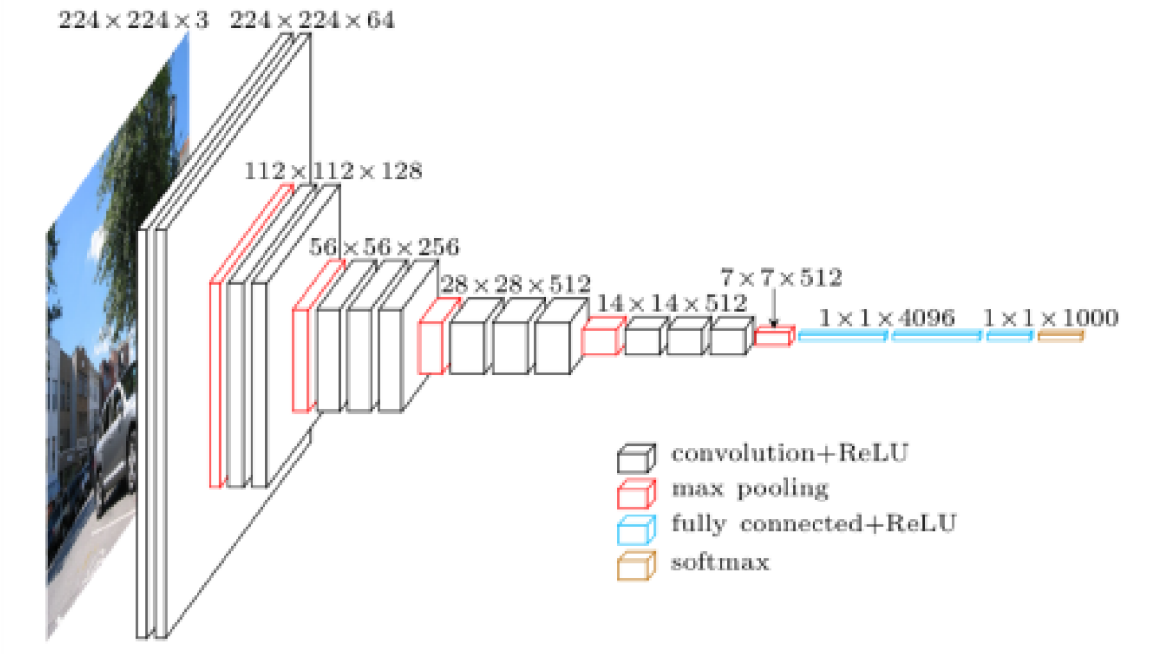
ktorá bude vložená do matice. Výsledná matica je potom štvrtinová oproti pôvodnej matici. Nasledujúca konvolučná vrstva využíva mapy vlastností tej predchádzajúcej. Po dosiahnutí dostatočného zmenšenia mapy je táto mapa plne prepojená na neurónovú sieť a tvorí jej vstupnú vrstvu. Takejto vrstve hovoríme plne prepojená vrstva, nakoľko je prepojená v pomere 1:1. Toto prepojenie sa uskutočňuje rozdelením obrázka na 1D vektor, ktorý je vložený do neurónovej siete. Táto sieť potom určí výsledok. [19]

4.7 Model VGG16

Podkapitola popisuje model VGG16 a čerpá z [24]. Model siete bol vytvorený K. Simonyan a A. Zisserman na Oxfordskej univerzite. V roku 2014 vyhral súťaž ILSVR (Imagenet) a dodnes je považovaný za jeden z najlepších modelov pre počítačové videnie. Výnimočnosť architektúry spočíva v použití konvolučných vrstiev s malými recepčnými poliami namiesto využívania veľkého množstva hyperparametrov.

Vstupom siete je farebný obrázok o veľkosti 224×224 pixelov. Architektúra modelu pozostáva z niekoľkých konvolučných vrstiev, ktoré využívajú malé receptívne polia a to o veľkosti 3x3. Táto veľkosť receptívneho poľa je najmenšia, ktorá je ešte schopná zachytiť postup z celého okolia pixelu. Krok konvolúcie je vždy zachovaný na jeden pixel.

Tieto konvolučné vrstvy môžu byť zaradené aj viackrát za sebou. Vrstvy obvykle ako svoju aktivačnú funkciu využívajú retifikované lineárne jednotky v skratke *ReLu*, predstavenej v podkapitole 4.2. Ďalšou vrstvou je max-pooling, ktorá slúži pre podzvorkovanie vstupu. Max-pooling sa vykonáva v okne o veľkosti 2×2 s krokom 2. Výsledkom max-pooling je o polovicu menšia matica. Takto zoradené vrstvy sú potom v „balíkoch“ niekoľko krát naskladané za seba viď obrázok 4.2.



Obr. 4.2: Architektúra modelu VGG16. Zdroj: [22].

Po konvolučných vrstvách nasledujú dve plne prepojené vrstvy, ktoré využívajú tak ako predchádzajúce konvolučné vrstvy aktivačnú funkciu ReLu. Každá z týchto vrstiev obsahuje 4096 kanálov. Tieto vrstvy slúžia ako vyhodnocovacia časť CNN. Nasleduje výstupná vrstva, ktorá využíva aktivačnú funkciu *softmax*. Výstupná vrstva obsahuje taký počet kanálov, koľko je hľadaných tried. Originálny dokument uvádza 1000 výstupov.

4.8 Učenie konvolučných sietí

Jednou z najpoužívanejších metód pre učenie neurónových sietí je *metóda spätného šírenia*. V učiacom procese znižuje nepresnosti, využíva pritom algoritmus *gradient descent*, ktorý upravuje jednotlivé váhy prepojení. Tieto váhy sú závislé na výpočte funkcie chyby (anglicky loss function). Táto funkcia opakovane počíta rozdiel medzi očakávaným a skutočným výstupom. Výstup tejto funkcie je potom pomocou algoritmu spätného šírenia prenesený do vyšších vrstiev. [19]

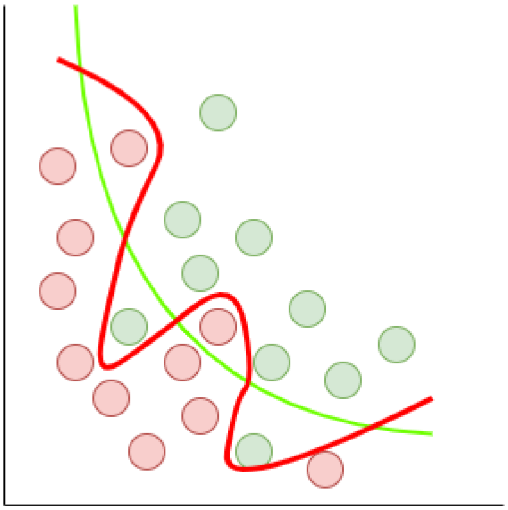
Na začiatku táto metóda bola používaná len na lineárne systémy. Až niekoľko udalostí ukázalo, že táto metóda je schopná riešiť aj zložitejšie problémy. Prvou touto udalosťou bolo zistenie, že prítomnosť lokálnych miním nemá v praxi väčší vplyv na úspech. Popularizáciu tejto metódy spôsobili aj jednoduché a efektívne postupy pre algoritmus spätnej propagácie na výpočet gradientu vo viacvrstvových nelineárnych systémoch. Treťou udalosťou, ktorá zapríčinila obrovskú rozšírenosť, bolo zistenie, že spätná propagácia aplikovaná na niekoľko vrstvovú neurónovú sieť s sigmoidovými jednotkami dokáže riešiť aj náročnejšie úlohy. Hlavnou myšlienkou tohto prístupu je fakt, že je oveľa jednoduchšie minimalizovať vplyv prebiehajúcej funkcie ako diskrétnu časť s výraznejšími hodnotami. [19]

Metóda spätnej propagácie našla uplatnenie v množstve smerov, za zmienku stojí hlavne rozpoznávanie vzorov. Základnou myšlienkou tejto metódy je, že gradient môže byť efektívne vypočítaný z výstupu až na vstup. Táto metóda môže byť pomerne náročná na veľkých sieťach, nakoľko sa musí počítať veľké množstvo premenných. [19] [6]

4.8.1 Pretrénovanie siete

Pretrénovane siete je stav kedy sieť vyberá detaily a šum z trenovacích dát ako koncept riešenia daného problému. To potom negatívne ovplyvňuje schopnosť modelu vyhľadávať všeobecné riešenie v nových dátach. Takýto model síce dosahuje vysokú presnosť na trénovacích dátach, no v prípade reálneho nasadenia je tento model nepoužiteľný.

Pre zlepšenie pochopenia problému pretrénovane môžme využiť obrázok 4.3. Môžme vidieť binárny problém a sieť ma rozhodnúť či bod patrí alebo nepatrí do množiny. Zelená čiara reprezentuje správne natrénovaný model. Červená reprezentuje pretrénovaný model kedy sieť už vyberá aj vzorky ktoré nespadajú do riešenia a tým sa znižuje úspešnosť siete na nových dátach. [1]



Obr. 4.3: Graf pretrénovania

Pretrénovane je síce častým problémom no sú techniky ako znížiť jeho pravdepodobnosť, že k nemu dôjde. Prvým spôsobom je krížová validácia. Umožňuje nám trénovať a testovať model na rôznych podmnožinách vytvorených z trénovacej množiny a tým zostaviť odhad výkonu modelu. Druhou možnosťou je vytvorenie validačného datasetu, ktorý je vytvorený z trénovacej množiny a sieti počas trénovania nieje sieti dostupný. Po dotrénovaní siete je táto podmnožina využitá na validáciu skutočnej presnosti modelu. Čím dostávame hodnoty podobne tým, ktoré by sme dostavali na reálnych dátach. [1]

Kapitola 5

Návrh riešenia

V práci budú sledovanými ochoreniami dyshidróza a bradavice. Ich prejavy sme si predstavili v podkapitole 2.3. Tieto ochorenia budú detegované a klasifikované konvolučnou neurnovou sieťou. Cieľom práce je teda vytvorenie klasifikátoru a detektoru, ktorý bude, najlepšie bez zásahu užívateľa, schopný: a) detegovať poškodenia spôsobené sledovanými ochoreniami na odtlačku prsta, b) klasifikovať tieto poškodenie s čo najvyššou presnosťou a priradiť ich jednotlivým ochoreniam, c) vrátiť užívateľovi oblasť, na ktorej sa ochorenie nachádza.

Po stanovení týchto cieľov, bude samotné riešenie práce pozostávať z nasledujúcich krokov:

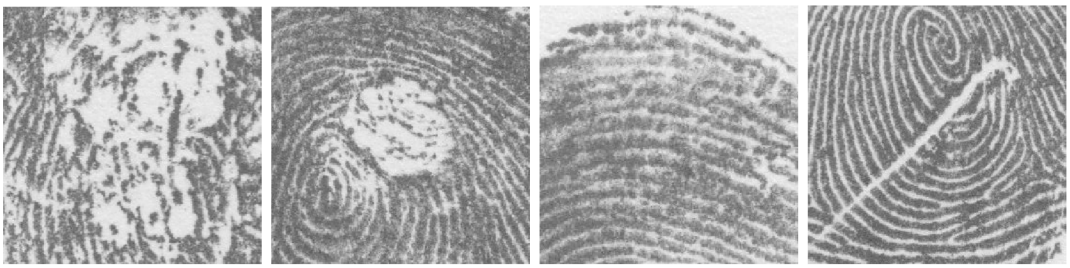
- Zostavenie databázy, ktorá bude dostatočne rozsiahla pre trénovanie siete.
- Vytvorenie sady funkcií na prípravu datasetu. Tieto funkcie budú pripravovať obrázky z databázy pre vkladanie do siete.
- Navrhnutie a vytvorenie modelu konvolučnej siete. Jej model bude vychádzať z už predstaveného modelu popísaného v podkapitole 4.7.
- Vytvorenie samotného detektoru, ktorý bude za pomoci neurónovej siete z predchádzajúceho bodu detegovať a klasifikovať ochorenia.

V nasledujúcich podkapitolách si predstavíme jednotlivé návrhy riešení. Postupne ako boli predstavené. V podkapitole 5.1 si predstavíme návrh databázy odtlačkov a popíšeme si čo by mala obsahovať. Nasledovať bude návrh funkcií pre spracovanie odtlačkov a ich prípravy pre neurónovú sieť v podkapitole 5.2. Nasledovať bude návrh samotnej siete (podkapitola 5.3) a jej trénovanie (podkapitola 5.4). Nakoniec kapitoly si v podkapitole 5.5 predstavíme tvorbu výstupu.

5.1 Zostavenie trénovacej množiny dát

Dáta do vstupnej databázy budú získavané z viacerých zdrojov. Najväčšiu časť budú tvoriť syntetické odtlačky prstov, do ktorých budú vygenerované prejavy sledovaných ochorení. Ďalším prípadným zdrojom dát môžu byť databázy s reálnymi prejavmi ochorenia. Týchto dát nieje dostatok, aby pokryli celý dataset. Na druhej strane ich prítomnosť v ňom môže výrazne napomôcť optimálnemu výsledku. Poslednou niemenej dôležitou súčasťou budú odtlačky bez prejavov ochorenia. Tie sú potrebné pre doladenie oblasti, kde sa hľadané ochorenia nenachádzajú, ale môžu byť poškodené bežnými vplyvmi, ako je vlhkosť alebo nečistoty.

Dôležitou časťou bude značenie dát, kde na danom odtlačku sa vyskytujú prejavy ochorenia. Toto označovanie bude musieť prebehnúť čiastočne ručne, no v prípade syntetických dat sa bude jednať o automatizovanú činnosť, ktorá bude bežať súbežne s generovaním ochorení do odtlačkov.



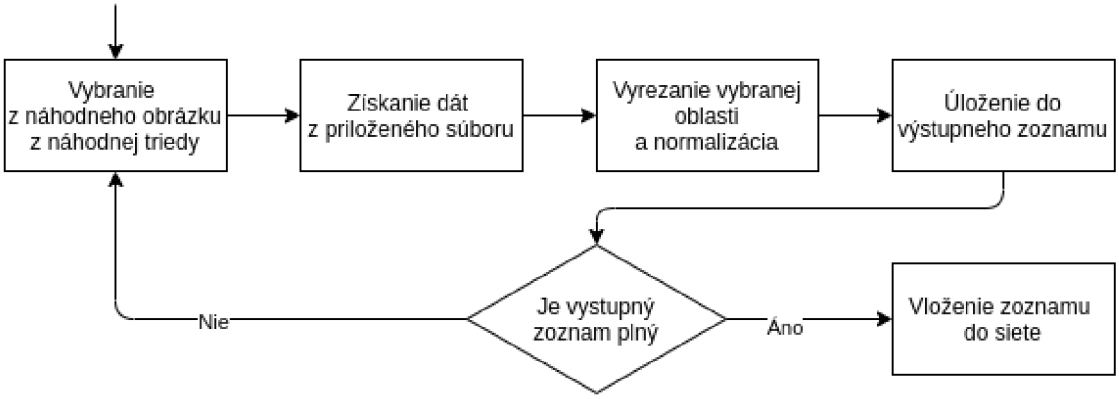
Obr. 5.1: Príklad vstupných výrezov (zľava doprava): dyshidróza, bradavica, zdravá oblasť, iné poškodenie. Zdroj: Interná databáza skupiny STRaDe.

Pri zostavovaní databázy bude dôležité pokrytie čo najväčšieho množstva ochorení, ale aj bežných defektov, spôsobených vonkajšími vplyvmi, či zdravými oblasťami. Príklady rôznych druhov, s ktorými sa neurnová sieť môže pri bežnom nasadení stretnúť, môžme vidieť na obrázkoch 5.1. Samotná sieť nebude vidieť obrázok takto komplexne, ale bude vidieť len jeho časť, ktorá bude navyše znormalizovaná pre vstupnú vrstvu. Viac o tomto procese bude popísané v kapitole implementácie, konkrétne v podkapitole 5.4.

5.2 Príprava datasetu

Pre prípravu datasetu sa ukázalo ako vhodnejšie použiť generátory, tieto generátory sú funkcie, ktoré pristupujú do nami vytvorenej databázy a vyberajú z nej položky podľa potreby. Ich hlavná výhoda je ušetrená operačná pamäť, nakoľko odpadá potreba ukladania veľkého množstva dát. Dáta si môže učiaca funkcia vyžiadať zavolaním generátora. Nevýhodou môže byť mierne zdržanie v rámci samotného učenia, pretože dáta sa musia pripravovať počas behu učenia.

Samotné obrázky s odtlačkami budú načítavané z užívateľom definovaného adresára, ako už bolo spomenuté v predchádzajúcej podkapitole. Každý obrázok bude musieť byť sprevádzaný istou formou metadát, v ktorých bude zapísané, aká časť odtlačku je poškodená akým typom ochorenia. Obrázky, ktoré nebudú obsahovať sprievodné metadáta, budú vynechávané, aby sa zamedzilo deformácií datasetu. Po načítaní obrázku do generátora budú metadáta spracované a určia sa z nich oblasti s ochoreniami, ktoré sa majú vyrezať a pripraviť. V prípade čistých obrázkov bude sprievodný súbor prázdny. V tom prípade sa z



Obr. 5.2: Vývojový diagram prípravy vstupného zoznamu siete

daného obrázku vyberú náhodné oblasti a zaznačia sa ako nepoškodené. Problémom môžu byť chybne zaznačené oblasti a oblasti s nulovou veľkosťou, pripadne príliš veľké oblasti. Riešenia týchto problémov sú popísane v podkapitole 6.3.

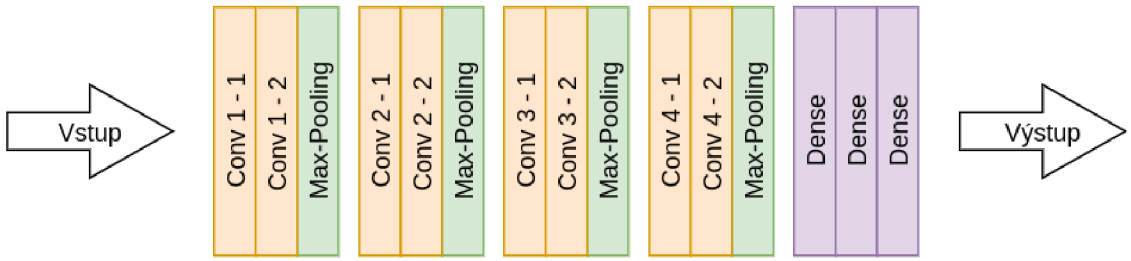
Poslednou časťou tejto fázy bude vhodná normalizácia pre neurónovú sieť. Tu bude využité odstránenie pozadia pomocou vyhľadávania kontúr a ich vyplnenia. Ďalším krokom bude adaptívne prahovanie, ktoré nájde využitie pri odtlačkoch s malým rozdielom v stupňoch šedej. Následne sa tieto vyrezané oblasti znormalizujú na veľkosť vstupnej vrstvy a vložia do výstupného zoznamu spolu s ich označením. Po naplnení tohto zoznamu budú vkladané do neurónovej siete.

5.3 Vytvorenie modelu neurónovej siete

Model bude navrhnutý ako sekvenčný a jeho vytvorenie bude vychádzať z už existujúcich riešení pre detekciu objektov na obrázku. Hlavným vzorom bude model VGG16, ktorý už bol predstavený v kapitole 4.2. Tento model bude obsahovať vstupnú vrstvu, do ktorej bude vkladaný obrázok na vyhodnotenie. Sieť podobne ako je tomu pri VGG16 bude pozostávať z niekoľkých blokov konvolučných a max-pooling vrstiev. Posledná max-poolingová vrstva v poslednom bloku bude plne prepojená s vyhodnocovacím blokom, ktorý bude obsahovať dve plne prepojené vrstvy a jednu výstupnú vrstvu. Táto výstupná vrstva bude obsahovať pravdepodobnosti výskytu jednotlivých tried. Tieto pravdepodobnosti budú využité pri samotnom vyhodnocovaní.

Samotná architektúra modelu ako už bolo spomenuté, je vytvorená z blokov. Všetky konvolučné vrstvy budú využívať aktivačnú funkciu, ReLu tak ako tomu bolo pri originálnom modeli. Prvý blok bude obsahovať vstupnú konvolučnu vrstvu, do ktorej budu vkladané obrázky. Rozmer tejto vrstvy bude 116x116 pixlov. Využívať bude 64 filtrov s veľkosťou 3x3. Nasledovať bude rovnaká konvolučná vrstva s rovnakými parametrami. Nakoniec bloku bude pridaná max-pooling vrstva, ktorá bude využívať jadro 2x2 s krokom dva. Táto vrstva zmenší obrázok o polovicu. Nasledovať budu 3 bloky podobné tomu prvému, pričom s každým blokom sa zdvojnásobí počet filtrov v konvolučných vrstvách.

Posledný blok bude obsahovať dve skryté plne prepojené vrstvy, ktoré budú mať 768 prepojení a využívať budú rovnako aktivačnú funkciu *ReLu*. Za každou plne prepojenou vrstvou bude *dropout* vrstva pre zníženie pretrénovania siete. Poslednou vrstvou v tomto



Obr. 5.3: Model siete.

bloku ako aj v sieti bude výstupná vrstva. Počet jej výstupov určuje koľko tried bude sieť schopná rozoznať. V návrhu sa vždy rata, že prvou triedou je vždy čistá oblasť. Nasledujúce triedy sú určené pre samotné ochorenia. V prípade, že sieť bude vyhľadávať len jedno ochorenie, je výhodnejšie použiť ako aktivačnu funkciu tejto vrstvy *sigmoid*. V prípade viacerých ochorení by už však *sigmoid* nebol schopný rozlíšiť viacero tried. Preto pri viac triedach sa ako aktivačná funkcia využíva *softmax*.

Pre samotné trénovanie je ešte potrebné vybrať loss funkciu, ktorá nám určuje úspešnosť trénovania siete. Pri výbere je podstatný rozsah, ktorý od siete očakávame a počet tried. Jednotlivé loss funkcie boli popísane v podkapitole 4.3. Pre prípad, že sieť bude obsahovať len 2 výstupné triedy, bude využitá *binárna krížová entropia*. V prípade, kedy sieť rozlišuje viac ako dve triedy, bude využívaná *kategorická krížová entropia*. Tieto dve *loss funkcie* boli predstavené v podkapitole 4.3. Posledným podstatným krokom je voľba optimalizátora. Ako výhodný sa ukazuje optimalizátor SGD, ktorý je jedným z najpoužívanejších. Bližšie je popísaný v 4.4. Samotná rýchlosť učenia bola nastavená na 0,001.

5.4 Učenie neurónovej siete

Nastavenia parametrov učenia siete bude môcť užívateľ ovplyvniť pomocou vstupných prepínačov. Bude mať možnosť zvoliť počet epoch, rovnako ako aj počet krokov v jednotlivej epoche. Ďalším dôležitým parametrom je veľkosť „batch“, ktorá odzrkadľuje koľko vzoriek sa vloží do siete na jeden krok epochy.

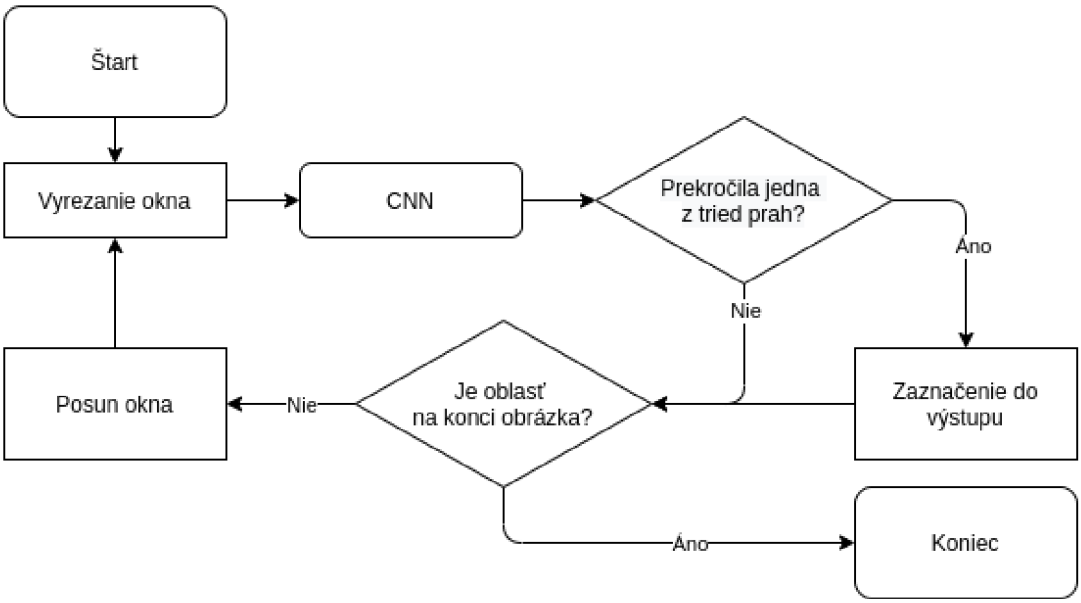
Samotné trénovanie už bude prebiehať v rámci funkcií, ktoré poskytuje daná knižnica. Dáta do siete počas trénovania budú vkladané na vyžiadanie samotnej trénovacej funkcie, čím odpadne nutnosť tvorby rozsiahlych a pamäťovo náročných polí. Návrh tohto generátoru sme si predstavili v 5.2. Pomer jednotlivých tried bude musieť byť čo najviac rovnomerný a každý „batch“ bude obsahovať rovnaký počet vzoriek z každej triedy. Dôležitou vlastnosťou pri učení bude výber loss funkcie, ktorá bude určovať, ako ďaleko je sieť od ideálneho riešenia. Samotný popis učenia bol popísaný v kapitole 4.8 a výber vhodnej loss funkcie v kapitole 5.3.

Dôležitou časťou bude vytváranie a ukladanie váh modelu. Aby užívateľ mal prístup k už pred-trénovaným modelom pre klasifikáciu. Tieto zálohy sa budú vytvárať počas učenia a bude sa ukladať vždy epocha s najvyššou presnosťou nad testovacím datasetom. V prípade potreby bude môcť užívateľ využívať tieto zálohy pri pokračovaní trénovania, prípadne pri samotnom klasifikovaní ochorení na odtlačkoch.

5.5 Očakávaný výstup

Samotný princíp vyhľadávania bude spočívať v prechádzaní vstupného okna obrázkom, ktorý užívateľ vloží na vstupe. Po každom posunutí tohto okna bude oblasť nad ktorou sa okno nachádza, vybraná a vložená do siete. Tá tento výrez spracuje a na jej výstupnej vrstve bude pre každú triedu zobrazená pravdepodobnosť. Vyberie sa tá trieda, ktorej výsledná pravdepodobnosť prekročí prah zvolený užívateľom. Funkcia softmax priradí každej triede pravdepodobnosť a súčet pravdepodobností týchto tried musí byť 1. Takže pri dostatočne vysokom prahu nehrozí, že by sieť dospela do nerozhodného stavu. Do výstupného záznamu sa vloží názov ochorenia, pravdepodobnosť, že sa jedná o dané ochorenie a dva body, ktoré budú určovať oblasť.

Výstupom siete bude textový dokument, do ktorého budú vložené záznamy o jednotlivých nájdených oblastiach s ochorením. Ďalšou možnosťou bude aj výstup na pôvodný obrázok, do ktorého algoritmus tieto oblasti vyznačí farebne.



Obr. 5.4: Vývojový diagram pre tvorbu výstupu.

Kapitola 6

Implementácia

Táto kapitola popisuje implementáciu práce, rozdelená je do niekoľkých podkapitol nasledovne. V prvej časti 6.1, si predstavíme samotné prostredie a použité komponenty. Nasledovať bude tvorba samotnej databázy, kde bude popísané z čoho bola vytvorená. Ďalšou časťou 6.3 je príprava trénovacích a testovacích množín za pomoci generátorov. Popísaný bude generátor týchto množín a riešenia problémov s ním spojených. Predposledná podkapitola 6.4 sa sústredí na tvorbu samotného modelu konvolučnej siete. Na konci kapitoly v časti 6.6 bude popísaná tvorba výstupu.

6.1 Prostredie a použité komponenty

Aplikácia bola vytvorená ako konzolová. Nastavenia vykonáva užívateľ pomocou vstupných parametrov. Program funguje v dvoch módoch. Trénovací, kedy užívateľ nastaví umiestnenia datasetu pre generátory a samotné parametre siete. Druhým módom je detegovanie poškodení odtlačku prsta. V tomto prípade sú do siete uložené váhy z checkpointu a umiestnenie odtlačkov určených na detekciu. Pre tvorbu programu bol využitý programovací jazyk Python 3.6. O spracovanie obrázkov sa stará knižnica OpenCV. Prácu s maticami zabezpečuje modul NumPy. Pre tvorbu samotnej neurónovej siete bol využitý framework Keras a modul TensorFlow. Aplikácia bola testovaná na Ubuntu 18.04.

6.1.1 Knižnica TensorFlow a API Keras

Pre tvorbu neurónovej siete bola zvolená knižnica TensorFlow (TF) spolu s jej API (z angl. application programming interface, rozhranie pre programovanie aplikácií) s názvom Keras, ktoré je zamerané na tvorbu konvolučných a rekurzívnych sietí, s dôrazom umožnenia rýchleho experimentovania a návrhu komplexnej neurónovej siete, ktoré podporuje aj samotný programovací jazyk Python 3.7.

Základom celej neurónovej siete v Keras je model. Je to objekt, v ktorom sú združené všetky inštancie tried vrstiev daného modelu. Tieto vrstvy je ale možné samostatne volať, čo nám umožňuje ich znovu používanie, rovnako ako aj využívanie ich váh, podobne ako je popísané v podkapitole 4.6. V prípade, že je návrh siete akceptovateľný, je možné model skompilovať a začať ho trénovať. Trénovanie modelu prebieha vytvorením dvoch množín, jednej vstupnej a druhej výstupnej. Tieto množiny sú prevedené na číselné hodnoty (napríklad v prípade obrázka 0 až 255) a vložené do funkcie `fit()`. Táto funkcia okrem váhových dostane aj údaj o počte iterácií nad datasetom. Po ich dokončení môže byť vytrénovaný model vyskúšaný na reálnych dátach pomocou funkcie `evaluate()`. [7]

6.2 Databáza

Pre trénovanie neurónovej siete bolo potrebné vytvoriť veľkú databázu s čo najrozličnejšími tvarmi a priebehmi ochorenia. Rozsah tejto databázy musí byť dostatočný na to, aby sieť pri učení neuviazla v lokálnom minime. Tu ale nastáva problém, keďže existujúce databázy neposkytujú dostatočné množstvo obrázkov ako môžme vidieť aj v tabuľke 6.1. Riešením je generovanie syntetických odtlačkov prsta, do ktorých sú potom vnesené prejavy ochorenia, menovite bradavice a dyshidróza.

Presne tento postup bol zvolený pri riešení práce. Ako prvá bola vytvorená sada syntetických odtlačkov prstov za pomoci generátoru SFinGe. Odtlačky boli generované ako už čiastočne poškodené, aby ich stav bol čo najvernejší reálnym odtlačkom. Tieto odtlačky boli potom vložené do generátora ochorení. Pre generovanie ochorení bola využitá diplomová práca „Generování onemocnění kůže do syntetických otisků prstů“ [2]. V tejto práci bol vytvorený konzolový program, ktorý generuje bradavice a ekzém. Problémom však naďalej zostávalo generovanie prejavov dyshidrózy. Preto bol navrhnutý nový generátor pre generovanie tohto ochorenia. Jeho implementácia je popísaná v podkapitole 6.2.2.

Tieto generátory ochorení ďalej boli rozšírené o tvorbu notácie o ochorení na danom syntetickom odtlačku. Výsledne poškodené syntetické odtlačky môžme vidieť na obrázku 6.1 Výsledná databáza potom bola rozdelená na jednotlivé množiny TRAIN, TEST a PREDICT.

Obr. 6.1: Príklad syntetických odtlačkov prsta so simulovaným poškodením dyshidróza (vľavo) bradavice (vpravo).

Už pri prvých spusteniach natrénovaného modelu bolo zrejmé, že sieť si nie je schopná poradiť s masívnejším poškodením odtlačku prsta, spôsobeného ochorením, preto bolo potrebné pridať niekoľko desiatok obrázkov reálnych odtlačkov, ktoré boli zasiahnuté sledovanými ochoreniami. Tieto odtlačky boli získane z databázy výskumnej skupiny STRaDe. Po tejto úprave sa schopnosti siete rozpoznávať ochorenia zlepšili, no problémom boli stále mechanicky poškodené oblasti v odtlačkoch (napr. vplyv potu).

Ďalším problémom boli samotné reálne nezasiahnuté oblasti. Tieto oblasti sú často zasiahnuté iným typom poškodení, ktoré je náročné nasimulovať v dostatočnej kvalite. Z tohto dôvodu bola databáza rozšírená o reálne odtlačky, ktoré nie sú nijak poškodené, alebo neobsahujú poškodenia spôsobené hľadanými ochoreniami. Pri tomto rozšírení bola využitá databáza NIST SD4 DB [29]. Táto databáza obsahuje okolo 4000 odtlačkov rôznej kvality

prevažne získavaných z daktyloskopických kariet. Nie všetky tieto odtlačky sú vhodné pre trénovanie neurónovej siete. Bolo preto potrebné databázu ručne pretriediť a vybrať len tie odtlačky, ktoré vhodne rozširovali spektrum poškodení. Odstraňované boli nekvalitné snímky ktoré by nijak pri učení nepomohli.

Samotné zloženie databázy v číslach môžme vidieť v tabuľke 6.1. Ako môžme vidieť tak pomer reálnych a syntetických odtlačkov je 1 k 5. Z týchto reálnych odtlačkov je len 4,5 % postihnutých ochorením, preto výsledok bude hlavne závisieť od generátorov ochorení a ich kvality.

Tabuľka 6.1: Databáza pre učenie, testovanie a verifikáciu siete (R - reálne odtlačky, S - syntetické odtlačky).

Typ poškodenia	**TRAIN**		**TEST**		**PREDICT**	
	S	**R**	**S**	**R**	**S**	**R**
Bradavice	7000	20	2850	20	4	8
Dyshidróza	7000	50	2990	50	1	13
Bez hladaneho poškodenia	0	2829	0	620	0	20
Spolu	14000	2899	5840	690	5	41

6.2.1 Zloženie databázy a jej tvorba

Pre zjednodušenie tvorby databázy bol vytvorený skript *set_up.py* v Python 3. Tento skript dostáva ako vstupné parametre adresár s umiestnením jednotlivých odtlačkov a JSON súborov. Ďalej sú to tri parametre, ktoré určujú rozdelenie do jednotlivých množín (TRAIN, TEST, PREDICT). Týmito parametrami užívateľ urči pomer rozdelenia vstupného adresára. Posledným dôležitým parametrom je parameter pre určenie triedy. Ďalšia sada parametrov označuje čisté odtlačky podľa očakávaného formátu siete. Posledným nemenej podstatným parametrom je *delete* ktorý zmaže aktuálne adresáre množín.

Po spustení skript overí, či existujú cieľové adresáre množín v prípade, že nie sú tieto adresáre vytvorené. V týchto adresároch sa ďalej vytvoria pod-adresáre pre jednotlivé triedy. Názov triedy je braný zo vstupného parametru definovaného užívateľom. Zo vstupného adresára sa vytvorí zoznam názvov súborov z ktorých sa vyfiltrujú len obrázkové súbory s odtlačkami. Pred samotným kopírovaním sa overí, či jeho vstupnom adresári dostatočný počet súborov pre rozdelenie do množín v prípade, že nie je skript ukončený. Nasleduje kopírovanie do množín pomocou funkcie *copy_images*. Funkcia na vstupe dostáva zoznam súborov a interval, ktorý má z tohto zoznamu prekopírovať. Funkcia v cykle získava cesty k obrázkom a vyberá z nich názvy súborov. Tento názov sa použije na vyhľadanie JSON súboru s popisom ochorenia, ktorý sa spolu s obrázkom následne prekopíruje do cieľového adresára. Ak sa v adresári už nachádza názov s rovnakým menom, je tomuto súboru priradené meno s číslom poradia, aby sa zabránilo nahradzovaniu súborov.

6.2.2 Generátor ochorenia dyshidróza

Ako už bolo spomenuté bolo nutne vytvoriť generátor prejavov dyshidrózy do syntetických odtlačkov prsta. K tomu boli využité znalosti o generovaní dyshidrozy popísane v už existujúcej bakalárskej práci „Generování onemocnění kůže do syntetických otisků prstů z SFinGe“ [27]. V tejto praci bol síce vytvorený generátor daného ochorenia no užívateľské rozhranie neumožňovalo automatizáciu procesu.

Generátor rozširuje funkcionalitu diplomovej práce [2], tým že do nej pridáva novú triedu *DyshidroticGenerator*. Táto trieda dostane pri svojej inicializácií vstupný obrázok do ktorého sa bude dané ochorenie generovať. Následne sa začne s generovaním daných prejavov. Prvým sú dlhé biele čiary, pre tento účel je využitá funkcia ktorá už bola napísaná v pôvodnej diplomovej práci a to *draw_eczema_lines*. Ako druhe sú generované biele oblasti pre tento účel bola vytvorená funkcia *draw_dishydrotic_spots*.

Funkcia vytvorí náhodne 2 až 8 oblasti na odtlačku. Počiatočným krokom je výber bodu na ploche odtlačku prsta k čomu bola využitá funkcia *is_far_enough*. V okolí tohto bodu je vytvorený štvorec do ktorého sa bude dané ochorenie vyznačovať. Tento štvorec je zároveň aj využitý pri zapisovaní notácie. Následne sa začne s generovaním plôch. Ako prve sa náhodne rozhodne aká farba bude využitá v danej oblasti či bude daná plocha generovať bielu oblasť alebo bude naopak biele oblasti mazať. Mazanie oblasti sa môže zdať na prvý pohľad kontraproduktívne no umožňuje vytvárať male oblasti kde presvitajú papilárne línie tak ako je tomu pri reálnom ochorení.

Po zvolení farby sa na danú pozíciu vyznačí kruh vyplnený zvolenou farbou a dve na seba kolmé úsečky. To to riešenie bolo prebraté z práce [27], nakoľko sa ukázalo ako účinne. Ďalším krokom ktoré už v pôvodnej práci nebolo je tvorba svetlejších a tmavších podoblasti. Týchto podoblasti je v okolí hlavného kruhu rôzne množstvo a líšia sa v odtieňoch sivej čo vytvára práve dojem šupiniek spôsobených dyshidrózou. Poloha týchto šupiniek sa rata z pozície hlavného kruhu pripočítaním náhodnej malej hodnoty. V spojení s náhodnými odtieňmi sivej dosahujeme pomerne náhodnú textúru napadne sa podobajúci prejavom ochorenia. Výsledky tohto generátoru môžme vidieť na obrázku 6.2. Pre porovnanie bol pridaný aj pôvodný výstup generátoru z práce [27].

Obr. 6.2: Porovnanie generátorov (vľavo) pôvodný generátor z práce [27], (vpravo) novo vytvorený generátor ako súčasť tejto práce. Obidva generátory boli spustené na syntetickom odtlačku bez poškodenia.

6.3 Generovanie trénovacích a testovacích množín

V tejto podkapitole je popísaná tvorba trénovacích obrázkov. Pre zníženie pamäťovej náročnosti bol namiesto tvorby veľkého zoznamu uloženého v pamäti uprednostnený generátor. Pre tieto funkcie bol vytvorený modul *TLmaker.py*. Volaný je podľa potreby samotnou trénovacou funkciou *fit_generator*. Tej vracia zadaný počet pred-spracovaných vzoriek. Volanie generátoru je zabezpečené pomocou funkcie *create_train_list*. Tejto funkcií je vložený ako prvý parameter cesta k adresáru s obrázkami. Nasleduje tvar jednotlivých obrázkov, ktorý je reprezentovaný usporiadanou trojicou: dĺžka, šírka a počet kanálov. Nasleduje kľúčové slovo *batch_size*, ktoré nám určuje počet vygenerovaných vzoriek vo výstupnom liste. Posledným parametrom je počet výstupných tried.

Ako prvý je vytvorený zoznam vstupných obrázkov v adresári. Tento zoznam je potom náhodne premiešaný funkciou *shuffle* z modulu *random*. Premiešanie zoznamu znižuje šancu pretrénovania konvolučnej siete. Po premiešaní sa začne s vyberaním názvov obrázkov zo vstupného zoznamu. Pomocou názvu obrázku sa vyhľadá v rovnakom adresári súbor JSON s uloženými informáciami o ochorení na danom obrázku. Každý obrázok v databáze využitý pri trénovaní a testovaní musí mať tento súbor, inak sa s takýmto obrázkom nepracuje. V prípade nepoškodených odtlačkov je tento súbor prázdny. Príklad záznamu v metadátach môže vyzerať napríklad takto:

```
[{"Type": "wart", "p1": [239, 179], "p2": [370, 290]}]
```

Názov súboru je potom vložený do funkcie *parse_json_json*. V nej je súbor otvorený a pomocou funkcie *read_json* z modulu *panda* je jeho obsah prevedený do zoznamu záznamov. Každý platný záznam obsahuje názov choroby a dva body, ktoré určujú obdĺžnikovú oblasť, v ktorej sa dané ochorenie nachádza. Po načítaní záznamu je overená jeho platnosť, záznam musí obsahovať správne označenie choroby. Ďalšou podmienkou je, že vstupné body nesmú zasahovať mimo obrázok, prípadne výsledná oblasť nesmie mať nulovú veľkosť. Tento problém nie je častý, no napríklad pri ručne označovaných dátach môže dojsť k chybe.

O dosť častejším problémom, s ktorým bolo nutné sa vysporiadať, sú príliš veľké oblasti. Tento problém je častý u dyshidrózy, kde býva zasiahnutá väčšia oblasť odtlačku. Po vykonaní normalizácie nad takouto oblasťou by dochádzalo k strate informácií (odfiltrovanie drobných znakov). Preto sa takáto oblasť odošle do funkcie *divide_frame*, ktorá oblasť rozdelí na menšie časti, ktoré je už sieť schopná spracovať bez straty detailov. Po prejdení celého zoznamu záznamov je vytvorený list oblastí, ktorý sa ďalej spracuje samotnou funkciou generátora.

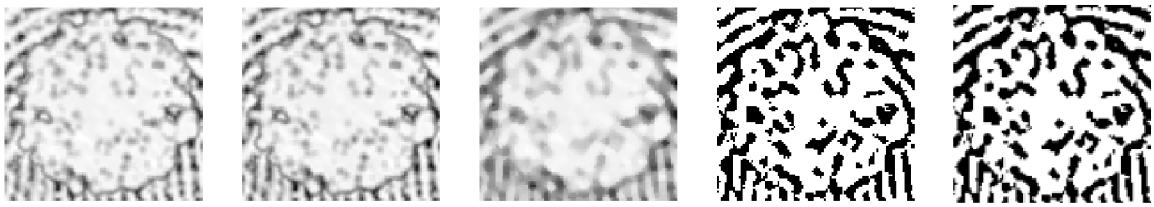
Po načítaní záznamov nasleduje tvorba samotných výrezov zo vstupného obrázka. Funkcia *frames_generator* postupne vyberá oblasti zo zoznamu z predchádzajúceho kroku. Táto oblasť sa vloží spolu so vstupným obrázkom do normalizačnej funkcie (bude popísané v podkapitole 6.3.1). Výsledný normalizovaný obrázok je pridaný do výstupného poľa obrázkov. Spolu s normalizovaným obrázkom je do druhého zoznamu označení vložený výsledný vektor. Tento vektor má dĺžku rovnú počtu tried a typ triedy je označený jednotkou. Vektor slúži ako náhrada namiesto slovného popisu ochorenia. Tento formát je čitateľný pre neurónovú sieť a určuje aj výsledné umiestnenia pravdepodobnosti pre jednotlivé triedy.

Ďalej funkcia *frames_generator* začne generovať výrezy mimo oblasti ochorenia. Počet generovaní týchto výrezov je daný parametrom *no_frames*. Každej oblasti sa náhodne vygeneruje prvý bod, k nemu sú vygenerované dve dĺžky. Tieto dĺžky sú generované náhodne

v rozmedzí 0 až maximálna dĺžky oblasti. V prípade, že maximálna dĺžka v súčine s aktuálnou polohou je mimo hranice obrázka, je táto hodnota upravená na maximálnu možnú veľkosť. Po vygenerovaní sa vyráta pozícia druhého bodu. Výsledný rámec sa vloží do funkcie *iou*, ktorá skontroluje, či sa výsledná oblasť neprekrýva s postihnutou oblasťou. Ak sa tieto oblasti neprekrývajú, je takáto oblasť vložená do normalizátoru a výsledný normalizovaný obrázok je spolu s popisujúcim vektorom vložený do výstupného zoznamu, podobne ako tomu bolo u postihnutých oblastí. Funkcia načítava obrázky do doby pokiaľ nie sú dĺžky výstupných zoznamov rovné veľkosti očakávaného počtu zadaného pomocou parametra *batch_size*. Po splnení tejto podmienky je výsledný zoznam pretypovaný na NumPy pole a vrátený funkcií.

6.3.1 Normalizácia

Z nutnosti, aby všetky vstupné obrázky mali rovnaké parametre na vstupe do siete, je potrebné ich normalizovať. O to sa stará funkcia *normalizér*. Táto funkcia na vstupe dostane pomocou parametra obrázok, a oblasť ktorú je potrebné vyrezať. Oblasť je vyrezaná z obrázku. Nasledné upravy výrezu jednotlivými funkciami môžme vidieť na obrázku 6.3. Ako prvé je výrez prevedený do stupňov šedej pomocou *cvtColor*. Po prevedení do šedej je za pomoci funkcie *medianBlur* odstránený drobný šum, spôsobený napríklad podkladom. Ďalším krokom je zvýraznenie papilárnych línií. Pre tento úkon bola vybraná funkcia *adaptiveThreshold* z modulu `cv2`, ktorá vykonáva adaptívne prahovanie. Prah je vypočítaný pomocou krížovej korelácie s Gaussovským oknom. Tento typ prahovania odstraňuje aj výraznejšiu zrnitosť spôsobenú podkladom odtlačku. Posledným krokom je zmenenie veľkosti obrázka na veľkosť vstupnej vrstvy konvolučnej siete, ktorá je 116x116 , pomocou funkcie *reshape* a zmenenie výsledného poľa na jeden kanál.



Obr. 6.3: Kroky normalizácie (zľava doprava), originálny výrez, prevedený do šedej, medianové rozmazanie, adaptívne prahovanie, zmena veľkosti.

6.4 Model siete

Model je jadrom celej práce. Jeho návrh bol predstavený v kapitole 5.3. Vychádza z už v praxi často používaného modelu VGG16, ktorý bol popísaný v kapitole 4.2. Pre zostavenie modelu bude využité API Keras s modulom TensorFlow. Toto API nám poskytuje pomerne dobrý a rýchly spôsob tvorby. Ako prvé je nutné zvoliť typ modelu. Využívať budeme sekvenčný model. V samotnom programe tento model vytvoríme volaním funkcie *Sequential*. Ďalej začneme vytvárať samotnú sieť podľa návrhu popísaného v kapitole 5.3.

V modeli, ktorý je vytvorený v Keras, je sieť vytváraná radením jednotlivých vrstiev za seba. Ako môžme vidieť na ukážke, vo vstupnej konvolučnej vrstve ako prvé určíme argumentom počet filtrov. Nasledujúca usporiadaná dvojica je veľkosť konvolučného okna,

ktoré je nastavene na 2×2. Posledným parametrom, ktorý je vo všetkých vrstvách, je aktivačná funkcia ReLu. Najzaujímavejšími parametrami v tejto vrstve sú *input_shape* ktorý určuje veľkosť vstupu tzn. vstupných obrázkov a *data_format* ktorý sieti nastavuje, že vo vstupných dátach je posledným indexom číslo kanálu.

Nasleduje ďalšia konvolučná vrstva s rovnakými parametrami, ako tá predchádzajúca, s výnimkou vstupných parametrov. Nasledovaná je poolingovou vrstvou, ktorá využíva veľkosť okna 2×2 a posun o 2 pixely. Táto vrstva nám zo vstupnej matice urobí o polovicu menšiu. Týchto blokov nájdeme v siete ešte 4. Jediná vec v čom sa líšia, je počet filtrov ktorý sa zdvojnásobuje. Po prejdení týmito blokmi dostaneme veľkosť matíc 7×7. Tie sú plne prepojené s nasledujúcou vrstvou.

Vyhodnocovací blok obsahuje dve plne prepojené vrstvy. Tieto vrstvy sú v Keras označené ako *Dense*. Každá z týchto vrstiev má 768 kanálov a využíva sa aktivačná funkcia ReLu. Za ňou nasleduje *Dropout* s parametrom 0,5 tzn., že vo vrstve je počas učenia náhodne ignorovaná polka neurónov a tým sa zabraňuje pretrénovaniu siete. Poslednou vrstvou v tomto bloku je výstupná vrstva. Jej parameter je definovaný podľa vstupného parametru *number_of_outputs*, ktorý nastavuje priamo užívateľ. Tento parameter nám určuje koľko bude mať sieť výstupných neurónov, čo označuje, koľko tried je sieť schopná rozoznať. Ďalej je týmto parametrom určená aj aktivačná funkcia poslednej vrstvy.

Po zostavení siete je potrebné pripojiť optimalizátor, ktorý bol vybraný v návrhu a následne nastaviť rýchlosť učenia. Nasleduje posledný krok a tým je kompilácia celého modelu pomocou funkcie *compile*. V parametri tejto funkcie sa nastavuje loss funkcia a vloží sa sem aj objekt optimalizátora. Po úspešnom skompilovaní je model pripravený na trénovanie.

6.5 Trénovanie siete

Po vytvorení modelu a predstavení generátorov trénovacích a testovacích množín je potrebné model natrénovať. Ako prví krok je nutné nastavenie záchytných bodov. Tieto záchytné body Keras ukladá ako súbory a v nich sú uložené váhy jednotlivých prepojení. Pre sledovanie pokroku siete bola zvolená presnosť nad testovacím setom. Ukladané sú len záchytné body, ktoré majú vyššiu presnosť, ako tie predošlé. Výhodou týchto bodov je, že v prípade nutnosti prerušenia trénovania nie je nutné začínať učenie odznova, ale sieť si načíta uloženú pozíciu. Rovnako sú tieto body využívané aj pri samotnej klasifikácií.

Z pohľadu implementácie je samotné trénovanie v Keras pomerne jednoduché, keďže knižnica disponuje dvojicou funkcií *fit* a *fit_generator*. Tieto funkcie slúžia pre trénovanie siete a už názov napovedá, že pri druhej menovanej môžme namiesto používania rozsiahlych zoznamov využiť o niečo menej pamäťovo náročný generátor. Preto aj v implementácií využijeme funkciu na trénovanie pomocou generátorov *fit_generator*. Hlavným parametrom tejto funkcie je objekt generátoru trénovacej množiny. Tu a pri definovaní generátora pre testovacie dáta bude využitý nami vytvorený generátor, predstavený v kapitole 6.3. Ďalšími parametrami v tejto funkcií je počet krokov za jednu epochu a počet epoch. Tieto hodnoty sú plne ponechané na užívateľa, ktorý ich nastaví pomocou parametrov z príkazového riadku. Predposledným parametrom je počet testovacích krokov. Pomocou týchto testovacích krokov vieme určiť s akou presnosťou vie sieť rozoznávať jednotlivé výrezy. Pre optimálny počet krokov bol využitý vzorec počet trénovacích krokov deleno veľkosť *batch_size*. Presnosť nad trénovacím setom je dôležitá aj pre tvorbu záchytných bodov, ktorých objekt je do funkcie vložehý pomocou kľúčového slova *callbacks*.

6.6 Klasifikátor ochorenia

Odtlačok je po vstupe do funkcie *find_dissase* prekonvertovaný do odtieňov šedej. Vytvorená je jeho kópia, tá prejde prahovaním aby sa zvyraznili kontury odtlačku. Tie su nasledne vyhľadané funkciou *findContours*. Tá vyhľadá kontúry výrazných oblastí na obrázku, hlavne oblasť samotného odtlačku. Tieto kontúry sú uložene a následne pomocou *drawContours* sú v originálnom obrázku okolia týchto kontúr vyplnené bielou farbou, čím sa odstránia drobné nečistoty a šum, ktorý by mohol vytvárať falošné detekcie ochorení. Takto spracovaný obrázok bude obsahovať len samotný odtlačok a je odstranené jeho okolie. Následne je na tento obrázok aplikované adaptívne prahovanie. Tento proces zvýrazní papilárne línie a ochorenia. Takýto obrázok je potom pripravený na prehľadanie neurónovou sieťou.

Samotné vyhľadávanie ochorenia prebieha posúvaním konvolučného okna siete. Toto okno má niekoľko veľkosti, aby sa zachytili aj drobné prejavy ochorenia, ktoré by sa pri väčších oknách stratili pri normalizácií. Posúvanie okna po obrázku sa vykonáva s adaptívnym krokom. Tento krok je upravovaný pomocou predchádzajúcich výsledkov zo siete. Pokiaľ sieť nič nedeteguje, je veľkosť kroku rovná maximálnej veľkosti. Naopak, pokiaľ sa na výstupe siete začne zvyšovať pravdepodobnosť, že okno sa nachádza nad poškodenou oblasťou, je tento krok delený dvomi až kým nedosiahne minimálneho kroku, alebo sa nedostane mimo poškodenej oblasti. Použitie adaptívneho kroku je výhodné hlavne z pohľadu rýchlosti vyhľadávania, nakoľko nie je potrebné podrobne prechádzať zdravé oblasti odtlačku prsta, pripadne jeho okolie.

Po posunutí okna je vždy dané okno z obrázku vyrezané. Ďalej je podobne ako pri učiacej fáze znormalizované na veľkosť vstupnej vrstvy siete. Tento proces je popísaný v podkapitole 5.4. Výrez je vložený pomocou funkcie *predict* do modelu siete, kde je spracovaný. Výstup tejto funkcie vracia zoznam pravdepodobností, ktorý je identický s tým, ktorý sme používali pri učení siete a predstavili sme si ho v predchádzajúcej kapitole. Z tohto zoznamu sú potom vybrané pravdepodobnosti pre jednotlivé ochorenia.

Program využíva dve pravdepodobnosti. Jedna je využívaná ako prah detekcie, po prekročení tohto prahu je daná oblasť vložená do zoznamu na spracovanie a je označená za kandidátnu oblasť. Druhý prah sa využíva, pokiaľ je dosiahnutá pravdepodobnosť, ktorú potom sieť označí za určite poškodenú. Samozrejme platí, že medzi pravdepodobnosťami platí vzťah, že prvá je väčšia ako druhá.

Po prejdení celého obrázka je vytvorený výstupný súbor JSON, do ktorého sú uložené jednotlivé detekcie. Vo formáte názov ochorenia, body určujúce oblasť a pravdepodobnosť, že sa daná choroba nachádza v oblasti. Druhým možným výstupom praktickejším pre samotného užívateľa, je vykreslenie týchto oblastí do obrázku. Tento výstup zaleží od nastavenia prepínačov zadaných užívateľom. Oblasti sú do obrázku vykreslené podľa už predstavených 2 prahov pravdepodobnosti. Oblasti, ktoré prekročili kandidátny prah, sú oblasti vyznačené polopriesvitnými obdĺžnikmi, ktoré sú vyplnené farbou. V prípade oblastí, ktoré prekročili aj prah kedy je oblasť považovaná za poškodenú, sú tieto oblasti vyznačené do obrázka farebným rámom. Po vykreslení všetkých oblasti do obrázku je tento obrázok vložený do objektu *figure* z knižnice *matplotlib*. Ďalej je do tohto objektu vložený aj čistý obrázok, aby užívateľ mohol porovnať zaznačené oblasti so skutočnosťou. Následne sú tieto dva obrázky uložené do výstupného adresára.

6.7 Príprava testovania

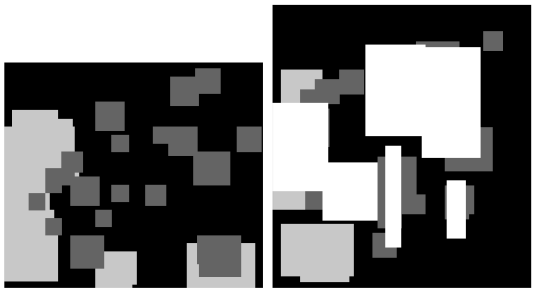
Pre testovanie samotného algoritmu bol vytvorený skript *tests.py*. Tento skript ako parametre dostáva dva adresáre. Prví kde je uložený výstup siete a druhý adresár z ktorého sieť čerpala. V prvom kroku skript vytvorí zoznam všetkých JSON súborov vo výstupnom adresári. Z každého takéhoto súboru sa získa pôvodné meno, ktoré je vyhľadané v adresári, z ktorého sieť čerpala. Následne sa otvoria obidva súbory (pôvodný a výstup zo siete) a overí sa ich správny zápis. Nad každou takouto dvojicou súborov sa spúšťajú následne testy.

6.7.1 Test plochy

Prvým navrhnutým testom je test plochy. Tento test má za úlohu zodpovedať nasledujúce otázky:

- Koľko plochy ochorenia bolo vyznačenej správne, koľko vôbec a koľko nesprávne?
- Koľko z vyznačenej plochy bolo mimo ochorenia?

Všetky tieto údaje sú zbierané v percentuálnom podiele, nakoľko veľkosť obrázkov sa môže líšiť. Ako prvé je vytvorené za pomoci *np.zeros* dvoj-rozmerné pole s všetkými bunkami nastavenými na 0. Následne sú zoradené vzostupne všetky pravdepodobnosti získane z výstupného súboru siete. Toto poradie je dôležité, aby vo výslednom poli neprekrývali detegované oblasti s nižšou detekciou tie s vyššou. Následne sú tieto oblasti vyznačené do poľa pomocou funkcie *rectangle* z knižnice *OpenCV*. Týmto oblastiam je priradená špecificka hodnota podľa toho, o aké ochorenie sa jedná. Príklad výslednej „mapy“ môžme vidieť na obrázku 6.4.



Obr. 6.4: Príklady máp pre testovanie plochy, na pravo môžme vidieť mapu po vyznačení všetkých detekcií. Vľavo mapu pre zrátanie oblasti mimo ochorení

V ďalšom cykle sa začne s prechádzaním zoznamu s reálne označenými ochoreniami. Podľa bodov, ktoré sú v zázname ochorenia sa toto ochorenie vyhľadá na matici a začne sa kontrolou koľko bodov sa nachádza v danej oblasti. Ako prvé sa kontrolujú nedetegované časti, to znamená,že sa zráta počet pixelov, ktoré neboli označené. Podobne sa postupuje aj pri chorobách s tým rozdielom, že sa kontroluje počet pixelov s špecifickým stupňom šedej. Po sčítaní a uložení sa daná oblasť preznačí na skontrolovanú oblasť. Po prejdení všetkých oblasti sa spočítajú počty pixelov pre jednotlivé ochorenia ktoré sa nachádzajú mimo obrázka. Tento udaj hovorí o veľkosti plochy, ktorú sieť vyznačila naviac oproti pôvodnému ochoreniu.

6.7.2 Test správneho určenia

Nasledujúci test je dôležitý z pohľadu natrénovania samotnej siete, určuje ako dobre je sieť natrénovaná. Tento test by mal zodpovedať otázky ako:

- Aká je úspešnosť detekcie siete?
- Koľko detekcií bolo falošných, sieť ich označila za ochorenie aj keď sa tam žiadne nenachádza?
- Aká časť detekcií síce detegovala poškodené miesto, ale zle mu priradila ochorenie?

Odpovede na tieto otázky nám dajú pomerne reálny obraz na úspešnosť klasifikácie a detekcie ochorenia. Sú tiež hlavným ukazovateľom úspešnosti detekcie siete. Podobne, ako pri predchádzajúcom teste, je vytvorené dvojrozmerné pole. Do tohto poľa sa vyznačia všetky oblasti s ochoreniami z meta súboru k pôvodnému odtlačku. Tak ako pri predchádzajúcom teste sú rozdielne ochorenia vyznačené rozdielnym stupňom šedej. Po vyznačení všetkých poškodených oblasti sa začne s prechádzaním zoznamu, ktorý bol uložený vo výstupnom súbore. Pri každej oblasti sa vyberie oblasť z pomocného poľa a zráta sa počet pixelov pre jednotlivé ochorenia. Potom sa podľa typu ochorenia skontroluje, či sa v detegovanej oblasti nachádza dane ochorenie. Existujú tri stavy:

- **Správny** - sieť označila chorobu správne.
- **Chyba triedy** - sieť detegovala chorobu, ale priradila jej nesprávne. označenie.
- **Nesprávny** - sieť detegovala ochorenie niekde kde sa nenachádza.

Pre každú oblasť je vytvorená tabuľka v podobe slovníka. Po dosiahnutí jedného zo stavov je v tejto tabuľke inkrementovaná hodnota daného stavu. Po prejdení všetkých oblastí sú na konci tieto hodnoty percentuálne zobrazené užívateľovi.

Kapitola 7

Testovanie a výsledky

V tejto kapitole popíšeme výsledky práce. V podkapitolách 7.1 a 7.2 si predstavíme metodiky testovania práce, samotný priebeh testovania a tiež sa zameriame na testovací dataset. Následne si v podkapitole 7.3 predstavíme výsledky testovania a tiež sa pozrieme na zaujímavejšie výstupy zo siete. Nakoniec popíšeme možnosti budúceho rozšírenia práce v podkapitole 7.4. Neurónová sieť bola trénovaná na syntetických odtlačkoch z generátorov opísaných v podkapitole 6.3. Testovanie prebiehalo na súbore reálnych odtlačkov.

Samotné testovanie bolo rozdelené na tri časti, pre každú triedu zvlášť. Vzorka sa skladala z 36 obrázkov, ktoré neboli poškodené hľadanými ochoreniami. 13 obrázkov s poškodeniami spôsobenými dyshidrózou a 14 obrázkov obsahujúcich prejavy bradavíc. Dôvodom takéhoto malého testovacieho setu je nedostatok reálnych dát s ochoreniami. Bola možnosť využiť dáta syntetické no tento krok by mohol výrazne deformovať reálnu presnosť, takže od neho bolo upustené.

7.1 Test plochy

Ako už bolo popísane v 7.1. Test sleduje plochu označenej oblasti na obrázkoch. V tabuľke 7.1 môžme vidieť výsledky týchto testov. Všetky údaje sú uvádzané ako percentuálny podiel oblasti. Pri tabuľke 7.1 je touto oblasťou plocha ochorenia v rámci odtlačku. V prípade tabuľky 7.2 sa jedná o celkovú detegovanú oblasť, ktorú vyznačila sieť.

Pri bradaviciach môžme vidieť vcelku vysoký podiel správne označenej plochy bradavíc. No tiež za povšimnutie stojí aj 15.92 % podiel zle označených oblastí. Po bližšom preskúmaní výsledkov zistíme, že sa obvykle jedná o okrajové oblasti bradavíc, ktoré sieť môže nesprávne označiť za prejavy dyshidrózy.

Naopak pri označovaní oblastí poškodených dyshidrózou vidíme až tretinový podiel, ktorý zostal neoznačený. Do tohto podielu sú obvykle zaradené časti bielych rýh, ktorými sa dyshidróza prejavuje. Ďalším častým prejavom, ktorý nieje obvykle nijak označený sú male biele šupinky. Do zle označených oblasti ktorá ma 19,27 % sú taktiež obvykle zaradené oblasti so slabším poškodením.

Tabuľka 7.1: Podiel úspešnosti detekcie na ploche ochorenia

	Bradavice	**Dyshidróza**
Správne označená oblasť	73,26 %	38,27 %
Zle označená oblasť	6,32 %	21,09 %
Oblasť bez označenia	20,42 %	40,64 %

Ďalšími nemenej podstatnými údajmi sú oblasti, ktoré sieť označila ako prejav ochorenia aj keď v popise k odtlačku tieto oblasti označené nie sú. Tabuľka 7.2, nám ukazuje percentuálny podiel z celkovo detegovanej plochy v rámci jednotlivých ochorení. Pri obidvoch ochoreniach je sem často zaradená okolitá oblasť ochorenia. V prípade dyshidrózy sa tu môžu objaviť aj vnútorne časti napríklad medzi dvoma rozsiahlejšími plochami. Ďalej do tohto podielu radíme aj prípady falošnej detekcie, kedy sieť z rôznych dôvodov deteguje na nepoškodenej oblasti danú chorobu. Tieto prípady nie je úplne možné odlíšiť len z výpočtu obsahu vyznačenej plochy, preto túto problematiku rieši test správneho určenia v podkapitole 7.2.

Tabuľka 7.2: Podiel oblasti mimo ochorenia z celkovej detegovanej oblasti

	Bradavice	**Dyshidróza**
Oblasť v rámci ochorenia	57,97 %	70,12 %
Oblasť mimo ochorenia	42,03 %	29,88 %

7.2 Test správneho určenia

Ďalším testom je test správneho určenia. Tento test sa nezameriava ani tak na veľkosť detegovanej oblasti ako skôr na príčinu detekcie. Ako už bolo popísane v 6.7.2, test sa zameriava na dôvod detekcie. Tak ako pri minulom teste aj tu použijeme rovnaký počet testovacích vzoriek a test bude tiež rozdelený podľa ochorenia. Výsledky testu môžme rozdeliť do 3 kategórií:

- Správne - v detegovanej oblasti sa skutočne nachádza prejav označeného ochorenia
- Nesprávna trieda - v detegovanej oblasti sa nachádza prejav iného ochorenia
- Nesprávne - v detegovanej oblasti sa nenachádza prejav žiadneho z hľadaných ochorení

Výsledky týchto testov sú predstavené v tabuľke 7.3 pre dyshidrózu a v tabuľke 7.4 pre bradavice. V tabuľke testujúcej dyshidrózu si môžme všimnúť vysoký podiel bradavíc v riadku „zlej triedy“. Tento jav je spôsobený práve nedostatkom dát v trénovacej množine a preto sú tieto oblasti nesprávne prezentované ako bradavice, čo má podiel na úspešnosti detekcie daného ochorenia.

Tabuľka 7.3: Úspešnosť detekcie z pohľadu dyshidrózy.

	Dyshidróza	**Bradavice**
Správne	99,91 %	0,00 %
Zlá trieda	0,00 %	34,89 %
Nesprávne	0,9 %	65,11 %

Na druhej strane v tabuľke 7.4, môžme vidieť dve kritické hodnoty. Tou prvou je riadok „zlá trieda“ kde dyshidróza dosahuje 84 percent. Toto číslo sa môže zdať vysoké, no veľká časť týchto detekcií je „prekrytá“ detekciami s vyššou pravdepodobnosťou ktoré už bradavicu v ochorení ukazujú celkom spoľahlivo. Druhým problémom je vysoký podiel nesprávne detegovaných bradavíc, je spôsobený rôznymi vonkajšími vplyvmi na odtlačok zvlášť nedostatočný prítlak na papier/senzor.

Tabuľka 7.4: Úspešnosť detekcie z pohľadu bradavíc.

	Dyshidróza	**Bradavice**
Správne	0,00 %	16,16 %
Zlá trieda	50,70 %	0,00 %
Nesprávne	49.30 %	83,84 %

Druhou časťou tohto testu je testovanie na odtlačkoch, ktoré nie sú zasiahnuté hľadanými ochoreniami. Test bol vykonávaný na 36 odtlačkoch. Tento test sleduje aký podiel z testovanej množiny sieť vyhodnotí ako odtlačky so známkami ochorenia. Aj v prípade keď sa na odtlačku ochorenie nenachádza. Ako môžme vidieť v tabuľke 7.5. Podiel falošnej detekcie je vcelku vysoký. Po bližšom preskúmaní zistíme, že program často deteguje objekty v okolí odtlačku, ako sú rôzne čiary, či písmená z podkladu. Ďalšou skupinou odtlačkov, ktoré sú falošne detegované sú tie so zle odtlačenými oblasťami.

Tabuľka 7.5: Falošne detekcie v čistých odtlačkoch. Veľkosť vzorky bola 36 odtlačkov.

	Dyshidróza	**Bradavice**
Podiel zle označených	38,89 % (14)	41,67 % (15)

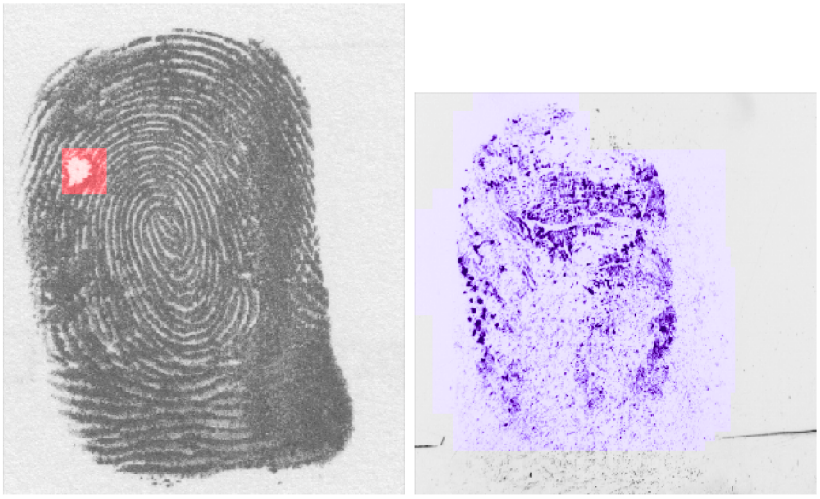
7.3 Výsledky

Ako budeme vidieť aj v jednotlivých častiach tejto podkapitoly, môžme prehlásiť, že sieť by bola oveľa úspešnejšia v detekcií keby máme dostatok reálnych vstupných dát. Výsledky siete sú teda závisle od dát generovaných. Tieto dáta sú síce v dostatočnej kvalite no nepokrývajú celý rozsah ochorení. Vylepšenie rozsahu vstupných dát by mohlo pomôcť zlepšiť celkovú detekciu ochorení.

V nasledujúcich častiach budú prestavené zaujímave výsledky. Ako prve budú predstavené ideálne výsledky. Nasledovať budu zaujímavé a časté problémy pri jednotlivých ochoreniach. Vo všetkých uvedených príkladoch platí že červenou sú označené bradavice a modrou dyshidróza.

7.3.1 Ideálne výsledky

Ideálne výsledky sú tie kde sa sieti správne podarilo priam ukážkovo označiť ochorenie. Ako príklad si môžme uviesť obrázok 7.1. V týchto prípadoch môžme vidieť ideálne zachytenú bradavicu aj keď sa jedná o jednu z najmenších v databáze. Tiež si môžme všimnúť, že sieť ohraničuje aj oblasti v okolí bradavice. To je spôsobene veľkosťou okna, ktoré dané ochorenie detegovalo. A tiež tu môžme vidieť dyshidrózou ťažko zasiahnutý odtlačok ktorý sieť pokryla skoro stopercentne. Celkovo platí, že sieť si lepšie poradí s dyshidrózou ťažko postihnutými odtlačkami.



Obr. 7.1: Príklady ideálnej detekcie a klasifikácie ochorení.

7.3.2 Problémy u bradavíc

Jedným z asi najzaujímavejších problémov spojených s bradavicami, sú menšie bradavice. Tieto bradavice zvláda sieť celkom obstojne identifikovať pokiaľ sú dobre ohraničené prípadne obsahujú male bodky vnútri. Ako to môžme vidieť na obrázku 7.2 kde bradavica bola vyhodnotená ako dyshidróza, naopak obdobná bradavica predstavená na obrázku v predchádzajúcej podkapitole je považovaná za ideálny výsledok.

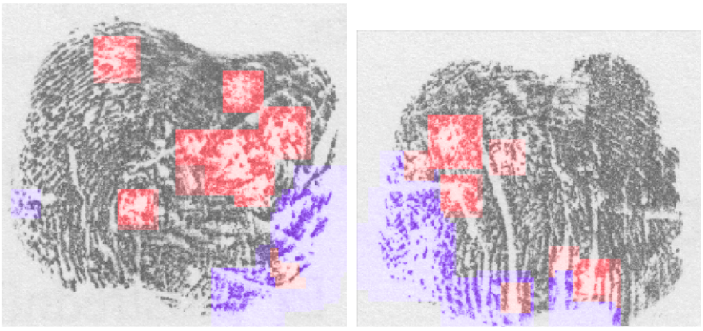
Obr. 7.2: Vľavo môžme vidieť prípad kde je bradavica na kraji, v pravo zas zle vyhodnotenú bradavicu.

Druhým zaujímavým prípadom sú bradavice umiestnené na okraji odtlačku prsta. Tento príklad môžme vidieť na obrázku 7.2 vľavo. S týmto stavom si sieť nieje schopná poradiť. Dôvodom je, že takýto stav sa vo vstupnej množine nenachádza nakoľko nám ho generátory neumožňujú vytvoriť.

7.3.3 Problémy u dyshidrózy

Problém kedy je dyshidróza klasifikovaná ako bradavica sa aj vo vo všeobecných výsledkoch ukazoval celkom často, príklady môžme vidieť na obrázku 7.3. Obvykle sa jedná o malú bielu plochu s pár tmavými bodmi uprostred. Ako už bolo spomenuté v podkapitole 7.1, tento problém je spôsobený nedostatkom takýchto stavov v trénovanej množine. Ďalším

pomerne častým problémom u dyshidrózy je neschopnosť siete detegovať biele čiary aj keď sú súčasťou trénovacej množiny. Tento pretrvávajúci problém môže byť predmetom ďalšieho rozširovania tejto práce.



Obr. 7.3: Príklady kedy boli menšie biele plochy zle vyhodnotené ako bradavice.

7.4 Možnosti budúceho rozšírenia

Práca poskytuje základ pre rozpoznávanie a detekciu ochorení na odtlačkoch prsta pomocou konvolučných neurónových sietí. Problémami ako už bolo mnoho krát spomenuté, zostávajú veľkosti vstupných reálnych databáz jednotlivých chorob, ktoré je potom nutné nahrádzať syntetickými odtlačkami s umelo generovanými ochoreniami.

Možnosťou rozšírenia práce by bolo spojenie s už existujúcimi technikami počítačového videnia. Ako je napríklad vyhľadávanie poškodených oblastí pomocou smerového poľa a následne prehľadávanie neurónovou sieťou. Ako možnosti ďalšieho vylepšenia práce sa ukazujú najmä:

- Rýchlosť a spotreba výkonu - algoritmus by mohol byť optimalizovaný a zrýchlený aj pre použitie na menej výkonných zariadeniach. Prípadne by mohol byť ušetrený čas pri prechádzaní odtlačkom.
- Vyhľadávanie potencionálne poškodených oblasti - odhaľovanie poškodených oblastí klasickými metódami a následná klasifikácia ochorenia neurónovou sieťou.
- Rozšírenie datasetov - rozšírenie existujúcich databáz s reálnymi odtlačkami.
- Vylepšenie generátorov - keďže tvorba databáz s reálnymi odtlačkami je cieľ na dlhšie obdobie, riešením môže byť vylepšenie generátorov ochorení s dôrazom čo na najväčšiu podobnosť.

Kapitola 8

Záver

Cieľom práce bolo navrhnutie a naimplementovanie detektoru a klasifikátoru prejavov kožných ochorení na odtlačkoch prstov za pomoci konvolučných neurónových sieti. Táto sieť mala byť schopná rozpoznávať ochorenia na reálnych odtlačkoch.

Hlavným problémom pri tvorbe tejto práce bol nedostatok vstupných dát pre trénovanie preto boli využite syntetické odtlačky, do ktorých boli následne generované prejavy hľadaných ochorení. Pre tento účel bola rozšírená predchádzajúca práca na simulovanie ochorení o generovanie dyshidrózy pričom boli využite znalosti z ďalších prác. Tiež bola pridaná automatická tvorba notácie pre syntetické prejavy ochorenia. Ďalej boli využite odtlačky z internej databázy STRaDe a NIST DB 4.

Bola preštudovaná problematika konvolučných neurónových sieti, vďaka ktorej bol ako predloha zvolený model konvolučnej siete VGG16. Z tohto modelu bola vytvorená zjednodušená verzia, ktorá tvorila jadro práce. Pre pomocné funkcie k programu bol vytvorený modul *TLmaker* pre generovanie výrezov, ktorý bol využitý pri trénovaní. Ako posledné bola vytvorená vyhodnocovacia funkcia, ktorá za pomoci netrénovaného modelu deteguje a klasifikuje poškodenia na odtlačkoch prstov. Poškodená oblasť je detegovaná pomocou pravdepodobnosti ktorá sa objaví na výstupnej vrstve siete po vložení oblasti na jej vstupnú vrstvu. Vybraná je tá trieda ktorej pravdepodobnosť prekročí zvolený prah a samozrejme je najvyššia. Možnými rozšírením práce môže byť zlepšenie rýchlosti detekcie, rozšírenie databáz, prípadne nástrojov na generovanie syntetických prejavov ochorenia.

Vo výsledkoch sa ukázalo, že sieť vcelku dobre zvláda detekciu ťažšieho rozsahu dyshidrózy no problém jej robia menšie oblasti, ktoré sieť nieje schopná úplne zachytiť. Pripadne im priradí chybne označenie. Tento problém bude pravdepodobne spôsobený nedostatkom dát kde dyshidróza nemá tak ťažký priebeh.

Ohľadne situácie s bradavicami je situácia trochu odlišná. Sieť si vie dobre poradiť s klasickými prejavmi tohto ochorenia no problém jej robia menšie male bradavice. Prekvapivo si vie vcelku dobre poradiť s veľkými bradavicami a tiež celkom dobre zvláda ohraničiť bradavice ktoré boli vďaka atramentovej metóde rozmazané.

Literatúra

[1] Al Masri, A. *What Are Overfitting and Underfitting in Machine Learning?* Towards Data Science, Jun 2019. [Online; navštívené 26. Máj, 2020]. Dostupné z: https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-learning-a96b30864690.

[2] Bárta, M. *Generování onemocnění kůže do syntetických otisků prstů.* Brno, CZ, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: https://www.fit.vut.cz/study/thesis/17348/.

[3] Bottou, L. *Stochastic Gradient Descent Tricks.* Neural Networks, Tricks of the Trade, Reloaded. Springer, January 2012. 430-445 s. Lecture Notes in Computer Science (LNCS). Dostupné z: https://www.microsoft.com/en-us/research/publication/stochastic-gradient-tricks/.

[4] Brownlee, J. *A Gentle Introduction to the Rectified Linear Unit (ReLU).* Machine Learning Mastery Pty. Ltd., Aug 2019. Dostupné z: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

[5] Bruballa, R. G. *Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names.* May 2018. [Online; navštívené 27. Máj, 2020]. Dostupné z: https://gombru.github.io/2018/05/23/cross_entropy_loss/.

[6] Che, Z.-G. Feed-forward neural networks training: a comparison between genetic algorithm and back-propagation learning algorithm. 2011. Dostupné z: https://pdfs.semanticscholar.org/82c1/72382f2c98567b30ebf223f25cfb715c1f28.pdf.

[7] Chollet, F. et al. *Keras documentation: Developer guides* [https://keras.io]. 2020. [Online; navštívené 28. Marca, 2020].

[8] Chow, C. a Kaneko, T. Automatic boundary detection of the left ventricle from cineangiograms. *Computers and Biomedical Research.* 1972, zv. 5, č. 4, s. 388 – 410. DOI: https://doi.org/10.1016/0010-4809(72)90070-5. ISSN 0010-4809. Dostupné z: http://www.sciencedirect.com/science/article/pii/0010480972900705.

[9] Davies, E. R. *Computer and Machine Vision, Fourth Edition: Theory, Algorithms, Practicalities.* 4th. USA: Academic Press, Inc., 2012. ISBN 0123869080.

[10] Drahanský, M. *Biometric security systems fingerprint recognition technology.* Saarbrücken: Vědecké spisy Vysokého učení technického v Brně, 2005. ISBN 80-214-2969-0.

[11] Drahanský, M. *Hand-Based Biometrics*. The IET, 2018. ISBN 978-1-78561-224-4.

[12] Du Vivier, A. a McKee, P. *Atlas of clinical dermatology*. Oxford: Elsevier Saunders, 2013. ISBN 978-0-7020-3421-3.

[13] Goodfellow, I., Bengio, Y. a Courville, A. *Deep Learning*. MIT Press. Dostupné z: http://www.deeplearningbook.org.

[14] Haykin, S. S. *Neural networks and learning machines*. 3. vyd. Upper Saddle River, NJ: Pearson Education, 2009.

[15] Häggström, M. *Layers of the epidermis*. 2010. [Online; navštívené 28. Januára, 2020]. Dostupné z: https://commons.wikimedia.org/wiki/File:Epidermal_layers.svg.

[16] Jackson, A. R. W. a Jackson, J. M. *Forensic science*. 3rd edition. New York : Prentice Hall, 2011. ISBN 0273738402 (pbk.).

[17] Jirásková, M. Bradavice – věčný problém a co s nimi. *Interní medicína pro praxi*. 2009. Dostupné z: https://www.solen.cz/pdfs/int/2009/12/11.pdf.

[18] Kanich, O. *Fingerprint damage simulation: a simulation of fingerprint distortion, damaged sensor, pressure and moisture*. Saarbrücken: Lambert academic publishing, 2014. ISBN 978-3-659-63942-5.

[19] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. Taipei, Taiwan. 1998, zv. 86, č. 11, s. 2278–2324. Dostupné z: http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf.

[20] Maltoni, D. *Handbook of fingerprint recognition*. London: Springer, 2009. ISBN 978-1-84882-253-5.

[21] Munakata, T. *Fundamentals of the New Artificial Intelligence: Neural, Evolutionary, Fuzzy and More (Texts in Computer Science)*. 2nd ed. Springer Publishing Company, Incorporated, 2008. ISBN 184628838X.

[22] Orczyk, T. a Wieclaw, L. Fingerprint ridges frequency. In: *2011 Third World Congress on Nature and Biologically Inspired Computing*. Oct 2011, s. 558–561. DOI: 10.1109/NaBIC.2011.6089649. ISSN null.

[23] Ruder, S. *An overview of gradient descent optimization algorithms*. 2016.

[24] Simonyan, K. a Zisserman, A. Very deep convolutional networks for large-scale image recognition. *ArXiv preprint arXiv:1409.1556*. 2014. Dostupné z: https://arxiv.org/pdf/1409.1556.pdf.

[25] Sonka, M., Hlavac, V. a Boyle, R. *Image Processing: Analysis and Machine Vision*. 2. vyd. CL-Engineering, 1998. ISBN 053495393X.

[26] Suzuki, S. a be, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*. 1985, zv. 30, č. 1, s. 32 – 46. DOI: https://doi.org/10.1016/0734-189X(85)90016-7. ISSN 0734-189X. Dostupné z: http://www.sciencedirect.com/science/article/pii/0734189X85900167.

[27] Svoradová, V. *Generování onemocnění kůže do syntetických otisků prstů z SFinGe.* Brno, CZ, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: https://www.fit.vut.cz/study/thesis/21893/.

[28] Walia, A. S. *Activation functions and it's types-Which is better?* [online]. 2017. [Online; navštívené 7. Januára, 2020]. Dostupné z: https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f.

[29] Watson, C. I. a Wilson, C. L. NIST special database 4. *Fingerprint Database, National Institute of Standards and Technology.* Citeseer. 1992, zv. 17, č. 77, s. 5.

[30] Win, K. N., Li, K., Chen, J., Viger, P. F. a Li, K. Fingerprint classification and identification algorithms for criminal investigation: A survey. *Future Generation Computer Systems.* 2019. DOI: https://doi.org/10.1016/j.future.2019.10.019. ISSN 0167-739X. Dostupné z: http://www.sciencedirect.com/science/article/pii/S0167739X19315109.

[31] Šetelíková, A. *Pojem a podstata daktyloskopie.* 2012. Diplomová práca. Univerzita Karlova v Praze.

[32] Štork, J. *Dermatovenerologie.* Saarbrücken: Galén, 2013. ISBN 978-80-7262-898-8.