



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**ZPRACOVÁNÍ SNÍMKU Z MEIBOGRAFU A DETEKCE
MAZOVÝCH/MEIBOMSKÝCH ŽLÁZ**

MEIBOGRAF IMAGE PROCESSING AND DETECTION OF SEBACEOUS/MEIBOMIAN GLANDS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL SANDANUS

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ŠTĚPÁN RYDLO,

BRNO 2023

Zadání diplomové práce



146445

Ústav: Ústav inteligentních systémů (UITS)
Student: **Sandanus Michal, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Strojové učení
Název: **Zpracování snímku z meibografu a detekce mazových/meibomských žláz**
Kategorie: Zpracování obrazu
Akademický rok: 2022/23

Zadání:

1. Seznamte se s možnostmi způsobů pozorování slzných žláz na víčku oka a metod používaných pro jejich zobrazení a snímání.
2. Následně prostudujte možnosti zpracování obrazu z videosekvence pro zvýraznění slzných žláz. Zaměřte se na metody strojového učení pro určení identifikace oblasti zájmu.
3. Navrhněte metodu pro detekci oblasti očního víčka z nasnímané videosekvence s následným zvýrazněním a detekcí slzných žláz.
4. Výše uvedenou metodu implementujte a vytvořte aplikaci umožňující zpracovávat již nahrané videosekvence pro zobrazení a zvýraznění výsledků v obraze. Druhý režim činnosti aplikace bude umožňovat analýzu a zobrazování vstupu z kamery v reálném čase.
5. Otestujte aplikaci, zhodnoťte dosažené výsledky a případně navrhněte další možná rozšíření.

Literatura:

- Wise RJ, Sobel RK, Allen RC. Meibography: A review of techniques and technologies. Saudi J Ophthalmol. 2012 Oct;26(4):349-56. doi: 10.1016/j.sjopt.2012.08.007.
- ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing, analysis, and machine vision*. 4th edition. United States of America: Cengage Learning, 2015, xxxv, 870 stran : ilustrace, grafy ; 23 cm. ISBN 978-1-133-59369-0.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodu 1 až 3.

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

Vedoucí práce: **Rydlo Štěpán, Ing.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1.11.2022
Termín pro odevzdání: 17.5.2023
Datum schválení: 3.11.2022

Abstrakt

Táto práca popisuje meibomove žľazy, metódy ich pozorovania a zobrazenia. Predstavuje súčasný stav poznania v oblasti segmentácie očného viečka a meibomových žliaz a na jeho základe je vytvorený funkčný prototyp aplikácie pre riešenie danej problematiky metódami konvolučných neurónových sietí. Segmentácia je vykonávaná 19-vrstvovou adaptáciou architektúry U-Net a architektúrou MobileNetV3-large so segmentačnou hlavicou LR-ASPP. Spojením týchto architektúr boli dosiahnuté hodnoty, ktoré v určitých metrikách presahujú výsledky súvisiacich prác. Aplikácia je implementovaná ako desktopová aplikácia vo frameworku .NET 7.0 s použitím knožnice OpenCV.

Abstract

This thesis describes meibomian glands, methods of their observation and imaging. The current state-of-the-art in the field of segmentation of the eyelid and meibomian glands is presented, and based on it, a functional prototype of the application for solving the given problem using the methods of convolutional neural networks is created. Segmentation is performed by a 19-layer adaptation of the U-Net architecture and the MobileNetV3-large architecture with the LR-ASPP segmentation head. By combining these architectures, values that exceed the results of related works in certain metrics were achieved. The application is implemented as a desktop application in .NET 7.0 framework using OpenCV library.

Klíčové slová

meibografia, meibomove žľazy, strojové učenie, konvolučné neurónové siete, detekcia, segmentácia

Keywords

meibography, meibomian glands, machine learning, convolutional neural networks, detection, segmentation

Citácia

SANDANUS, Michal. *Zpracování snímku z meibografu a detekce mazových/meibomských žláz*. Brno, 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Štěpán Rydlo,

Zpracování snímku z meibografu a detekce mazových/meibomských žláz

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Štěpána Rydla. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....

Michal Sandanus

14. mája 2023

Podakovanie

Rád by som sa poďakoval vedúcemu mojej práce Ing. Štěpánovi Rydlovi za rady pri písaní tejto práce.

Obsah

1	Úvod	5
2	Slzné žľazy a metódy ich snímania a detekcie	7
2.1	Meibomove žľazy	7
2.2	Metódy evaluácie meibomových žliaz	9
2.3	Meibografia	10
2.4	Súvisiace práce	10
3	Neurónové siete určené na segmentáciu obrazových dát	14
3.1	Základné princípy neurónových sietí	14
3.2	Konvolučné neurónové siete	18
3.3	Metriky úspešnosti segmentačných neurónových sietí	19
3.4	Relevantné architektúry konvolučných neurónových sietí	20
4	Návrh riešenia	25
4.1	Špecifikácia kamery použitej pre účely tejto práce	25
4.2	Dátová sada	26
4.3	Modely neurónových sietí	28
4.4	Popis funkcionality aplikácie	31
5	Implementácia	35
5.1	Použité hardvérové a softvérové prostriedky	35
5.2	Tvorba modelov neurónových sietí	36
5.3	Aplikácia s používateľským rozhraním	40
6	Evaluácia	47
6.1	Evaluácia segmentácie očného viečka	47
6.2	Evaluácia segmentácie meibomových žliaz	50
6.3	Porovnanie s existujúcimi prácami	52
6.4	Zhodnotenie funkčnosti aplikácie	54
7	Záver	56
	Literatúra	57
A	Prehľad vlastností aplikácie	61
A.1	Používateľské rozhranie aplikácie	61
A.2	Používateľská príručka aplikácie	65

B	Podrobné výsledky tréovania neurónových sietí	68
C	Obsah pamäťového média	76

Zoznam obrázkov

2.1	Štruktúra očných viečok z ich vnútornej strany [7]	7
2.2	Zobrazenie meibomových žliaz na snímkach z meibografie [32]	8
2.3	Ukážka štruktúry meibomových žliaz na snímkach v závislosti od meiboskóre [1]	9
2.4	Použitie meibografického systému od Arita a spol. [3]	10
2.5	Metodika detekcie meibomových žliaz použitá v práci od Koha a spol.[18] .	11
2.6	Reprezentácia modelov neurónových sietí v práci [32]	12
3.1	Usporiadanie neurónov v neurónovej sieti	15
3.2	Aplikácia kernelového jadra na vstupnú maticu	18
3.3	Ukážka transponovanej konvolúcie [41].	21
3.4	Architektúra plne prepojenej siete z práce [23]. Červenou sú znázornené konvolučné vrstvy, zelenou združovacie, modrou vrstvy transponovanej konvolúcie s faktorom nadvzorkovania a žltá vrstva predstavuje softmax funkciu. Obrázok bol prebraný z práce [10].	21
3.5	Architektúra segmentačnej konvolučnej neurónovej siete U-Net [31].	22
3.6	Rozširovanie výstupu prostredníctvom <i>unpoolingu</i> [10].	23
3.7	Algoritmus detekcie objektov architektúrou R-CNN [12].	23
3.8	Klasická konvolúcia s kernelom veľkosti 3x3 znázornená červenou farbou v porovnaní s dilatovanou konvolúciou s kernelom o rozmeroch 3x3 a hodnotou dilatácie 1 znázornená zelenou [10].	24
4.1	Kamera použitá na snímanie očného viečka	26
4.2	Príklad snímky horného viečka z dátovej sady MGD-1K	26
4.3	Príklad snímky dolného viečka z dátovej sady MGD-1K	27
4.4	Porovnanie výstupu z prototypu kamery a snímky dátovej sady MGD-1K .	28
4.5	Architektúra segmentačnej neurónovej siete U-Net použitá v práci od Prabhua a spol.[28]	29
4.6	Architektúra segmentačnej neurónovej siete s kostrou MobileNetV3 a segmentačnou hlavicou LR-ASPP[16]	29
4.7	Obrazovky aplikácie a prechody medzi nimi	31
4.8	Obrazovka analýzy uloženej snímky	32
4.9	Obrazovka analýzy videa v reálnom čase	33
5.1	Súborová štruktúra tréningových dát - segmentácia meibomových žliaz	36
5.2	Vývoj tréningových metrík počas sietí U-Net a MobileNetV3 - segmentácia očného viečka	38
5.3	Vývoj tréningových metrík počas sietí U-Net a MobileNetV3 - segmentácia meibomových žliaz	39

5.4	Diagram obrazoviek a ich štruktúry. Prvky s názvami končiacimi <i>Window</i> sú typu <i>ContentPage</i> , prvky končiace <i>View</i> sú typu <i>ContentView</i> . Skratka <i>SVV</i> predstavuje <i>SideVideoView</i> , skratka <i>IV</i> znamená <i>ImageView</i>	41
5.5	Obrazovka analýzy uloženej snímky (<i>SavedImageWindow</i>)	41
5.6	Obrazovka analýzy uloženého videa (<i>SavedVideoWindow</i>)	42
5.7	Obrazovka analýzy videa v reálnom čase (<i>RealTimeVideoWindow</i>)	42
6.1	Segmentácia očného viečka adaptáciou architektúry U-Net s úpravami - najlepší a najhorší výstup	48
6.2	Segmentácia očného viečka architektúrou MobileNetV3 s úpravami - najlepší a najhorší výstup	48
6.3	Percentuálne rozdiely v obsahu vysegmentovanej oblasti jednotlivých prístupov a architektúr sietí v porovnaní so segmentačnou maskou určenou lekárom	49
6.4	Segmentácia meibomových žliaz adaptáciou architektúry U-Net s prekryvom MobileNetV3 - najlepší a najhorší výstup	50
6.5	Segmentácia meibomových žliaz architektúrou MobileNetV3 s prekryvom MobileNetV3 - najlepší a najhorší výstup	51
6.6	Percentuálne rozdiely v obsahu vysegmentovanej oblasti jednotlivých prístupov a architektúr sietí v porovnaní so segmentačnou maskou určenou lekárom	51
6.7	Hodnoty poklesu meibomových žliaz v testovacej časti dátovej sady	53
6.8	Analýza snímky horného viečka z prototypu kamery	54
6.9	Analýza snímky dolného viečka z prototypu kamery	54
A.1	Domovská obrazovka	61
A.2	Obrazovka výberu z uložených dát	62
A.3	Obrazovka analýzy uloženej snímky	62
A.4	Obrazovka analýzy uloženého videa	63
A.5	Obrazovka identifikácie pacienta	63
A.6	Obrazovka analýzy videa v reálnom čase	64
A.7	Obrazovka správy filtrov	64
A.8	Obrazovka nastavení	65
A.9	Obrazovka O aplikácii	65
B.1	Vývoj hodnôt stratovej funkcie na tréningových dátach	68
B.2	Vývoj hodnôt stratovej funkcie na validačných dátach	69
B.3	Vývoj hodnôt miery prekryvu na validačných dátach	69
B.4	Vývoj hodnôt stratovej funkcie na tréningových dátach	70
B.5	Vývoj hodnôt stratovej funkcie na validačných dátach	71
B.6	Vývoj hodnôt miery prekryvu na validačných dátach	71
B.7	Vývoj hodnôt stratovej funkcie na tréningových dátach	72
B.8	Vývoj hodnôt stratovej funkcie na validačných dátach	73
B.9	Vývoj hodnôt miery prekryvu na validačných dátach	73
B.10	Vývoj hodnôt stratovej funkcie na tréningových dátach	74
B.11	Vývoj hodnôt stratovej funkcie na validačných dátach	75
B.12	Vývoj hodnôt miery prekryvu na validačných dátach	75

Kapitola 1

Úvod

Meibomove žľazy majú za úlohu sekréciu lipidov do slzného filmu, čím zabraňujú jeho excesívnemu odparovaniu. Dysfunkcia meibomových žliaz je najčastejšou príčinou syndrómu suchého oka a je spojená s morfológickými zmenami mazových žliaz, ktoré môžu viesť k zníženiu produkcie lipidov, či zablokovaniu kanálikov určených na ich vylučovanie. Vysušovanie oka narušuje ostrosť videnia a spôsobuje očný diskomfort a ďalšie symptómy vedúce k zmene kvality života.

Na skúmanie funkčnosti a štruktúry meibomových žliaz sa používajú viaceré techniky, mnohé z nich však žľazy analyzujú iba prostredníctvom nepriamych faktorov. Najúčinnjšou priamou metódou je meibografia. Ide sa o neinvazívnu zobrazovaciu techniku založenú na princípe snímania obrazu v infra-červenom spektre bez spôsobovania väčšieho diskomfortu pacientovi. Výsledné snímky alebo videosekvencia sú však pre oftamológov ťažko čitateľné a vzniká potreba automatizovaného vizuálneho odlišenia meibomových žliaz od okolitého tkaniva očného viečka.

V súčasnosti sa okrem transformácií obrazu zmenami kontrastu, jasú, expozície, či aplikáciou rozličných filtrov používajú na sémantickú segmentáciu očného viečka a meibomových žliaz, či určenie stupňa poškodenia metódy založené na konvolučných neurónových sieťach. Tie dosahujú výsledky porovnateľné s profesionálnymi oftamológmi, pričom skracujú čas segmentácie a objektivizujú hodnotenie stupňa poškodenia žliaz.

Cielom tejto práce bolo preskúmať súčasný stav poznania v tejto oblasti a na základe analýzy vytvoriť funkčný prototyp aplikácie umožňujúcej segmentáciu očného viečka a meibomových žliaz z nasnímanej videosekvencie a rovnako analýzu meibografického videa v reálnom čase. Pre tieto účely sú v práci navrhnuté architektúry neurónových sietí U-Net a MobileNetV3_large so segmentačnou hlavicou LR_ASPP na analýzu už nasnímaných videosekvencií a algoritmy určené na transformácie obrazu vo videu v reálnom čase. Dosiahnuté výsledky sú na úrovni súčasného stavu poznania a implementované v používateľsky zrozumiteľnej aplikácii.

V práci sa v kapitole 2 najskôr zaoberám analýzou slzných žliaz a metódami ich snímania z medicínskeho uhlu pohľadu. Predstavené sú aj metódy segmentácie slzných žliaz a očného viečka použité v súvisiach prácach. Kapitola 3 je venovaná analýze konvolučných neurónových sietí s dôrazom na architektúry a algoritmy určené na segmentáciu obrazových dát. Ďalej je v kapitole 4 predstavený návrh finálneho riešenia aplikácie a neurónových sietí aj s dostupnou dátovou sadou pre tréning. Okrem toho je stručne predstavený aj prototyp kamery, ktorý bol použitý pre účely tejto práce. Implementačné detaily sú popísané v kapitole 5 a na záver je v kapitole 6 vykonaná evaluácia natrénovaných modelov neuróno-

vých sietí spolu s porovnaním s podobnými prácami a zhodnotením funkčnosti vytvorenej aplikácie.

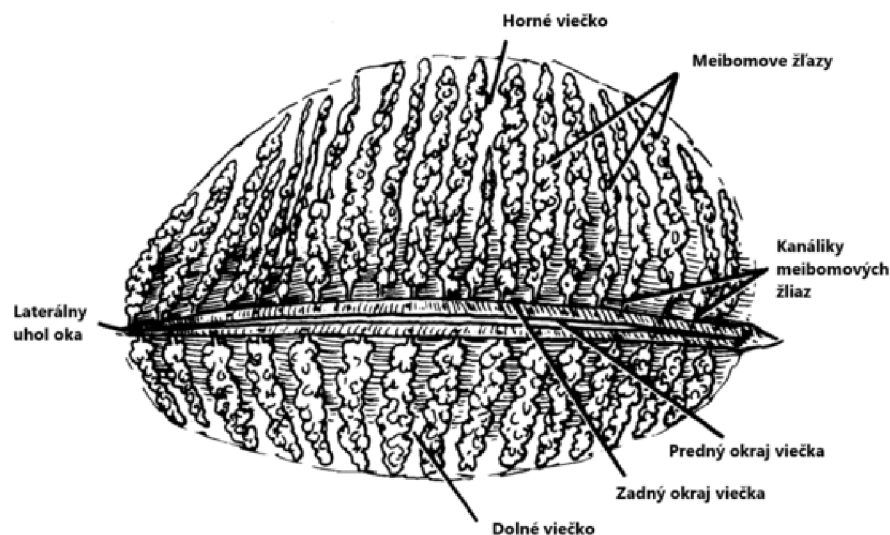
Kapitola 2

Slzné žľazy a metódy ich snímania a detekcie

Táto kapitola sa zameriava na uvedenie do problematiky slzných, alebo inak nazývaných meibomových žliaz a porúch ich fungovania z medicínskeho uhlu pohľadu. Následne sú popísané metódy na odhaľovanie týchto porúch s dôrazom na techniku meibografie a nakoniec sú analyzované postupy použité v prácach zaoberajúcich sa automatizovanou segmentáciou meibomových žliaz a očného viečka.

2.1 Meibomove žľazy

Oko je jedným z najdôležitejších zmyslových orgánov pre človeka. Odhaduje sa, že až okolo 80% nášho vnímania, učenia a poznania je uskutočňovaného prostredníctvom zraku.[30] Pre udržiavanie správnej funkčnosti musí byť oko neustále hydratované, k čomu prispievajú aj meibomove žľazy. Tie zohrávajú významnú úlohu v procese tvorby slz tým, že produkujú lipidy pre tukovú vrstvu slzného filmu.[39]



Obr. 2.1: Štruktúra očných viečok z ich vnútornej strany [7]

Funkčnou jednotkou meibomových žliaz sú meibocyty. Ide sa o mazové bunky, ktoré sú usporiadané do zhlukov v rámci žlazových váčkov. Meibocyty syntetizujú lipidy nazývané meibum a prostredníctvom kanálikov meibomových žliaz ich vylučujú do slzného filmu. Meibum pokrýva povrch slznej vrstvy, kde zabraňuje odparovaniu slz, a teda vysychaniu oka. Meibum má rovnako aj lubrikačnú funkciu a zabraňuje tak podráždeniu povrchu oka, ktoré by mohlo vzniknúť napríklad žmurkaním alebo inými vonkajšími vplyvmi. Práve žmurknutie oka zabezpečuje distribúciu slznej tekutiny po celej ploche oka, a tak aj jeho ochranu.[39]



Obr. 2.2: Zobrazenie meibomových žliaz na snímkach z meibografie [32]

Meibomove žľazy sú umiestnené v rade pozdĺžne vedľa seba a kolmo na okraj očného viečka z jeho vnútornej strany, ako je možné vidieť na obrázkoch 2.1 a 2.2. V zdravom oku sa v hornom viečku nachádza približne 30 až 40 meibomových žliaz, v dolnom 20 až 30. Žľazy v hornom viečku sú dlhšie, majú dĺžku približne 7 mm, zatiaľ čo žľazy v dolnom viečku merajú približne 4 mm.[7]

Dysfunkcia meibomových žliaz sa definuje ako chronická abnormalita meibomových žliaz, zvyčajne charakterizovaná upchaním koncových kanálikov a/alebo kvalitatívnymi/kvantitatívnymi zmenami v žlazovej sekrécii.[27] Podľa množstva vylučovaných lipidov môžeme dysfunkciu deliť na hyposekrečnú alebo hypersekrečnú.[39]

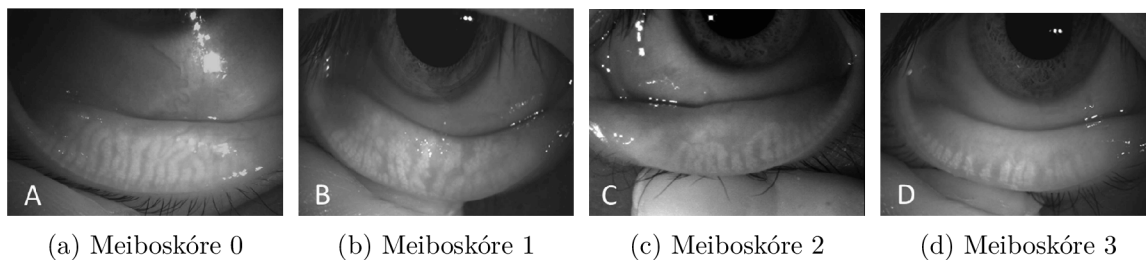
Poruchy fungovania meibomových žliaz môžu vyústiť do zmeny zloženia slzného filmu, symptómov podráždenia oka, zápalu alebo ochorenia očného povrchu. Dysfunkcia meibomových žliaz je najčastejšou príčinou syndrómu suchého oka v dôsledku odparovania slz a v oftamologickej praxi je pomerne častá. Najčastejšie je možné sa s ňou stretnúť v ázijskej populácii, kde je rozšírenie v rozmedzí 46,2 - 69,3%. V populácii kaukazského typu sa vyskytuje u 3,5 - 19,9% jedincov.[39]

Stupnica	Stupeň	Popis	
Meiboskóre	0	Bez poklesu množstva žliaz	
	1	Pokles menej ako jednu tretinu z celkovej plochy oblasti žliaz	
	2	Pokles medzi jednou tretinou a dvomi tretinami z celkovej plochy oblasti žliaz	
	3	Pokles viac ako dve tretiny z celkovej plochy oblasti žliaz	
Meiboznámka	0	Bez očividných zmien v očnom viečku	V každej z kategórií: kľukatenie, skrátenie, zníženie počtu žliaz
	1	Zmeny v menej ako 33% očného viečka	
	2	Zmeny v 33 - 66% očného viečka	
	3	Zmeny vo viac ako 66% očného viečka	

Tabuľka 2.1: Porovnanie stupníc meiboskóre a meiboznámka [36]

Dysfunkcia meibomových žliaz sa štruktúrne prejavuje najmä dilatáciou ich kanálikov, zhrubnutím, kľukatením, či poklesom množstva meibomových žliaz.[39] Ako metodický prístup pre kvantitatívne určenie morfológie žliaz sa najčastejšie používajú meiboskóre[2]

alebo meiboznámka[5]. Popis určovania stupňa poškodenia podľa spomínaných stupníc sa nachádza v tabuľke 2.1.



Obr. 2.3: Ukážka štruktúry meibomových žliaz na snímkach v závislosti od meiboskóre [1]

2.2 Metódy evaluácie meibomových žliaz

Existujú mnohé techniky na priame, či nepriame určenie štruktúry a funkčnosti meibomových žliaz. Niektoré z nich môžu byť použité *in vivo*, teda priamym pozorovaním časti oka alebo očného viečka človeka, iné *ex vivo*, čo najčastejšie znamená odobratím vzorky a jej následnou analýzou mimo tela človeka. Toto sú najpoužívanejšie techniky na evaluáciu meibomových žliaz[39]:

- **Meranie produkcie slz a miery odparovania** - na meranie produkcie slz sa používajú dva testy. Prvým je Schirmerov test, ktorý je invazívnym testom. Druhým je test na čas rozpadu slzného filmu, ktorý predstavuje neinvazívny ekvivalent, ale ani jedným z týchto testov nie je možné spoľahlivo diagnostikovať dysfunkciu meibomových žliaz.
- **Biomikroskopické vyšetrenie** - bežne používaná neinvazívna metóda na nepriame skúmanie štruktúry a funkčnosti meibomových žliaz skúmaním zmien v oblasti očného viečka.
- **Zavedenie kontrastnej látky** - kontrastná látka slúži predovšetkým na odlíšenie jednotlivých vrstiev slzného filmu, čím je možné nepriamo diagnostikovať abnormality spojené s dysfunkciou meibomových žliaz.
- **Interferometria lipidovej vrstvy** - obrazová technika zameraná na meranie stability slz skúmaním ich povrchových kontúr a hrúbky lipidovej vrstvy.
- **Osmolarita slz** - meranie množstva rozpustených látok v slzách, ktoré sa zvyšuje v prípade nedostatočnej tukovej vrstvy v slznom filme v dôsledku odparovania.
- **Meibometria** - technika vyvinutá na meranie množstva meiba na okraji očných viečok odobratím vzorky na špeciálnu pásku a jej následnou fotometrickou analýzou.
- **Meibografia** - zobrazovacia technika vyvinutá na priame skúmanie štruktúry meibomových žliaz vizualizáciou ich morfológie.

Meibografia je jedinou z vyššie spomenutých metód, ktorá priamo skúma štruktúru meibomových žliaz, a teda poskytuje najlepšie možnosti diagnostiky ich dysfunkcie. Problematika meibografie je priblížená v sekcii 2.3.

2.3 Meibografia

História meibografie sa začína v roku 1977, kedy bolo pri biomikroskopickom vyšetrení použité ultra-fialové svetlo na zvýraznenie kanálikov meibomových žliaz a infra-červené svetlo na osvetlenie meibomových žliaz na infra-červenej fotografii.[39] V roku 1991 bola prvýkrát meibografia opísaná ako infra-červená fotografia meibomových žliaz v práci od Mathersa a spol.[25] V roku 1994 bola predstavená infra-červená meibografia s použitím videa a tento typ meibografie je najrozšírenejším dodnes. S ďalším vývojom postupne prišli nové metódy bez použitia infra-červených technológií, ako napríklad laserová konfokálna meibografia, alebo tomografická meibografia optickej koherencie, ktorá umožňuje aj vytváranie trojrozmerných snímok meibomových žliaz.[39]



(a) Snímanie dolného viečka

(b) Snímanie horného viečka

Obr. 2.4: Použitie meibografického systému od Arita a spol. [3]

Existujú dve meibografické techniky - kontaktná a bezkontaktná. Pri kontaktnej meibografii je nutné využiť svetelnú sondu, ktorá slúži na prevrátenie a presvetlenie očného viečka, aby mohli byť následne špeciálnou kamerou vytvorené jeho snímky.[39] To však pacientovi spôsobuje značný diskomfort. Bezkontaktná meibografia bola uvedená v roku 2008 v článku od Arita a spol.[2] Využíva sa pri nej biomikroskop so štrbinovou lampou a infra-červenou kamerou. V roku 2012 bol vytvorený aj neinvazívny a prenosný meibografický systém (jeho použitie je znázornené na obrázku 2.4), ktorý používa luminiscenčnú diódu s vlnovou dĺžkou 940 nm a je jednoduchý na manipuláciu.[3]

2.4 Súvisiace práce

Častokrát sa úlohy segmentácie očného viečka a detekcie meibomových žliaz zo snímok infra-červenej meibografie riešia rovnakými/podobnými postupmi, alebo v rámci jedného kroku. V prevažnej časti prác sú vykonané kroky segmentácie viečka a žliaz ako príprava pre konečné určenie hodnoty buď meiboskóre alebo meiboznámky, prípadne na zhodnotenie morfológie, či počtu meibomových žliaz.

V súčasnosti sa na riešenie danej problematiky používajú najmä metódy založené na konvulučných neurónových sieťach, ale predsa sa nájdu aj riešenia bez ich použitia. Oba prístupy aj so stručným rozborom relevantných prác sa nachádzajú v nasledujúcich sekciách.

Metódy bez použitia neurónových sietí

Prác, ktoré na riešenie problematiky segmentácie očného viečka a meibomových žliaz nevyužívajú neurónové siete nie je mnoho a väčšina z nich bola publikovaná pred viac ako desiatimi rokmi. Najnovšia práca s takýmto prístupom je od Xiaoa a spol.[40] Práca sa zaoberá okrem segmentácie spomínaných oblastí aj analýzou morfológie meibomových žliaz pre každú žlazu zvlášť. Na segmentáciu sú použité metódy zmien kontrastu a redukcie šumu v obraze, ktorých varianty sa postupne striedajú a aplikujú na meibografické snímky. Algoritmus dosiahol pre segmentáciu očného viečka v metrike index podobnosti s manuálne anotovanými dátami hodnotu 0.94 ± 0.02 a 0.87 ± 0.01 v úlohe segmentácie meibomových žliaz.



Obr. 2.5: Metodika detekcie meibomových žliaz použitá v práci od Koha a spol.[18]

Práca od Koha a spol.[18] rieši problematiku detekcie a klasifikácie meibomových žliaz do dvoch tried - zdravé a poškodené. Detekcia je vykonávaná formou nájdenia stredovej línie žľazy. Použitá metodika je zhrnutá v diagrame 2.5.

V práci od Celika a spol.[6] je detekcia meibomových žliaz vykonávaná pomocou Gaborových filtrov binárnej skeletonizácie. Získajú sa tak črty ako orientácia žliaz, hrúbka, dĺžka, či pokrytie. Algoritmus detekuje pixely stredovej línie žliaz s úspešnosťou 92%.

Metódy založené na neurónových sieťach

Sémantickú segmentáciu objektov z obrazových dát pomocou neurónových sietí je možné vykonať dvomi spôsobmi, a to vytvorením ohraničenia v tvare obdĺžnika (anglicky *bounding box*), v ktorom sa objekt nachádza, alebo klasifikovaním každého pixela obrazu podľa príslušnosti danému objektu. Vzhľadom na morfológiu meibomových žliaz a potrebu ich presného odlíšenia od okolitého tkaniva sa využíva výlučne postup klasifikácie jednotlivých pixelov. Pre tento účel sa používajú konvolučné neurónové siete vo všetkých nižšie spomínaných prácach.

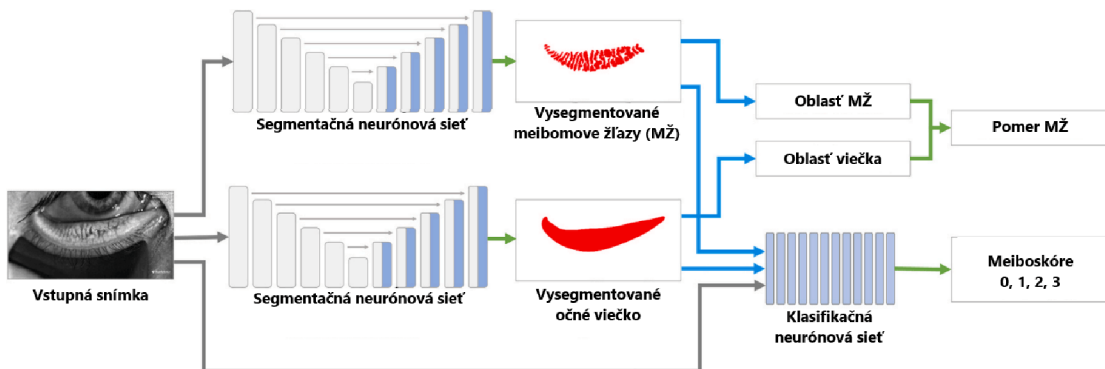
V práci od Prabhua a spol.[28] je použitá adaptácia architektúry U-Net s celkovo 19 konvolučnými vrstvami na segmentáciu meibomových žliaz. Pre účely tejto práce bol vytvorený dataset snímok prevráteného očného viečka, ktorý obsahoval 400 snímok zhotovených prototypovou mobilnou kamerou a 400 snímok zhotovených prístrojom Oculus Keratograph

5M. V rámci predspracovania bola na zvýšenie kontrastu v snímkach použitá technika adaptívneho vyrovnávania histogramu s obmedzeným kontrastom (CLAHE). Zaujímavým krokom v tejto práci bolo použitie techník elastickej transformácie[34] a perturbácie za účelom augmentácie dát namiesto klasických techník ako posun, či rotácia. Trénovanie prebiehalo 300 epoch s použitím stratovej funkcie binárnej krížovej entropie. Na vyhodnotenie bolo použitých 5 metrik, z čoho najrelevantnejšia pre moju prácu je pomer medzi povrchom viečka, ktorý nie je pokrytý žlazami a celkovým povrchom viečka. Porovnanie s anotovanými dátami bolo vykonané vypočítaním p-hodnoty na základe dvojitýberového t-testu s predpokladom rovnakých variancií. Výsledné p-hodnoty pre testovaciu dátovú sadu rozdelenú do štyroch podmnožín podľa typu kamery a prítomnosti známkov degenerácie očného viečka sa nachádzajú v tabuľke 2.2.

Kamera	Degenerácia meibomových žliaz	Priemerná hodnota a rozptyl pre anotované dáta	Priemerná hodnota a rozptyl pre segmentované dáta	p-hodnota
prototyp	áno	$0,69 \pm 0,14$	$0,70 \pm 0,13$	0,568
prototyp	nie	$0,54 \pm 0,08$	$0,56 \pm 0,06$	0,250
occulus	áno	$0,66 \pm 0,10$	$0,64 \pm 0,13$	0,173
occulus	nie	$0,59 \pm 0,08$	$0,59 \pm 0,05$	0,403

Tabuľka 2.2: Vyhodnotenie úspešnosti segmentácie meibomových žliaz v práci [28]

V práci od Sahu a spol.[32] bola na segmentáciu očného viečka aj meibomových žliaz využitá enkodér-dekodér architektúra s použitím ResNet34 ako enkodéra. Vytvorený bol dataset MGD-1K, ktorý bol následne voľne prístupný. Jeho bližší popis sa nachádza v sekcii 4.2. Pred samotným trénovaním prebehlo predspracovanie snímok a bola natrénovaná sieť typu GAN pre odstránenie odlesku, teda priveľmi osvetlených plôch v obraze. Na dátovú augmentáciu bolo použité zaostrenie a rotácia v rozmedzí $\pm 15^\circ$. Model bol trénovaný 200 epoch a pre oba typy segmentácie (očné viečko / žlazy) bola použitá rovnaká architektúra a postup trénovania. Ako testovacia metrika bol použitý pomer prieniku a zjednotenia (prekryv). Pri segmentácii meibomových žliaz bolo dosiahnuté priemerné skóre 67,63%, pričom skóre pri porovnaní medzi dvomi odborníkmi z praxe bolo stanovené na 64,03%. V práci bola následne riešená aj problematika určenia meiboskóre. Grafickú reprezentáciu použitého postupu je možné vidieť na obrázku 2.6.



Obr. 2.6: Reprezentácia modelov neuronových sietí v práci [32]

Setu a spol.[33] prezentujú použitie U-Net architektúry, v ktorej využívajú Inception-ResNet-v2. Cieľom práce je segmentácia meibomových žliaz a následne štúdium ich morfológie. Ich dátová sada pozostáva zo 178 snímok vytvorených prístrojom Oculus Keratograph 5M. V rámci prípravy sú snímky upravené na veľkosť 256 x 256 pixelov. Pred samotným tréňovaním siete je model predtréňovaný na röntgenových snímkach hrude počas piatich epoch. V procese tréňovania je použitý optimalizačný algoritmus Adam a ako stratová funkcia je použitý rozdiel krížovej entropie a Diceovej objektívnej funkcie, ku čomu je následne pripočítaná hodnota 1. Sieť sa trénuje počas 30 epoch. Segmentácia meibomových žliaz dosiahla presnosť 83%, úplnosť 81%, F1 skóre 84% a Diceov koeficient 84%.

V práci od Wanga a spol.[38] je použitá sieť ResNet50. Táto práca sa nezaobrá priamo segmentáciou jednotlivých meibomových žliaz, ale oblasti, ktorú pokrývajú, zahŕňajúc aj medzi-žlazové priestory. Taktiež je segmentované očné viečko, aby mohol byť následne vypočítaný pomer týchto dvoch oblastí a meiboskóre. Dataset tvorí 775 snímok z prístroja Oculus Keratograph 5M. Sieť využívala PSPNet[42] (pyramídová analýza scény). Výstupom siete bola dvojkanálová matica, kde jeden kanál reprezentoval segmentačnú masku očného viečka a druhý segmentačnú masku oblasti meibomových žliaz. Okrem toho bol pomocným výstupom aj vektor určujúci, či sa jedná o atrofiu alebo nie. Pre tento vektor bol nastavený podiel stratovej funkcie na 0,1. Tréňovanie prebiehalo 200 epoch. Segmentácia očného viečka dosiahla úspešnosť 97,6% a priemerný pomer prieniku a zjednotenia 95,4%. Pre určenie plochy pokrytej meibomovými žlazami bola úspešnosť 95,5% a pomer prieniku a zjednotenia 66,7%.

Zaujímavou je práca od Liua a spol.[22], kde dáta nie sú anotované segmentačnou maskou, ale iba neúplnými obrysami hrán meibomových žliaz a očného viečka. Takto anotovaný dataset vyžaduje kratší čas prípravy a anotácie sú menej presné, ale aj napriek tomu sú dosiahnuté porovnateľné výsledky s ostatnými prácami. Použitá je U-Net architektúra.

Spomenuté práce je ťažké porovnať navzájom medzi sebou, nakoľko sú na meranie využité rôzne metriky a každá z prác pracuje s inou dátovou sadou. Je však možné zhodnotiť, že aktuálne sa najlepšie výsledky s ohľadom na metriku pomeru prieniku a zjednotenia pohybujú na úrovni 95% v oblasti segmentácie očného viečka a 66% v oblasti segmentácie meibomových žliaz.

Kapitola 3

Neurónové siete určené na segmentáciu obrazových dát

Táto kapitola popisuje teóriu neurónových sietí. Najskôr sú predstavené základné princípy a algoritmy, na ktorých sú neurónové siete postavené. Následne sú popísané konvolučné neurónové siete dôležité najmä pri segmentačných, či klasifikačných úlohách vyžadujúcich prácu s obrazom a nakoniec sú uvedené relevantné architektúry sietí používaných na segmentáciu obrazových dát najmä v zdravotníctve. Teoretické poznatky o fungovaní neurónových sietí popísané nižšie pochádzajú prevažne z práce od Goodfellowa a spol. [13]

3.1 Základné princípy neurónových sietí

Stavebné prvky neurónových sietí

Základy neurónových sietí boli definované v 50. a 60. rokoch dvadsiateho storočia Frankom Rosenblattem, ktorý predstavil prvý model perceptrónu. Ten sa stal základnou funkčnou jednotkou jednoduchých neurónových sietí.

$$y = f\left(\sum_{i=1}^n w_i * x_i + b\right) \quad (3.1)$$

Spôsob výpočtu výstupnej hodnoty perceptrónu alebo aj neurónu sa nachádza v rovnici 3.1. Vstupom je vektor vstupných hodnôt \vec{x} obsahujúci n prvkov a prislúcha mu rovnako dlhý vektor váh neurónu \vec{w} . Ku súčtinu jednotlivých prvkov vektorov je pripočítané skreslenie b . Súčet týchto hodnôt naprieč všetkými prvkami vektorov je vstupom pre aktivačnú funkciu f .

Medzi najznámejšie a najpoužívanejšie aktivačné funkcie patria:

- **Sigmoid** - škáluje hodnotu výstupu neurónu do intervalu od 0 do 1

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

- **ReLU** (*Rectified linear unit*) - veľmi často používaná funkcia, deaktivuje neuróny s výstupom menším ako 0, ostatné hodnoty prepúšťa bez ich zmeny

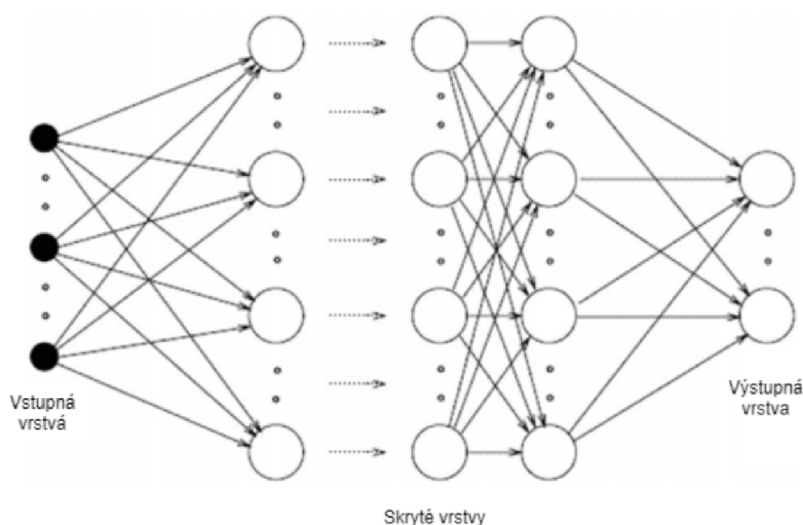
$$f(x) = \begin{cases} x & \text{ak } x \geq 0 \\ 0 & \text{ak } x < 0 \end{cases} \quad (3.3)$$

- **SoftMax** - najčastejšie použitá vo výstupnej vrstve siete klasifikujúcej do viacerých tried

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ pre } j = 1, \dots, K \quad (3.4)$$

Ďalšími známymi aktivačnými funkciami sú funkcia binárneho kroku (použitá i v pôvodnom návrhu perceptrónu), lineárna funkcia, funkcia hyperbolického tangensu, swish funkcia, či variaty ReLU funkcie ako presakujúca (*leaky*) ReLU, parametrizovaná ReLU a ELU.

Hlavný význam aktivačnej funkcie spočíva najmä v transformácii lineárnej kombinácie vstupov na nelineárny výstup. Táto nelinearita umožňuje neurónovým sieťam riešiť aj lineárne neseparovateľné problémy a zároveň má priaznivý vplyv v procese tréningu sietí pri výpočte gradientu, ktorý by bol pre lineárne funkcie v každom ich bode rovnaký.



Obr. 3.1: Usporiadanie neurónov v neurónovej sieti

Typicky sú neuróny usporiadané do neurónových sietí vo vrstvách ako na obrázku 3.1. Vstupná vrstva predstavuje vektor vstupných dát. Za vstupnou vrstvou sa môže nachádzať ľubovoľný počet skrytých vrstiev a sieť je ukončená výstupnou vrstvou, ktorá napríklad v prípade klasifikačnej úlohy obsahuje väčšinou taký počet neurónov ako počet klasifikačných tried. Neuróny medzi sebou môžu byť ľubovoľne poprepájané, avšak najčastejšie je možné sa stretnúť s plne prepojenými neurónovými sieťami. V prípade takéhoto prepojenia je výstup neurónu predchádzajúcej vrstvy napojený na každý z neurónov bezprostredne nasledujúcej vrstvy.

Spomínané usporiadanie neurónových sietí však neplatí napríklad v prípade rekurentných sietí, kde môže byť výstup neurónu napojený i na niektoré z neurónov, ktoré sa podieľajú na vstupe daného neurónu. Ide napríklad o Hopfieldovu sieť, Boltzmannov stroj, či siete používajúce pamäťové bloky LSTM, či GRU.

Trénovanie neurónových sietí

V procese trénovania neurónových sietí je potrebné definovať meranie chyby, akej sa sieť dopustila. Na to slúžia stratové alebo objektívne funkcie určujúce kvalitu parametrov siete. Cieľom trénovania je minimalizácia stratovej funkcie, alebo maximalizácia objektívnej funkcie.

Základnými stratovými funkciami je stratová funkcia absolútnej chyby (L1) alebo stratová funkcia kvadratickej chyby (L2). Medzi známe a často používané stratové funkcie nielen pre segmentačné úlohy patrí napríklad stratová funkcia krížovej entropie definovaná ako:

$$L(W) = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m (y_{ij} * \log(p_{ij})) \quad (3.5)$$

Kde n predstavuje počet pixelov predikovanej segmentačnej masky, m počet klasifikačných tried, y_{ij} nadobúda hodnoty 0 alebo 1 podľa toho, či je trieda i správnou klasifikačnou triedou pixelu j a p_{ij} je sieťou určená pravdepodobnosť toho, že daný pixel patrí do triedy i .

V prípade klasifikácie do dvoch tried, či segmentácii iba jedného typu objektu hovoríme o binárnej krížovej entropii, kedy sa pôvodný vzorec zmení na:

$$L(W) = \frac{1}{n} \sum_{i=1}^n (-y_{ij} * \log(p_{ij}) - (1 - y_{ij}) * \log(1 - p_{ij})) \quad (3.6)$$

Objektívnou funkciou, pri ktorej je cieľom jej maximalizácia je napríklad Diceova objektívna funkcia založená na Sørensen-Diceovom koeficiente. Funkcia bola prvýkrát použitá v práci od Milletariho a spol. [26] a je vhodná najmä pre segmentačné úlohy. Má nasledovný tvar:

$$D = \frac{2 \sum_{i=1}^n p_i g_i}{\sum_{i=1}^n p_i^2 + \sum_{i=1}^n g_i^2} \quad (3.7)$$

Kde sa sčítava cez n pixelov binárnej segmentačnej masky $p_i \in P$ vyprodukovanej sieťou a skutočnej binárnej segmentačnej masky $g_i \in G$. Diceova funkcia nadobúda hodnoty od 0 do 1.

Pred trénovaním neurónovej siete procesom učenia s učiteľom je potrebné rozdelenie dátovskej sady. Najčastejšie sa delí na tri podmnožiny - trénovaciú, validačnú a testovaciú. Dáta trénovacej množiny slúžia na jej trénovanie. Po každom predom stanovenom cykle trénovania je úspešnosť siete vyhodnotená na validačnej sade. Pokiaľ je pozorovaný klesajúci trend miery úspešnosti na validačnej sade a stúpajúca úpešnosť na trénovacej, značí to, že sieť prestáva dostatočne dobre generalizovať. Po ukončení trénovania je model vyhodnotený na testovacej sade.

Najčastejšie sa na trénovanie neurónových sietí používa metóda klesajúceho gradientu spolu so spätnou propagáciou chyby. Po prechode prvku trénovacej množiny sieťou je vypočítaná hodnota stratovej funkcie. Gradient pre jednotlivé parametre siete je vypočítaný na základe parciálnych derivácií reťazovým pravidlom postupujúc od výstupnej po vstupnú vrstvu siete. Parametre siete sú potom aktualizované podľa vzorca:

$$\vec{W} = \vec{W} - \epsilon * p \vec{D} \quad (3.8)$$

Kde \vec{W} značí parametre siete, $p \vec{D}$ vektor parciálnych derivácií a ϵ predstavuje mieru učenia, ktorá je hyperparametrom pri trénovaní siete a môže sa počas trénovania meniť.

Typicky sa chyba nevypočítava iba pre jeden prvok trérovacej množiny, ale pre jej podmnožinu. Tú označujeme ako minidávka a metódy výpočtu založené na tomto princípe sa nazývajú stochastické metódy a zefektívňujú trérovanie siete. Medzi najznámejšie metódy optimalizujúce aktualizáciu parametrov siete patria:

- **Stochastické klesanie gradientu** - najzákladnejší optimalizačný algoritmus založený na výpočte hodnoty stratovej funkcie z viac ako jedného prvku trérovacej množiny - minidávky
- **Momentum** - zachováva hybnosť klesania gradientu čím pomáha prekonať stacionárne miesta.
- **Nesterov momentum** - variácia algoritmu momenta
- **AdaGrad** [8] - algoritmus s adaptívnou mierou učenia, pracuje so všetkými historickými hodnotami gradientu.
- **RMSProp** - upravenie AdaGrad algoritmu, zmena hromadenia všetkých gradientov na exponenciálne vážený priemer pohybu.
- **Adam** [17] - kombinácia RMSProp a momenta s ďalšími vylepšeniami

Pri trérovaní sietí môže nastať situácia, kedy model veľmi dobre vyhodnocuje trérovacie dáta, ale vykazuje slabú úspešnosť na testovacej sade. Vtedy dochádza k pretrérovaniu siete, čiže sieť dostatočne dobre negeneralizuje. Tomu sa snažia zabrániť regularizačné techniky. Medzi najpoužívanejšie patria L1 a L2 regularizácia, či dropout.

L1 a L2 regularizácia spočívajú v pripočítaní regularizačného termu k stratovej funkcii. Regularizačné termy majú za cieľ eliminovať extrémne hodnoty parametrov, čo vedie k vytváraniu jednoduchších modelov menej náchylných na pretrérovanie. Pre L1 regularizáciu je tak stratová funkcia vypočítaná:

$$L_{\lambda}(W) = L(W) + \lambda * \sum_{i=1}^n |w_i| \quad (3.9)$$

Stratová funkcia po L2 regularizácii má tvar:

$$L_{\lambda}(W) = L(W) + \lambda * \sum_{i=1}^n w_i^2 \quad (3.10)$$

V týchto vzorcoch predstavuje λ regularizačný parameter, ktorý je hyperparametrom, n je počet parametrov siete a w_i je hodnota konkrétnej váhy siete.

Dropout je technika deaktivujúca určité neuróny v procese trérovania na základe zvolenej pravdepodobnosti. Výstup takýchto neurónov je následkom toho vynulovaný a model sa tak zjednodušuje.

Ďalšou možnosťou prevencie proti preučeniu je rozšírenie dátovej sady. V prípade, že nie je možnosť získať nové dáta, je vhodné využiť techniky dátovej augmentácie. Viac dát siete zabraňuje kopírovať príliš špecifické vzory trérovacej množiny, čím sa napomáha lepšej generalizácii.

Známymi augmentačnými technikami pre obrazové dáta v oblasti geometrických transformácií sú posuny, rotácie, zrkadlenie, náklon, či elastická transformácia. V oblasti zmien kvality, či intenzity obrazu ide o vnesenie šumu, zaostrenie, rozostrenie, alebo zmeny jasů a kontrastu.

3.2 Konvolučné neurónové siete

Konvolučné neurónové siete majú veľkú výhodu oproti bežným neurónovým sieťam pri spracúvaní viacdimeziálnych dát, ako napríklad obraz, či video, keďže zachovávajú informáciu a susednosti pixelov obrazu. Používajú sa ale i na prácu s jednorozmerným signálom, ako napríklad zvuk. Základnou operáciou v tomto type sietí je operácia konvolúcie.

Operácia konvolúcie

Pre zjednotenie predstavuje nasledujúca sekcia operáciu konvolúcie aplikovanú v sieti spracúvajúcej viacnásobné obrazové dáta.

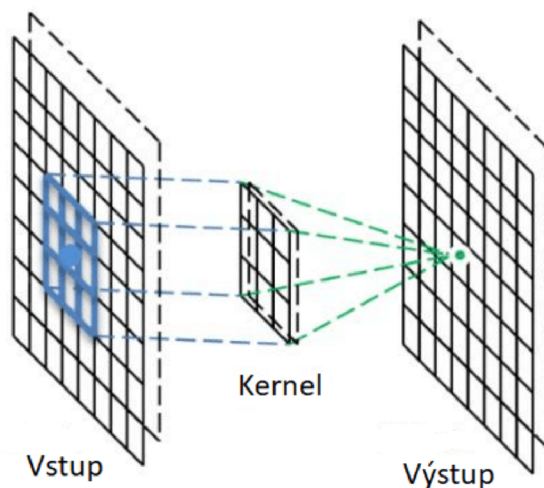
Operácia konvolúcie je v konvolučnej neurónovej sieti vykonávaná prostredníctvom filtrov nazývaných kernelové jadrá. Takéto jadro je definované svojimi rozmermi a hodnotami váh. V prípade trojkanálového vstupu je kernelové jadro zložené z troch samostatných maticí a každá je aplikovaná na príslušný kanál vstupu. Toto jadro sa pohybuje po vstupnej matici ako posuvné okno so stanovenou veľkosťou kroku (*stride*). Pre každú pozíciu a každý kanál vo vstupnej matici je potom vypočítaná hodnota výstupu podľa vzorca:

$$f[x, y] * g[x, y] = \sum_{i=1}^X \sum_{j=1}^Y f[x_i, y_j] g[x_i, y_j] \quad (3.11)$$

Kde f predstavuje oblasť jedného kanálu vstupného obrázka prekrytého kernelovým jadrom, g predstavuje kernelové jadro, X a Y rozmiery kernelového jadra. Výstupy konvolúcie na danej pozícii pre každý kanál sú následne sčítané. Operácia konvolúcie sa v literatúre označuje symbolom $*$.

Výstupom aplikácie kernelového jadra na celú vstupnú maticu je dvojrozmerná matica nazývaná aktivačná mapa. V jednej vrstve konvolučnej neurónovej siete je typicky aplikovaných viac kernelových jadier, čím vznikne ekvivalentný počet aktivačných máp.

V niektorých prípadoch, najmä pre zachovanie rozmerov sa ku vstupnej matici pridáva vyplnenie okrajov (*padding*), ktoré sa bežne vyplňajú nulovými hodnotami.



Obr. 3.2: Aplikácia kernelového jadra na vstupnú maticu

V paralele s klasickými neurónovými sieťami teda možno kernelové jadro chápať ako neurón, pričom jednotlivé váhy kernelového jadra sú ekvivalentné váham neurónu. Rozdiele-

lom je, že kernelové jadro má váhy zdieľané pre celú vstupnú maticu, čo znamená menší počet parametrov a translačnú invarianciu. Inak povedané, vďaka tejto vlastnosti sieť nie je naviazaná na konkrétnu pozíciu v matici pri hľadaní istej črty.

Pre ilustráciu práce konvolučnej vrstvy teda v prípade vstupu o rozmeroch 5×5 s tromi kanálmi zvolíme dve kernelové jadrá o rozmeroch 3×3 s veľkosťou kroku 1. Výstupom tejto vrstvy budú dve aktivačné mapy o rozmeroch 3×3 a celkový počet trénovateľných parametrov je 54.

Stavebné prvky konvolučných neurónových sietí

Základným stavebným prvkom je okrem konvolučnej vrstvy združovacia (*pooling*) vrstva. Tá slúži predovšetkým na zmenšenie výstupov, čím sa skracuje čas výpočtu, prípadne na potlačenie nedominantných signálov a odstránenie šumu.

Tak ako konvolučné jadro, aj združovacia vrstva má definované rozmery filtra a veľkosť kroku. Najčastejšie je pre veľkosť kroku 2 použitý rozmer filtra 2×2 . Takáto vrstva združuje 4 hodnoty vstupu do jednej, teda redukuje veľkosť výstupu o 75 %. Medzi najčastejšie združovacie filtre patrí maximalizačný, ktorý prepúšťa maximálnu hodnotu, ale často používaný je aj priemer.

Nakoľko je operácia konvolúcie lineárnou operáciou, je pre lepšie fungovanie siete potrebné vnesenie nelinearity zabezpečené použitím aktivačnej funkcie. V konvolučných neurónových sieťach je najčastejšie použitá ReLU aktivačná funkcia.

Siete určené na klasifikáciu sú vo veľkej miere tvorené striedaním konvolučných a združovacích vrstiev s aplikáciou aktivačnej funkcie. No konci siete je výstup posledného bloku transformovaný do jednorozmerného vektora a sieť ďalej pokračuje ako plne prepojená sieť.

V prípade segmentácie obrazových dát je však takýto prístup z dôvodu veľkej straty rozlíšenia a priestorového rozloženia nepoužiteľný, a preto sú do sietí vložené i ďalšie stavebné prvky. Jednotlivé prístupy riešenia tejto problematiky pre konkrétne typy architektúr sú popísané v podkapitole 3.4.

3.3 Metriky úspešnosti segmentačných neurónových sietí

Pre vyhodnotenie správnosti fungovania a porovnanie s existujúcimi riešeniami sa pre segmentačné úlohy využíva niekoľko metrík. Všetky z týchto metrík sa dajú vyjadriť prostredníctvom nasledujúcich termov:

- **TP** (*true positive*) - počet pixelov správne označených ako súčasť segmentovaného objektu
- **TN** (*true negative*) - počet pixelov správne označených ako oblasť mimo segmentovaného objektu
- **FP** (*false positive*) - počet pixelov nesprávne označených ako súčasť segmentovaného objektu
- **FN** (*false negative*) - počet pixelov nesprávne označených ako oblasť mimo segmentovaného objektu

Najčastejšie používané metriky sú nasledovné:

- **Správnosť (*accuracy*)** - slúži na výpočet percentuálneho pomeru počtu pixelov zaradených do správnej triedy.

$$\text{Správnosť} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.12)$$

- **Presnosť (*precision*)** - uvádza podiel správne označených pixelov výstupnej segmentačnej masky a celej výstupnej segmentačnej masky siete.

$$\text{Presnosť} = \frac{TP}{TP + FP} \quad (3.13)$$

- **Úplnosť (*recall*)** - pomer správne označených pixelov patriacich reálnej segmentačnej maske a celou reálnou segmentačnou maskou.

$$\text{Úplnosť} = \frac{TP}{TP + FN} \quad (3.14)$$

- **F1 skóre** - v literatúre pre segmentačné úlohy označovaný aj ako Diceov koeficient. Definované ako harmonický priemer presnosti a úplnosti. A a B predstavujú množiny pixelov reálnej segmentačnej masky a masky vygenerovanej sieťou.

$$F1 \text{ skóre} = \frac{2 * \text{Presnosť} * \text{Úplnosť}}{\text{Presnosť} + \text{Úplnosť}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} = \frac{2|A \cap B|}{|A| + |B|} \quad (3.15)$$

- **Špecifická** - predstavuje mieru schopnosti správne označiť pixel neprislúchajúci segmentačnej maske.

$$\text{Špecifická} = \frac{TN}{TN + FP} \quad (3.16)$$

- **Prekryv (*intersection over union*)** - vyjadruje mieru prekryvu pôvodnej masky a masky vysegmentovanej sieťou.

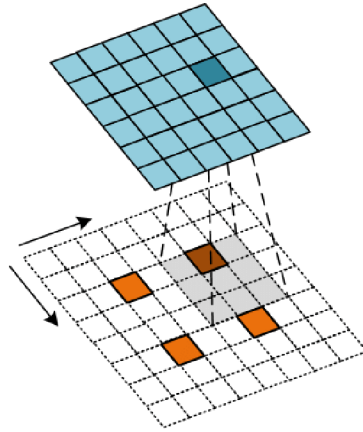
$$\text{Prekryv} = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \quad (3.17)$$

3.4 Relevantné architektúry konvolučných neurónových sietí

Na segmentáciu obrazových dát bolo vyvinutých množstvo architektúr a stále vznikajú ďalšie. V tejto podkapitole sú rozobrané najpoužívanejšie spolu s inováciami, ktoré priniesli.

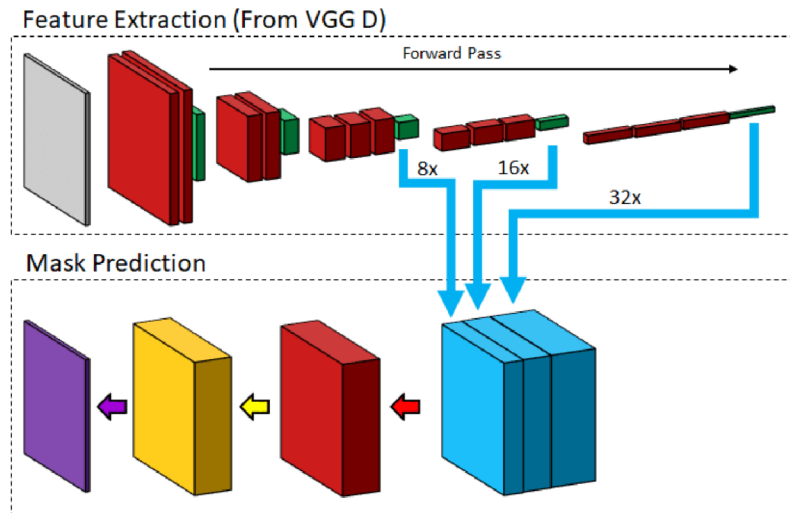
Plne konvolučné siete

Plne konvolučná sieť bola predstavená v práci od Longa a spol. [23]. Pri vývoji sietí tohto typu sa používajú architektúry konvolučných sietí bežne slúžiacich na klasifikáciu, pričom ich modifikujú odstránením posledných plne prepojených vrstiev a nahrádzajú ich vrstvou transponovanej konvolúcie, v práci nazývanej dekonvolúcia. V literatúre má však pojem dekonvolúcia mierne iný význam ako transponovaná konvolúcia [10].



Obr. 3.3: Ukážka transponovanej konvolúcie [41].

Cieľom transponovanej konvolúcie je narozdiel od bežnej konvolúcie generovanie priestorovo rozmernejších výstupov, teda nadvzorkovanie (*upsampling*) vstupu. Definovaná je rozmermi kernelového jadra obsahujúcim váhy, veľkosťou kroku a vyplnením. V tomto procese dochádza k dilatácii vstupnej matice, ako je možné vidieť na obrázku 3.3. Pre tento konkrétny prípad je aplikované kernelové jadro o veľkosti 3x3 na vstup o rozmeroch 2x2. Hodnota vyplnenia je nastavená na 2, čo je na obrázku znázornené dvomi nulovými stĺpcami a riadkami na okrajoch vstupu. Hodnota veľkosti kroku je stanovená na 3. Táto hodnota určuje dilatáciu medzi jednotlivými pixelmi vstupu, ktorá sa vypočíta ako *veľkosť kroku* - 1. Prázdne miesta sú vyplnené nulovými hodnotami. Takto upravený vstup je následne prechádzaný kernelovým jadrom s veľkosťou kroku 1. Výstup má rozmery 6x6, čo predstavuje 9-násobné nadvzorkovanie.

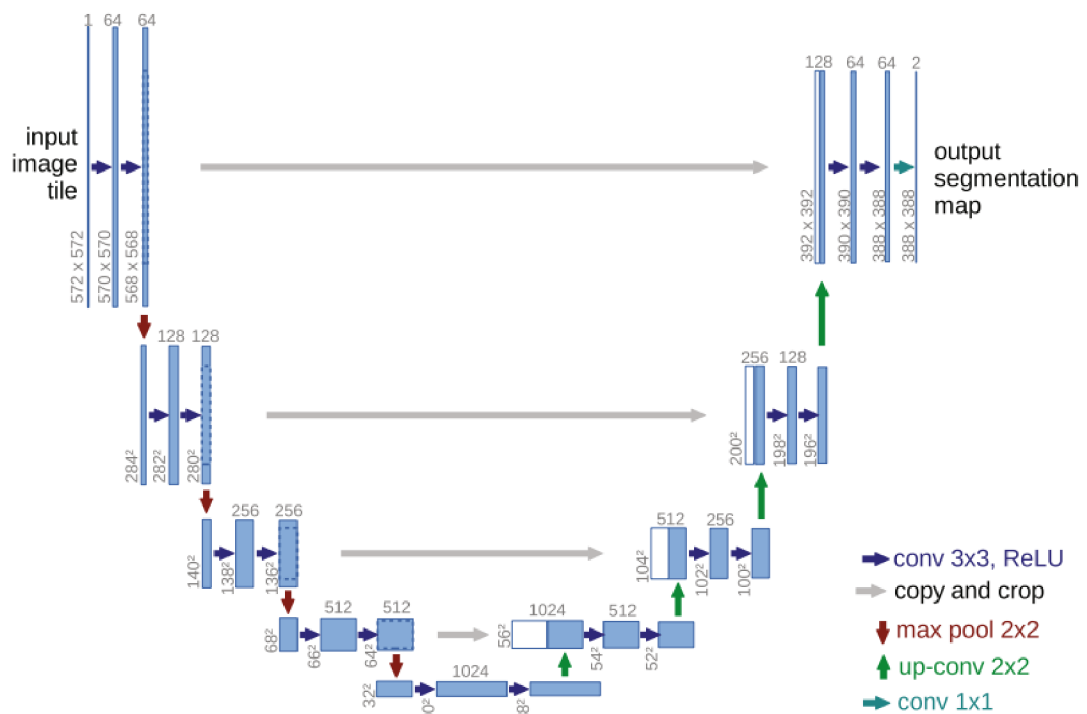


Obr. 3.4: Architektúra plne prepojenej siete z práce [23]. Červenou sú znázornené konvolučné vrstvy, zelenou združovacie, modrou vrstvy transponovanej konvolúcie s faktorom nadvzorkovania a žltá vrstva predstavuje softmax funkciu. Obrázok bol prebraný z práce [10].

V práci [23] boli vykonané experimenty pridaním vrstvy transponovanej konvolúcie k modifikovaným architektúram sietí AlexNet [20], VGG-16 [35] a GoogLeNet [37]. Najlepšie výsledky spomedzi spomenutých dosiahla architektúra s použitím VGG-16. Na dátovej sade PASCAL VOC [9] bola dosiahnutá správnosť 90,3 % a prekryv 62,7 %. Architektúra je znázornená na obrázku 3.4.

Enkodér-dekodér architektúra

Druhým typom architektúr sú architektúry pozostávajúce z dvoch častí - kódujúcej a dekódujúcej. V tomto procese je najčastejšie kódovanie vykonané striedaním konvolučných a združovacích vrstiev, čím sa postupne znižujú rozmery výstupov. Dekodér potom zväčšuje rozmery najčastejšie prostredníctvom transponovanej konvolúcie. Táto architektúra sa využívala na kódovanie obrázkov, pričom dekódujúca časť mala za úlohu rekonštrukciu obrázka. Generatívne vlastnosti dekodéra sa však dajú veľmi dobre využiť aj pri segmentačných úlohách.



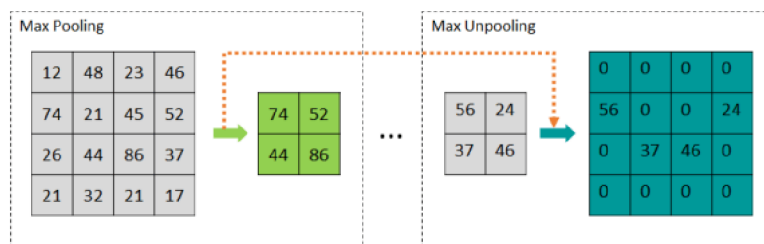
Obr. 3.5: Architektúra segmentačnej konvolučnej neurónovej siete U-Net [31].

Najznámejším príkladom takejto segmentačnej architektúry je architektúra U-Net [31], ktorá je znázornená na obrázku 3.5. Dekodér siete je takmer zrkadlovým obrazom enkodéra. Sieť pozostáva celkovo z 19 konvolyčných vrstiev prevažne s kernelovými jadrami o rozmeroch 3x3 a ReLU aktivačnou funkciou a štyroch vrstiev obsahujúcich transponovanú konvolyciu. Pre tento model je typickým preskakovaním spojením (*skip connections*), čo znamená že vyextrahované príznaky jednotlivých blokov enkodéra okrem toho, že pokračujú ďalej sieťou, sú v nezmenenej podobe pripojené k príznakom protilahlého bloku dekodéra.

Generatívny proces dekodéra napomáha riešeniu problému straty rozlíšenia v procese extrakcie príznakov enkodérom. Model U-Net bol vyhlásený za víťaza ISBI výzvy na sle-

dovanie buniek, kde dosiahol hodnotu miery prekryvu na úrovni 92,03 % na dátovej sade PhC-U373. Modifikáciou počtu konvolučných vrstiev, či rozmerov kernelových jadier vznikajú rôzne variácie architektúry U-Net. Modifikovaná, či nahradená inou architektúrou môže byť tiež celá časť enkodéru.

Na segmentáciu volumetrických dát bol vyvinutý model siete V-Net [26], s podobnou organizáciou vrstiev ako U-Net. Táto architektúra používa namiesto dvojrozmerných konvolučných filtrov trojrozmerné a výstupom sú trojrozmerné segmentačné masky. Model dosiahol skóre 82,39 vo výzve PROMISE 2012 [21].

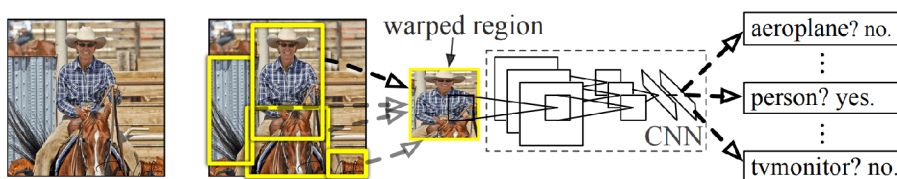


Obr. 3.6: Rozširovanie výstupu prostredníctvom *unpoolingu* [10].

SegNet architektúra [4] pozostáva z piatich kódujúcich blokov kopírujúc architektúru VGG-16. Nasleduje 5 dekódujúcich blokov kde je však namiesto transponovanej konvolúcie použité rozširovanie vstupu (*unpooling*). V každej združovacej vrstve s ponechaním maxima v redukujúcej časti siete sú zapamätávané indexy maxima a následne sú v procese expanzie tieto hodnoty dopĺňané na príslušné pozície. Proces *max-poolingu* a *unpoolingu* pre združovacie a rozširovacie filtre o rozmeroch 2x2 je znázornený na obrázku 3.6.

Architektúry segmentujúce oblasti

Architektúry označované ako R-CNN (*Regional Convolutional Neural Networks*) sú určené prevažne na detekciu objektov v obrazových dátach. Prvý zástupca tohto typu bol predstavený v práci od Girshicka a spol. [12] a základný postup detekcie objektov sa nachádza na obrázku 3.7. V prvom kroku sa vykonáva selekcia regiónov obrázka. Tých sa vyberie približne 2000 a každý z nich je vstupom klasifikačnej konvolučnej neurónovej siete, ktorá určuje, či sa v danom priestore objekt nachádza.

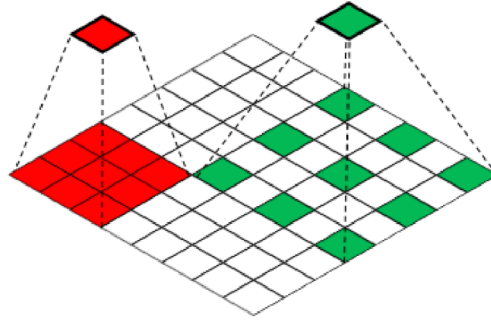


Obr. 3.7: Algoritmus detekcie objektov architektúrou R-CNN [12].

Proces výpočtu je časovo i výpočtovo náročný a postupne boli vyvíjané nové modely ako Fast R-CNN [11] a Faster R-CNN [29], v ktorých sa menil spôsob výberu regiónov ako aj vrstvy sietí. Prvou z tejto rodiny sietí, ktorá vykonávala aj sémantickú segmentáciu je architektúra Mask R-CNN [14] vyvinutá v roku 2017.

DeepLab

Architektúry rodiny DeepLab sa snažia vylepšiť problémy predchádzajúcich segmentačných architektúr, ktoré spočívajú najmä nedostatku kontextuálnej informácie, spôsobenej malými rozmermi kernelových jadier a straty rozlíšenia po združovacích vrstvách. Riešením je použitie dilatovanej (angl. *atrous*) konvolúcie, ktorej porovnanie s klasickou konvolúciou je vidieť na obrázku 3.8.



Obr. 3.8: Klasická konvolúcia s kernelom veľkosti 3x3 znázornená červenou farbou v porovnaní s dilatovanou konvolúciou s kernelom o rozmeroch 3x3 a hodnotou dilatácie 1 znázornená zelenou [10].

Pre dilatovanú konvolúciu sa okrem rozmeru filtra stanovuje i hodnota dilatácie. Na rozdiel od transponovanej konvolúcie, kde dilatácii podlieha vstup filtra, pri dilatovanej konvolúcii jej podlieha samotný filter, čo umožňuje pokrytie väčšej kontextuálnej oblasti vstupu bez nutnosti pridávania ďalších parametrov.

Rodina DeepLab postupne napredovala a vznikali DeepLabv1, DeepLabv2, DeepLabv3 a DeepLabv3+. V zásade tieto architektúry zapracúvávajú dilatovanú konvolúciu do existujúcich modelov klasifikačných konvolučných architektúr a segmentácia je vykonávaná aj s použitím príznakov vytvorených vo vnútri týchto architektúr.

Okrem dilatovanej konvolúcie je v architektúrach DeepLab použitá aj priestorové pyramídové združovanie (angl. *spatial pyramid pooling*) [15] alebo plne prepojené podmienené náhodné polia (angl. *fully connected conditional random fields*) [19], ktoré pri klasifikovaní prvku berú na vedomie aj hodnoty okolitých prvkov [24].

Kapitola 4

Návrh riešenia

Táto kapitola sa zaoberá návrhom riešenia implementovaného v rámci diplomovej práce. Na začiatku je uvedená stručná špecifikácia prototypu kamery použitej pre účely tejto práce. Nasleduje popis dátovej sady a spôsob jej spracovania. Neskôr sú predstavené zvolené architektúry neurónových sietí aj s postupom ich tréovania a evaluácie a nakoniec je spracovaný návrh aplikácie s používateľským rozhraním.

Cieľom tejto práce je segmentácia očného viečka a detekcia meibomových žliaz z videosekvencie. Finálnym produktom práce sú modely konvolučných neurónových sietí umožňujúce túto segmentáciu a aplikácia, ktorá ich využíva v rámci nasledujúcich dvoch režimov činnosti:

- **Spracovanie už nasnímaných videosekvencií** - V tomto režime aplikácia upravuje zvolenú videosekvenciu tak, aby boli vizuálne odlíšené meibomové žľazy a oblasť záujmu - teda očné viečko od ostatných častí obrazu.
- **Analýza a zobrazenie vstupu z kamery v reálnom čase** - V tomto režime aplikácia umožňuje analýzu videosekvencie z kamery popísanej v sekcii 4.1 v reálnom čase.

4.1 Špecifikácia kamery použitej pre účely tejto práce

Pri tvorbe funkčnej aplikácie na segmentáciu očného viečka a detekciu meibomových žliaz bola použitá kamera *e-CAM82_USB - Sony STARVIS™ IMX415 4K Ultra-HD USB Camera*¹. Kamera je určená do podmienok s nízkou svetelnosťou. Základná špecifikácia kamery sa nachádza v tabuľke 4.1.

Súčasťou prototypu je okrem samotnej kamery aj plastový rám s držadlom pre ľahšie uchopenie a manipuláciu. Prototyp tiež obsahuje používateľom nastaviteľné infračervené osvetlenie s nastaviteľnou vlnovou dĺžkou 840 - 940 nm. Pre zachytenie oblastí osvetlených týmito vlnovými dĺžkami je potrebné, aby mala kamera dostatočne vysokú spektrálnu senzitivitu v danej oblasti svetelného spektra.

Nakoľko kamera prechádzala pri vývoji zmenami, ktoré zahŕňali i typ kamery, nebolo možné do odovzdania práce zapracovať všetky modifikácie, a tak aplikácia podporuje iba prvý prototyp zariadenia snímacieho meibomové žľazy, ktorý obsahuje kameru See3CAM_CU27 od spoločnosti e-con Systems Inc.

¹<https://www.e-consystems.com/usb-cameras/sony-imx415-4k-usb-camera.asp>

Popis	Špecifikácia
Veľkosť (dĺžka x šírka x výška)	30 x 30 x 25 mm (bez šošoviek)
Formát videa	nekomprimovaný YUY2 a komprimovaný MJPEG
USB	USB 2.0
Rozlíšenie	FHD (1920 x 1080) 4K (3864 x 2176)
Veľkosť pixela	1.45 μm x 1.45 μm
Podporované operačné systémy	Windows, Linux, Android, MacOS

Tabuľka 4.1: Základná špecifikácia použitej kamery



Obr. 4.1: Kamera použitá na snímanie očného viečka

4.2 Dátová sada

K dispozícii je anonymizovaná dátová sada meibografických snímok s názvom MGD-1K², ktorá bola vytvorená v rámci práce od Sahu a spol.[32] Dátová sada obsahuje dohromady 1000 snímok obráteného horného i dolného viečka. Infra-červené snímky boli vytvorené s použitím LipiView interferometra (LipiView II Ocular Surface Interferometer, Johnson & Johnson Inc., Jacksonville, FL) metódou bezkontaktnej meibografie. Pre každú zo snímok boli určené segmentačné masky očného viečka, a meibomových žliaz. Príklad jednej vzorky dátovej sady pre horné viečko sa nachádza na obrázku 4.2 a dolné viečko na obrázku 4.3.



(a) Originálna snímka horného viečka (b) Segmentačná maska očného viečka (c) Segmentačná maska meibomových žliaz

Obr. 4.2: Príklad snímky horného viečka z dátovej sady MGD-1K

²<https://mgd1k.github.io>



Obr. 4.3: Príklad snímky dolného viečka z dátovej sady MGD-1K

Každá snímka bola anotovaná pod dohľadom expertov na dysfunkciu meibomových žliaz a profesionálnych oftamológov. Na anotáciu bol použitý nástroj Adobe Photoshop (CC 2019). 200 z celkového počtu 1000 snímok bolo anotovaných dvoj-kolovo, dvomi rôznymi expertmi v oblasti. Tu je nutné podotknúť, že výpočtom metriky podielu prieniku a zjednotenia (prekryv) medzi dvomi expertami bola dosiahnutá úspešnosť zhody 64,03% pre segmentačné masky meibomových žliaz.

Okrem binárnych segmentačných masiek obsahuje dátová sada pre každú snímku aj hodnotenie meiboskóre, ktoré bolo určené dvomi expertmi. Každý z nich určil meiboskóre pre danú snímku v troch kolách. Takto vzniknutých 6 hodnôt pre každú snímku bolo spriemerovaných, čím bola určená výsledná hodnota. 59 snímok bolo označených ako neohodnotiteľných aspoň v jednom zo šiestich kôl. Z každej snímky sa dá tiež na základe názvu určiť, či ide o horné, alebo dolné viečko.

Doplňujúce informácie o dátovej sade sa nachádzajú v tabuľke 4.2.

Počet pacientov	320
Vek	Muži (priemer 51, štandardná odchýlka 19) Ženy (priemer 55, štandardná odchýlka 19)
Pomer muži / ženy	322 (32,2%) / 678 (67,8%)
Počet snímok horného viečka	467
Počet snímok dolného viečka	533
Farebné kanály	Jednokanálové dáta (odtiene sivej)
Doba trvania zberu dát	Apríl 2019 - Apríl 2020 (1 rok)

Tabuľka 4.2: Doplnujúce informácie o dátovej sade MGD-1K

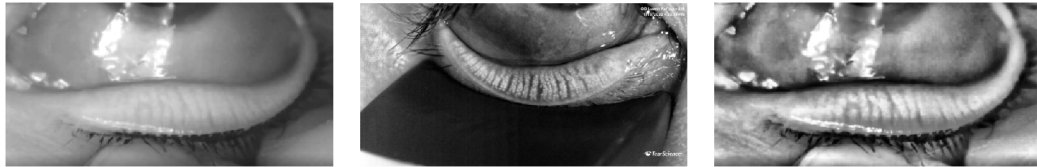
Za účelom obohatenia dátovej sady boli kontaktovaní aj autori prác [33] a [38], kde anotované dátové sady obsahovali 728 a 775 snímok. Z príslušných univerzít však nebola doručená odpoveď, a tak sú v práci na tréningovanie a vyhodnotenie použité iba dáta zo sady MGD-1K.

Predspracovanie dátovej sady

Pred samotným tréningovaním modelov neurónových sietí je dátová sada rozdelená do troch množín. Tréningovú množinu tvorí 70% dát, validačnú 10% a testovaciu 20%. Modely pre segmentáciu očného viečka sú tréňované oddelene od modelov pre segmentáciu meibomových žliaz, preto sú z pôvodnej dátovej sady vytvorené dve nové na základe segmentovanej oblasti.

Analýzou výstupných dát z kamery použitej pri tvorbe aplikácie opísanej v časti 4.1 boli zistené značné rozdiely vo vlastnostiach výstupných obrazových dát kamery a dátach

sady MGD-1K. Po aplikovaní CLAHE na snímku z prototypu kamery je vidieť, že obraz začína mať podobné vlastnosti ako dáta z trénovacej dátovej sady. Porovnanie snímok je možné vidieť na obrázku 4.4.



(a) Snímka z prototypu kamery (b) Snímka z dátovej sady MGD-1K (c) Snímka z prototypu kamery po aplikovaní CLAHE

Obr. 4.4: Porovnanie výstupu z prototypu kamery a snímky dátovej sady MGD-1K

Aby bol docielený efekt robustnosti neurónových sietí, sú na snímky v dátovej sade MGD-1K aplikované augmetačné techniky. Po zhodnotení pamätovej náročnosti offline augmentácie a relatívne nízkej časovej náročnosti online transformácií obrazu som sa rozhodol nerozširovať dátovú sadu pred trénovaním, ale vykonávať náhodné zmeny v priebehu trénovania. Viac informácií o aplikovaných technikách je popísaných v časti 4.3.

4.3 Modely neurónových sietí

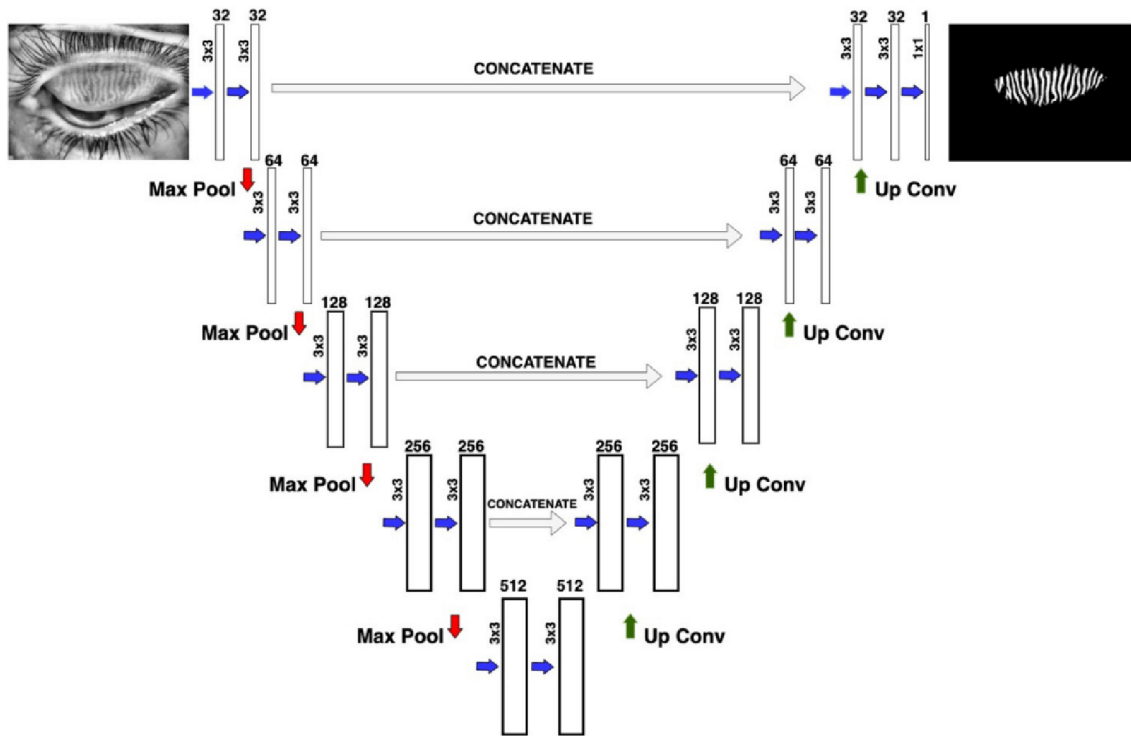
Architektúra modelov neurónových sietí

V rámci tejto práce experimentujem s dvomi typmi architektúr segmentačných neurónových sietí. Pre každý typ architektúry sú natrénované dva modely - jeden určený na segmentáciu očného viečka, druhý na segmentáciu meibomových žliaz.

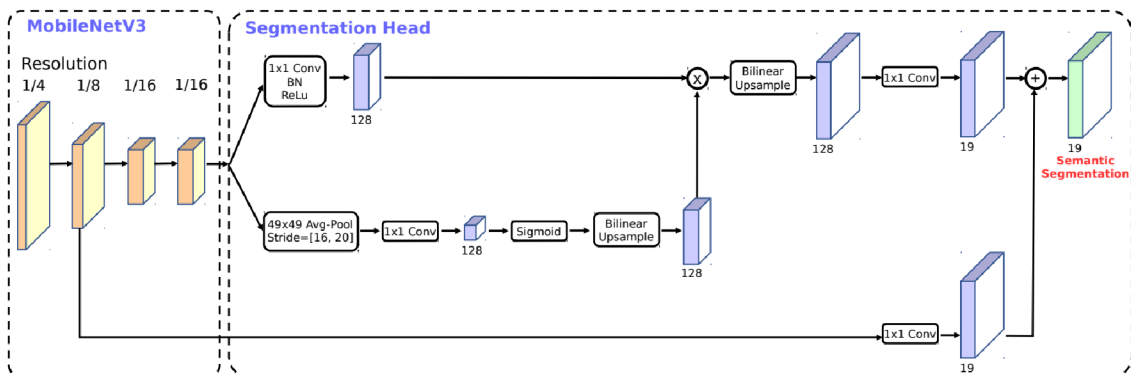
Prvým typom je adaptácia architektúry U-Net z práce [28], ktorú je možné vidieť na obrázku 4.5. Architektúra pozostáva z 19 vrstiev a je plne konvolučná. Tento typ architektúry bol zvolený na základe jej úspešnosti a rovnako aj preto, lebo spomedzi prác spomenutých v sekcii 2.4 ide o architektúru s najmenším počtom vrstiev, vďaka čomu je možné ju použiť aj na relatívne rýchlu analýzu snímok na CPU.

V sieti sú použité konvolučné filtre o rozmeroch 3x3 a 1x1 v poslednej vrstve a konvolučné filtre 2x2 pre proces expanzie na pravej strane siete. Aktivačné funkcie v celej sieti sú ReLU, okrem poslednej vrstvy, kde je použitá sigmoidálna aktivačná funkcia. Vstupom do siete sú jednokanálové obrazové dáta z kamery a jej výstupom je opäť jednokanálová segmentačná mapa viečka, alebo meibomových žliaz.

Navrhnutá architektúra neurónovej siete aj napriek tomu, že je najplytšia spomedzi všetkých súvisiacich prác, je pomerne hlboká, a teda nie je vhodná na spracovanie videa v reálnom čase. Pre tieto účely v práci experimentujem s architektúrou neurónovej siete MobileNetV3-large so segmentačnou hlavicou LR-ASPP predstavenou v práci od Howarda a spol.[16], ktorá je zobrazená na obrázku 4.6.



Obr. 4.5: Architektúra segmentačnej neurónovej siete U-Net použitá v práci od Prabhu a spol.[28]



Obr. 4.6: Architektúra segmentačnej neurónovej siete s kostrou MobileNetV3 a segmentačnou hlavicou LR-ASPP[16]

Trénovanie modelov neurónových sietí

Rovnako ako v práci [28] je na tréningovanie segmentačných sietí použitá stratová funkcia binárnej krížovej entropie. Pre všetky siete je ako optimalizačný algoritmus zvolený Adam[17]. V priebehu tréningovania sietí sú využité augmetačné techniky popísané v tabuľke 4.3. Techniky, ktoré nemajú uvedenú pravdepodobnosť sú aplikované vždy.

Na záver každej epochy je vykonaný prechod na validačnej množine. Okrem validácie na pôvodných snímkach je vykonaná aj validácia na 4 sadách augmentovaných snímkov s

Transformácia	Hodnota
Rotácia	$[-30^\circ, 30^\circ]$
Horizontálny posun	$[-0.1, 0.1]$ x šírka snímky
Vertikálny posun	$[-0.1, 0.1]$ x výška snímky
Škálovanie	$[50\% - 150\%]$
Zošikmenie	$[-20^\circ, 20^\circ]$ na osi x a y nezávisle
Zrkadlenie pozdĺž vertikálnej osi	pravdepodobnosť 50%
Elastická transformácia	- pravdepodobnosť 50% - veľkosť posunov = $[0.2, 0.5]$ - hladkosť posunov = 5
Gaussovské rozostrenie	- pravdepodobnosť 50% - veľkosť kernelu = 5×9
Zmena jasu	- pravdepodobnosť 80% - $[0.8, 1.2]$ násobku aktuálneho jasu
Zmena kontrastu	- pravdepodobnosť 80% - $[0.8, 1.2]$ násobku aktuálneho kontrastu

Tabuľka 4.3: Použité augmentačné techniky

fixnými augmentačnými konštantami. Finálna hodnota stratovej funkcie a miery prekryvu je vypočítaná so zahrnutím aj takto augmentovaných dát.

V procese tréovania je postupne znižovaná miera učenia v prípade, že nenastane počas doby trpezlivosti zlepšenie miery prekryvu na validačnej množine. Pôvodná miera učenia je vynásobená faktorom klesania miery učenia a tréovanie pokračuje na modeli s váhami siete, ktorá bola vyhodnotená ako dovedty najlepšia na validačnej množine.

Tréovacie hyperparametre boli na začiatku zvolené empiricky a ich hodnoty po optimalizácii sa nachádzajú v tabuľke 4.4. Veľkosť minidávky bola obmedzená pamäťovými reštrikciami tréovacieho prostredia.

Parameter	Nastavenie
Počiatková miera učenia	0,1
Minimálna miera učenia	0,00001
Faktor klesania miery učenia	0,5
Veľkosť minidávky (U-Net architektúra)	4
Veľkosť minidávky (MobileNet architektúra)	8
Trpezlivosť	50
Maximálny počet epoch	300

Tabuľka 4.4: Počiatočné nastavenie hyperparametrov tréovania

Evaluácia modelov neurónových sietí

Evaluácia je vykonaná na testovacej množine pre všetky natréované modely neurónových sietí. Hlavnou evaluačnou metrikou je hodnota prekryvu (anglicky *intersection over union*) so segmentačnými maskami vytvorenými expertmi.

Okrem evaluácie na výstupných segmentačných maskách, je vykonaná aj evaluácia na upravených výstupoch sietí. Pre siete produkujúce segmentačné masky očné viečka je výstup upravený tak, aby zostala zachovaná iba najväčšia súvislá časť označená za očné

viečko. Zároveň sú vyplnené prípadné medzery vo vnútri masky. Pre výstupy sietí segmentujúce meibomove žľazy je aplikované prekrytie výstupnou maskou segmentačnej siete pre očné viečko a odstránenie všetkých vysegmentovaných oblastí nachádzajúcich sa mimo tejto masky.

Aby mohol byť prístup použitý v tejto práci porovnaný aj s inými súvisiacimi prácami, je tiež pre každú snímku testovacej množiny vypočítaná hodnota poklesu meibomových žliaz podľa vzorca:

$$\text{Pokles} = \frac{\text{Plocha očného viečka} - \text{Plocha žliaz}}{\text{Plocha očného viečka}} \quad (4.1)$$

Následne je s použitím dvojvýberového t-testu s predpokladom rovnakých variancií vypočítaná p-hodnota.

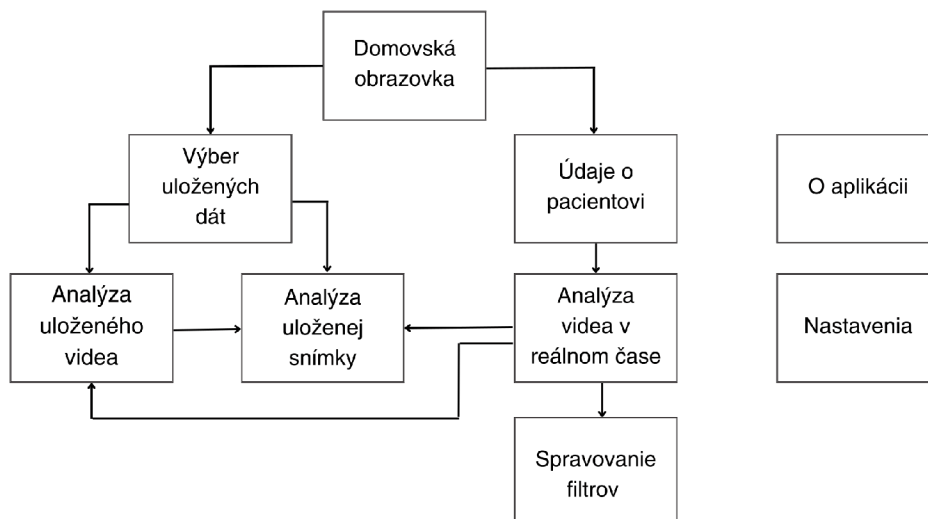
Nakoľko v prípade segmentácie očného viečka aj meibomových žliaz ide o klasifikáciu jednotlivých pixelov do určitej triedy, môžu byť pre každú snímku vypočítané aj hodnoty správnosti (anglicky *accuracy*), presnosti (anglicky *precision*), úplnosti (anglicky *recall*), F1 skóre a špecificity, ktoré môžu byť následne spriemerované pre celú testovaciu množinu a porovnané so súvisiacimi prácami.

Okrem evaluácie kvality segmentácie je vykonaná aj evaluácia priemernej doby spracovania jednej snímky neuronovou sieťou v prototypu finálnej aplikácie. Táto metrika je podstatná najmä pri experimentovaní so segmentačnými sieťami určenými na analýzu videosekvencie v reálnom čase.

V rámci evaluácie sú bližšie preskúmané nesprávne vysegmentované snímky a navrhnuté možnosti vylepšenia. Vizualne je okrem snímok datasetu MGD-1K zhodnotená aj presnosť segmentácie na dátach z prototypu kamery z časti 4.1.

4.4 Popis funkcionality aplikácie

Aplikácia pozostáva z dvoch režimov činnosti, ktorých funkcionality je predstavená v nasledujúcich dvoch sekciách. Jazyk aplikácie je anglický s možnosťou rozšírenia o ďalšie jazyky v budúcnosti.



Obr. 4.7: Obrazovky aplikácie a prechody medzi nimi

Používateľské rozhranie pozostáva z obrazoviek, ktoré sú znázornené v diagrame na obrázku 4.7. Ľavá vetva od domovskej obrazovky predstavuje časť aplikácie určenú pre spracovanie uložených dát. Pravá vetva predstavuje analýzu videa z pripojenej kamery v reálnom čase. Pohyb používateľa medzi jednotlivými obrazovkami je znázornený šípkami. Z každej obrazovky je tiež možné vrátiť sa na predchádzajúcu rodičovskú obrazovku. Rovnako je z každej obrazovky možná navigácia na domovskú obrazovku. Obrazovky *O aplikácii* a *Nastavenia* sú dostupné z ktorejkoľvek obrazovky okrem nich dvoch samotných.

Spracovanie už nasnímaných videosekvencií

V tomto móde aplikácia používateľovi umožňuje nahratie snímky vo formáte PNG alebo JPEG, alebo videa vo formáte MP4, prípadne voľbu z uložených videí a snímok vytvorených v druhom režime aplikácie.

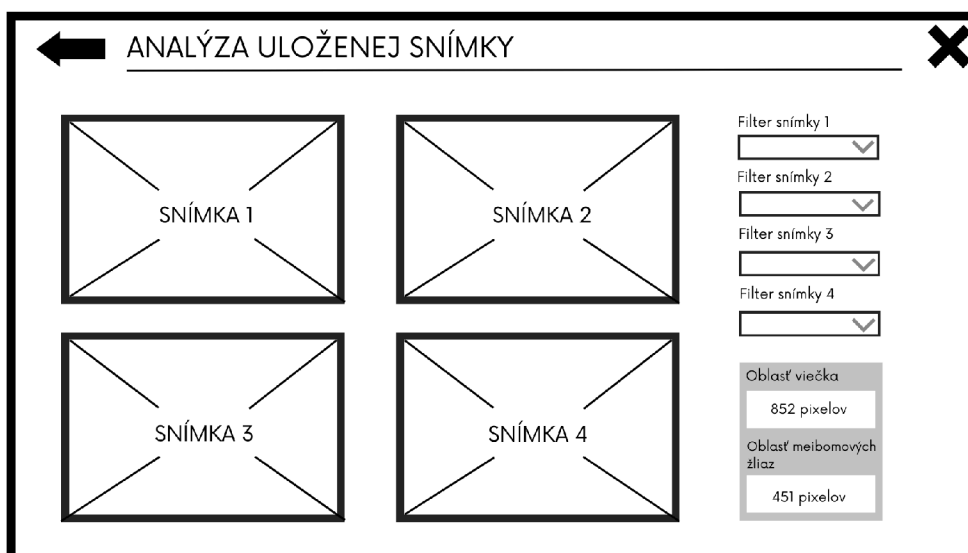
V prípade, že je zvolená na analýzu meibomových žliaz snímka, tak po kliknutí na tlačidlo *Analyzovať*, je zobrazená obrazovka *Analýza uloženej snímky*, kde je snímka zobrazená v štyroch oknách so štyrmi aplikovanými filtermi:

- Pôvodná snímka bez filtra
- Vylepšená snímka pomocou odšumenia a aplikácie CLAHE
- Vylepšená snímka pomocou odšumenia a aplikácie CLAHE prekrytá segmentačnou maskou očného viečka
- Vylepšená snímka pomocou odšumenia a aplikácie CLAHE prekrytá segmentačnou maskou meibomových žliaz

Filtere aplikované na snímky môže používateľ meniť podľa potreby. Voliť môže spomedzi všetkých filtrov uložených v aplikácii.

V prípade, že je zvolená analýza videa, zobrazí sa obrazovka *Analýza uloženého videa*. Tu môže používateľ video prehrávať, zastavovať, pretáčať, či aplikovať obrazové filtre uložené v aplikácii. Okrem toho je možné zvoliť konkrétnu snímku videa, ktorá je následne analyzovaná v obrazovke *Analýza uloženej snímky*.

Návrh obrazovky *Analýza uloženej snímky* sa nachádza na obrázku 4.8.



Obr. 4.8: Obrazovka analýzy uloženej snímky

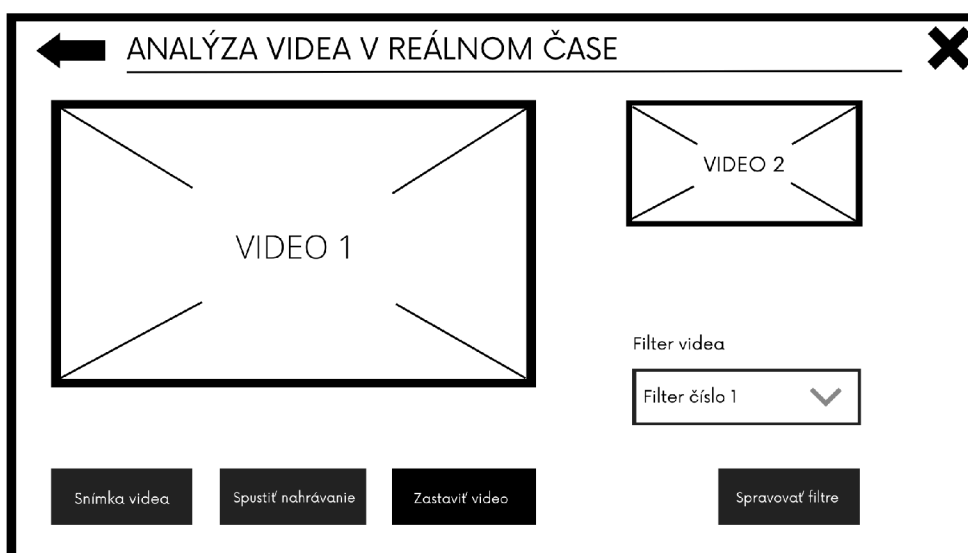
Analýza a zobrazenie vstupu z kamery v reálnom čase

Pri zvolení analýzy videa v reálnom čase sa používateľovi zobrazí obrazovka pre vyplnenie údajov o pacientovi. Tie nie sú povinné, avšak umožňujú štruktúrovanejšie ukladanie videí a snímok z aplikácie. Po kliknutí na tlačidlo *Pokračovať* sa zobrazí obrazovka *Analýza videa v reálnom čase*. Na tejto obrazovke aplikácia prijíma videosekvenciu z kamery prostredníctvom USB portu a zobrazuje ju používateľovi. Ten má možnosť video pozastaviť a znovu spustiť, uložiť snímku videa, či časť video sekvencie nahráť a uložiť. Uložené snímky či videá je možné analyzovať v druhom režime aplikácie.

Na zobrazované video môže používateľ aplikovať filtre uložené v aplikácii. Aplikácia podporuje zobrazovanie videa v dvoch oknách súčasne, pričom v každom okne môže byť aplikovaný iný filter obrazu.

Používateľ má okrem preddefinovaných filtrov možnosť definovať vlastné filtre prechodom na obrazovku *Spravovanie filtrov*. Filter môže pozostávať z jednej alebo viacerých transformácií obrazu.

Návrh obrazovky *Analýza videa v reálnom čase* sa nachádza na obrázku 4.9



Obr. 4.9: Obrazovka analýzy videa v reálnom čase

Transformácie obrazu

Aplikácia umožňuje aplikovať rôzne transformácie obrazu na snímky, uložené videá alebo aj video v reálnom čase. Jeden filter v rámci aplikácie pozostáva z jednej alebo viacerých transformácií obrazu.

Transformácie obrazu aplikovateľné na video v reálnom čase dostupné v aplikácii sú nasledovné:

- Binarizácia obrazu
- Zmena jasů
- Adaptívne vyrovnávanie histogramu s obmedzeným kontrastom - CLAHE
- Zmena kontrastu
- Gamma korekcia

- Vyrovnávanie histogramu
- Detekcia hrán
- Rozostrenie - mediánový filter
- Rozostrenie - Gaussovský filter
- Zaostrenie

Pre uložené snímky či video sú okrem vyššie spomenutých transformácií obrazu dostupné aj nasledovné:

- Odšumenie
- Segmentácia viečka modelom U-Net
- Segmentácia viečka modelom MobileNetV3
- Segmentácia meibomových žliaz modelom U-Net
- Segmentácia meibomových žliaz modelom MobileNetV3

Okrem týchto transformácií je tiež možnosť v aplikácii nastavovať hardvérové vlastnosti prototypu kamery na obrazovke *Nastavenia*, a to nasledovné:

- Jas
- Kontrast
- Zisk
- Saturácia
- Ostrosť
- Expozícia
- Rozlíšenie
- Počet snímok za sekundu

S vlastnosťami kamery sa však neodporúča manipulovať, nakoľko sú nastavené tak, aby bol výsledný obraz čo najkvalitnejší a najlepšie spracovateľný neurónovými sieťami.

Kapitola 5

Implementácia

V tejto kapitole je opísaná implementácia riešenia problému segmentácie očného viečka a jeho žliaz. Na začiatku sú priblížené použité hardvérové a softvérové prostriedky. Následne sú popísané implementačné detaily tvorby, trénovania a testovania neurónových sietí a nakoniec samotná aplikácia s používateľským rozhraním.

5.1 Použité hardvérové a softvérové prostriedky

Predspracovanie dátovej sady a modely neurónových sietí

Pre účely predspracovania dátovej sady a tvorby, trénovania a evaluácie modelov jednotlivých neurónových sietí bol zvolený jazyk Python 3.8. Modely neurónových sietí boli vytvorené a trénované s použitím frameworku PyTorch¹. Väčšina transformácií obrazu ako aj výpočet prekryvu segmentačných masiek boli vykonané s využitím knižnice torchvision². Skripty boli pripravené vo vývojovom prostredí JetBrains PyCharm³.

Samotný proces trénovania a evaluácie neurónových sietí prebiehal s využitím výpočtových prostriedkov prostredia Kaggle⁴. Na akceleráciu trénovania bol zvolený grafický procesor NVIDIA Tesla P100 s podporou CUDA a veľkosťou operačnej pamäte 16 GB. Pre účely sledovania postupu trénovania, logovania trénovacích a testovacích metrík a ukladania natrénovaných modelov bol využitý nástroj Weights & Biases⁵.

Pre vývoj aplikácie bol zvolený jazyk C# s využitím frameworku .NET 7.0 SDK⁶. Pre používateľské rozhranie bol použitý framework .NET MAUI⁷. Tento framework je podporovaný v operačných systémoch Windows, macOS, iOS a Android. Na prácu s neurónovými sieťami je využitý framework pre strojové učenie v jazyku C# ML.NET⁸. Na prácu s obrazom je využitá primárne obálka (*wrapper*) Emgu CV⁹ pre knižnicu OpenCV.

Aplikácia bola vyvíjaná vo vývojovom prostredí Visual Studio 2022 Community Edition¹⁰. Vývoj aplikácie bol vykonávaný na zariadení HP Pavilion Laptop 14-ce0xxx s pro-

¹<https://pytorch.org/>

²<https://pytorch.org/vision/stable/index.html>

³<https://www.jetbrains.com/pycharm/>

⁴<https://www.kaggle.com/>

⁵<https://wandb.ai/>

⁶<https://dotnet.microsoft.com/en-us/download/dotnet/7.0>

⁷<https://dotnet.microsoft.com/en-us/apps/maui>

⁸<https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet>

⁹https://www.emgu.com/wiki/index.php/Main_Page

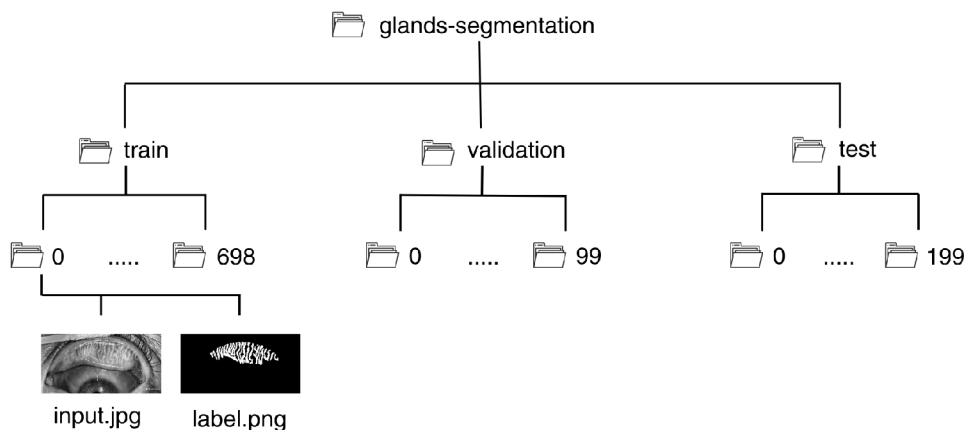
¹⁰<https://visualstudio.microsoft.com/cs/vs/community/>

cesorom Intel(R) Core(TM) i5-8250 CPU @ 1.60GHz, 1,80 GHz so 4-mi jadrami, grafickou kartou Intel(R) UHD Graphics 620 a s operačnou pamäťou počítača o veľkosti 8,0 GB s využiteľnými 7,88 GB.

5.2 Tvorba modelov neurónových sietí

Predspracovanie dátovej sady

V procese predspracovania dátovej sady bolo zistené, že dva elementy dátovej sady MGD-1K sú identické. Jeden z nich bol manuálne odstránený a celkový počet prvkov klesol na 999. Dáta sú rozdelené do troch priečinkov, z ktorých jeden obsahuje originálne snímky z meiboskopu vo formáte JPEG s rozmerom 1280 x 640 pixelov a zvyšné dva obsahujú segmentačné masky očného viečka a meibomových žliaz vo formáte PNG s rovnakými rozmermi. Masky sú previazané s pôvodnou snímku pomocou rovnakého mena súboru.



Obr. 5.1: Súborová štruktúra tréningových dát - segmentácia meibomových žliaz

V rámci skriptu *data_preprocessing.py* je najskôr vytvorená súborová štruktúra pre dve predspracované dátové sady (jedna pre segmentáciu očného viečka, druhá pre segmentáciu meibomových žliaz). Náčrt organizácie dát do priečinkov sa nachádza na obrázku 5.1. Každá predspracovaná sada obsahuje tri priečinky: *train*, *validation* a *test*. Následne je vytvorený zoznam prvkov dátovej sady a premiešaný pomocou funkcie *random.shuffle(image_names)* pre zaistenie náhodnej distribúcie prvkov. Zoznam s názvami súborov je rozdelený na tréningovú časť (699 prvkov), validačnú časť (100 prvkov) a testovaciu časť (200 prvkov). V rámci priečinka určeného na tréning siete potom vznikne 699 podpriečinkov označených číslami 0-698, z ktorých každý obsahuje dvojicu obrázkov - pôvodný obrázok a jeho segmentačnú masku pre príslušný typ dátovej sady. Rovnako postupujem pre validačnú a testovaciu časť.

Pre načítanie dát v procese tréningu a testovania je použitá trieda *Dataset* z modulu *torch.utils.data*. Sieť je tréningovaná na jednokanálových obrázkoch, ktorých hodnoty pixelov sú preškálované do intervalu od 0 po 1.

Definícia modelov neurónových sietí

Modely neurónových sietí použité v tejto práci sú vytvorené vo frameworku PyTorch. Definícia modelu využívajúceho adaptáciu U-Net architektúry sa nachádza v bloku kódu 5.1. V rámci tohto modelu je použitá metóda *double_convolution(in_channels, out_channels)*,

ktorá postupne dvakrát po sebe vykonáva 2D konvolúciu, ReLU a batch-normalizáciu. Metóda *up_convolution(in_channels, out_channels)* aplikuje 2D transponovanú konvolúciu a batch-normalizáciu a nakoniec metóda *dim_reduction(in_channels)* vykoná 2D konvolúciu s kernelom o veľkosti 1x1 a aplikuje sigmoid aktivačnú funkciu.

```
# Class representing U-Net segmentation model
class MyUNet(nn.Module):
    def __init__(self):
        super(MyUNet, self).__init__()

        # define max-pool
        self.max_pool = nn.MaxPool2d(kernel_size=(2, 2), stride=(2, 2))

        # define down-sampling layers
        self.down1 = self.double_convolution(in_channels=1, out_channels=32)
        self.down2 = self.double_convolution(in_channels=32, out_channels=64)
        self.down3 = self.double_convolution(in_channels=64, out_channels=128)
        self.down4 = self.double_convolution(in_channels=128, out_channels=256)

        # define bottleneck layers
        self.bottleneck = self.double_convolution(in_channels=256, out_channels=512)

        # define up-sampling layers
        self.up_tconv4 = self.up_convolution(in_channels=512, out_channels=256)
        self.up4 = self.double_convolution(in_channels=512, out_channels=256)
        self.up_tconv3 = self.up_convolution(in_channels=256, out_channels=128)
        self.up3 = self.double_convolution(in_channels=256, out_channels=128)
        self.up_tconv2 = self.up_convolution(in_channels=128, out_channels=64)
        self.up2 = self.double_convolution(in_channels=128, out_channels=64)
        self.up_tconv1 = self.up_convolution(in_channels=64, out_channels=32)
        self.up1 = self.double_convolution(in_channels=64, out_channels=32)

        # define dimensionality reduction layer
        self.out_layer = self.dim_reduction(in_channels=32)
```

Blok kódu 5.1: Model U-Net segmentačnej architektúry. Pri doprednom prechode sú vrstvy označené ako *down* nasledované *max_pool* vrstvou. Za nimi nasleduje časť *bottleneck* a postupné zväčšovanie výstupu vrstvami *up_tconv* striedajúcimi sa s vrstvami *up*. Model je ukončený vrstvou pre redukciu výstupných kanálov.

Na tvorbu modelu siete MobileNetV3_large so segmentačnou hlavicou Lite R-ASPP bola využitá beta verzia modelu *lraspp_mobilenet_v3_large* z modulu *torchvision.models.segmentation*. Upravený bol počet výstupných tried a na záver bola pridaná sigmoidálna aktivačná funkcia. Tento model pracuje s trojkanálovým vstupom, preto sú pôvodné jednokanálové dáta rozšírené o ďalšie dva identické kanály.

Trénovanie neurónových sietí

Trénovanie neurónových sietí je implementované v skriptoch *u-net-train.py* a *mobile-net-train.py*, ktoré sú prispôbené pre spustenie v prostredí Kaggle.

Na začiatku tréovania sú inicializované objekty triedy *torch.utils.data.DataLoader*, ktoré zabezpečujú načítanie dát minidávky a ich náhodné poradie v procese tréovania. Ako optimalizačný algoritmus je použitý *torch.optim.Adam*. Pre výpočet stratovej funkcie binár-

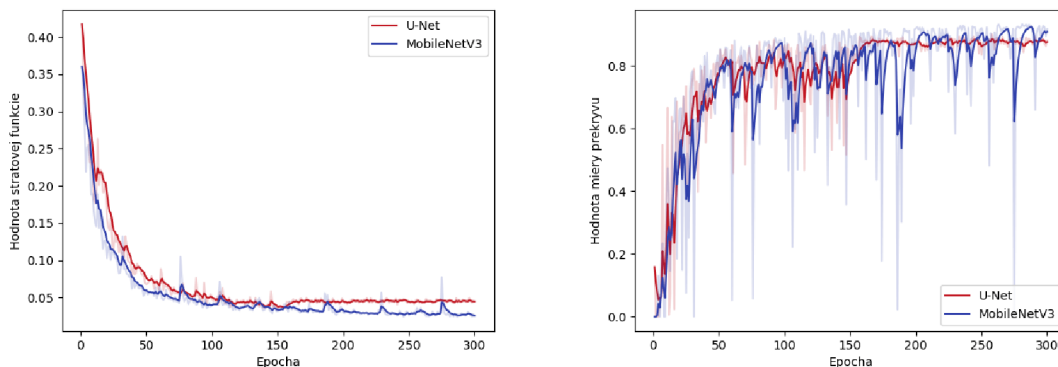
nej krížovej entropie bola zvolená jej PyTorch implementácia `torch.nn.BCELoss()`. Na zabezpečenie zníženia miery učenia v prípade dosiahnutia miery trpezlivosti, teda počtu epoch bez zlepšenia siete na validačnej metrike, je použitá trieda `torch.optim.lr_scheduler.ReduceLROnPlateau`.

V rámci učenia sa pre každú tréningovú minidávkou vykonáva dátová augmentácia za účelom vytvorenia čo najrobustnejšieho riešenia a zabránenia preučeniu. Sú využité dva typy augmenatčných techník:

- Dátová augmentácia v priestorovej oblasti:
 - `torchvision.transforms.RandomAffine` - rotácia, posun, škálovanie, sklon
 - `torchvision.transforms.RandomHorizontalFlip` - zrkadlenie pozdĺž osi y
 - `torchvision.transforms.ElasticTransform` - elastická transformácia
- Dátová augmentácia v oblasti zmien hodnôt pixelov:
 - `torchvision.transforms.GaussianBlur` - gaussovské rozostrenie
 - `torchvision.transforms.functional.adjust_brightness` - zmena jasú
 - `torchvision.transforms.functional.adjust_contrast` - zmena kontrastu

Dátová augmentácia v priestorovej oblasti musí byť s rovnakými hodnotami vykonaná ako na pôvodnej snímke oka, tak aj na príslušnej segmentačnej maske. Hodnoty pre jednotlivé augmentačné funkcie boli generované náhodne z predom stanovených rozsahov, a preto je len málo pravdepodobné, že by sieť videla identický obrázok viac než jedenkrát. Rozsahy pre hodnoty augmentačných techník sa nachádzajú v tabuľke 4.3.

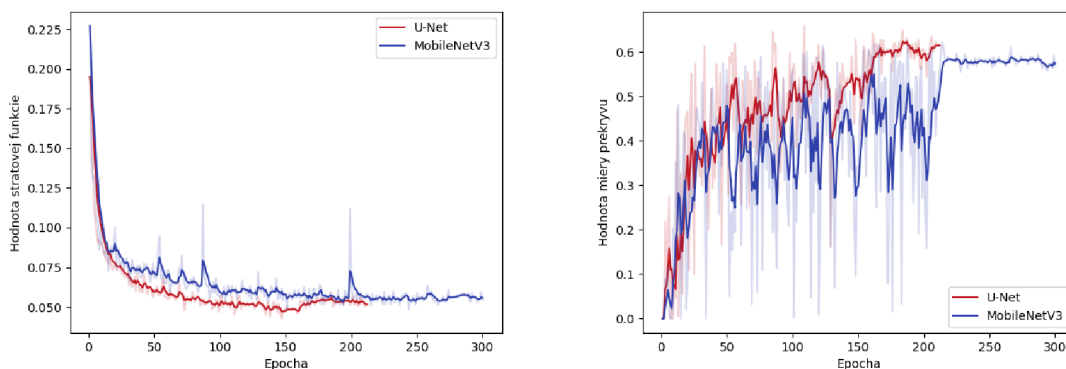
Po prechode všetkých tréningových dát v rámci jednej epochy je vykonaná evaluácia na validačnej dátovej sade. Tá sa okrem pôvodných validačných dát vykonáva aj na 4 sádach dát s predom definovanými konštantami pre dátovú augmentáciu. Do výpočtu validačnej miery prekryvu a hodnoty stratovej funkcie sú potom zahrnuté aj výsledky z týchto augmentovaných dát.



(a) Vývoj hodnôt stratovej funkcie (tréningová sada)

(b) Vývoj hodnôt miery prekryvu (validačná sada)

Obr. 5.2: Vývoj tréningových metrík počas sietí U-Net a MobileNetV3 - segmentácia očného viečka



(a) Vývoj hodnôt stratovej funkcie (trénovacia sada)

(b) Vývoj hodnôt miery prekryvu (validačná sada)

Obr. 5.3: Vývoj tréningových metrick počas sietí U-Net a MobileNetV3 - segmentácia meibomových žliaz

Priebeh tréningovania najlepšieho behu jednotlivých modelov je zaznamenaný nástrojom Weights & Biases. Vývoj hlavných tréningových metrick v priebehu tréningovania segmentácie očného viečka sa nachádza na obrázku 5.2 a segmentácie meibomových žliaz na obrázku 5.3. Detailnejšie diagramy priebehu tréningovania sú k nahliadnutiu v prílohe B.

Tréningovanie adaptácie architektúry U-Net pre segmentáciu meibomových žliaz bolo ukončené predčasne po hardvérovej chybe, avšak daný beh vyprodukoval sieť s najlepšimi výsledkami. Časy tréningovania sietí sa nachádzajú v tabuľke 5.1.

Model	Čas tréningovania
U-Net - očné viečko	22 hodín 32 minút 51 sekúnd
U-Net - meibomove žľazy	14 hodín 15 minút 53 sekúnd
MobileNetV3 - očné viečko	7 hodín 4 minúty 13 sekúnd
MobileNetV3 - meibomove žľazy	7 hodín 12 minút 5 sekúnd

Tabuľka 5.1: Časy tréningovania modelov neurónových sietí

Po ukončení tréningovania je stav, kedy sieť dosiahla najlepšiu mieru prekryvu na validačných dátach uložený a výsledný model je otestovaný na testovacej množine.

Testovanie neurónových sietí

Po uložení všetkých štyroch natréningovaných modelov je pre každý z nich spustený samostatný skript, ktorý vytvorí súbor s príponou .csv s evaluačnými metrikami. Takto vytvorený súbor obsahuje nasledovné stĺpce:

- *Index* - označuje číslo priečinka testovacej snímky
- *Area label* - počet pixelov pôvodnej masky
- *Area output* - počet pixelov vysegmentovanej masky
- *IoU* - miera prekryvu vysegmentovanej masky a pôvodnej masky
- *Accuracy* - miera správnosti klasifikácie pixelov
- *Recall* - miera úplnosti klasifikácie pixelov

- *Precision* - miera presnosti klasifikácie pixelov
- *F1 score* - F1 skóre klasifikácie pixelov
- *Specificity* - Špecificita klasifikácie pixelov

Výstupné csv súbory pre segmentačné masky očných viečok obsahujú okrem vyššie spomenutých aj stĺpec *Area - biggest* označujúci počet pixelov najväčšej súvislej časti vysegmentovaných pixelov s vyplnenými medzerami. K tomuto stĺpcu sú vypočítané príslušné metriky miery prekryvu s pôvodnou maskou, správnosť, úplnosť, presnosť, F1 skóre a špecificita.

Testovacie súbory s dátami segmentačných masiek meibomových žliaz zas navyše obsahujú stĺpce *Area - overlay U-Net* a *Area - overlay MobileNetV3* označujúce počet pixelov označujúcich žľazy, ktoré sa tiež nachádzajú pod segmentačnou maskou očného viečka vytvorenou príslušnou sieťou. Pre oba z týchto stĺpcov sú rovnako vypočítané hodnoty miery prekryvu s pôvodnou maskou, správnosť, úplnosť, presnosť, F1 skóre a špecificita.

Po vytvorení spomenutých súborov je pre každú sieť samostatným skriptom vykreslený a analyzovaný jej výstup pre najlepších a najhorších 10 snímok podľa hodnoty miery prekryvu. Výsledky evaluácie sa nachádzajú v kapitole 6.

Konverzia modelov neurónových sietí

Aby mohli byť siete natrénované vo frameworku PyTorch použité vo frameworku .NET 7.0 SDK, je potrebná ich konverzia do formátu ONNX (*Open Neural Network Exchange*)¹¹. ONNX formát je štandard uľahčujúci interoperabilitu rozličných frameworkov v oblasti modelov strojového učenia.

Konverziu podporuje aj samotný PyTorch vo funkcii *torch.onnx.export* a pre jednotlivé modely je v rámci práce implementovaná v skriptoch *convert-u-net.py* a *convert-mobile-net.py*.

5.3 Aplikácia s používateľským rozhraním

Aplikácia je implementovaná ako desktopová aplikácia pre operačný systém Windows 10 a Windows 11. Podporovaný je anglický jazyk, pričom je tu možnosť jednoduchého rozšírenia o ďalšie jazyky. Pri vývoji bol použitý návrhový vzor MVVM (*Model-View-ViewModel*)¹².

Frontend

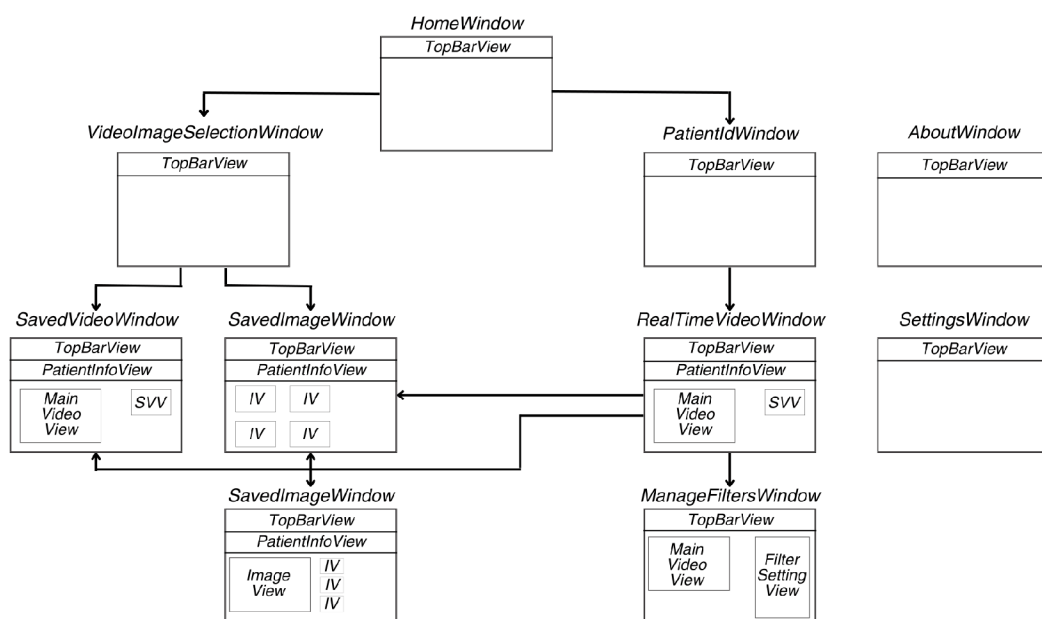
Framework .NET MAUI definuje dva typy obsahu - *ContentPage* (definuje obsah obrazovky aplikácie) a *ContentView* (definuje vlastný znovupoužiteľný UI komponent). Jednotlivé obrazovky aplikácie sú teda typu *ContentPage*, pričom môžu obsahovať rôzny počet *ContentView*. Pohyb medzi obrazovkami je implementovaný prostredníctvom triedy *NavigationPage*, ktorá si udržuje zásobník obrazoviek/stránok a umožňuje hierarchickú navigáciu. Diagram obrazoviek a ich štruktúry s nachádza na obrázku 5.4.

Každá z obrazoviek obsahuje *TextBarView* s názvom obrazovky a tromi tlačidlami - domov, o aplikácii a nastavenia. Tlačidlo späť je natívne a nachádza sa v hornej lište aplikácie, rovnako ako tlačidlo pre ukončenie aplikácie.

Časť zobrazujúca video v *MainVideoView* je implementovaná pomocou elementu *Image* s fixným pomerom strán 2:1. Okrem toho je súčasťou aj prvok *Frame* s rovnakými rozmermi

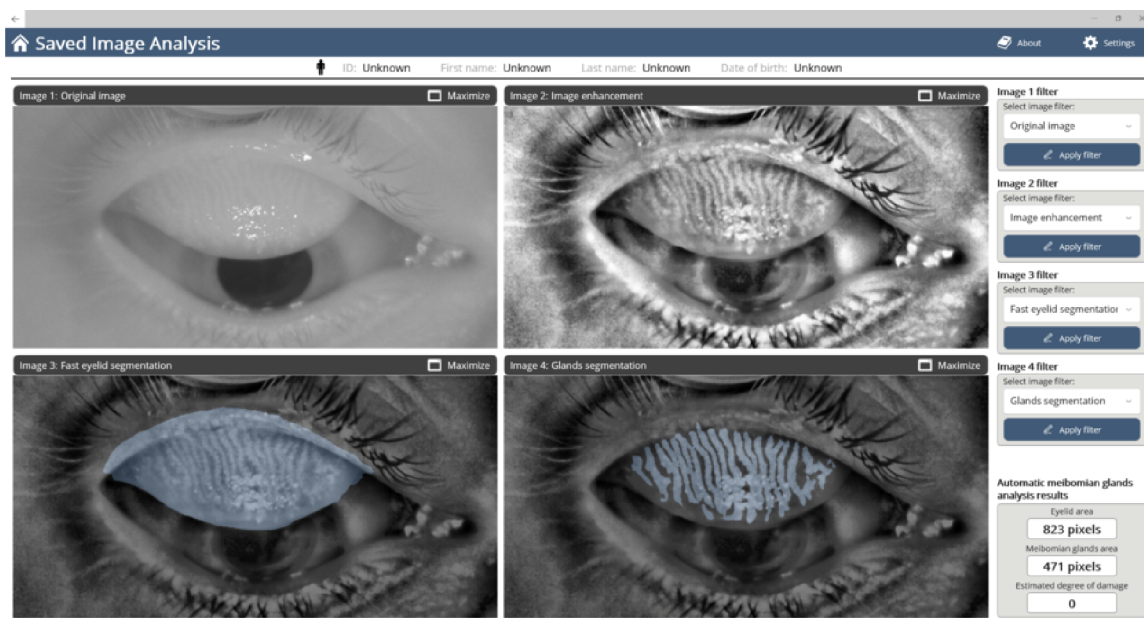
¹¹<https://onnx.ai/>

¹²<https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>



Obr. 5.4: Diagram obrazoviek a ich štruktúry. Prvky s názvami končiacimi *Window* sú typu *ContentPage*, prvky končiace *View* sú typu *ContentView*. Skratka *SVV* predstavuje *SideVideoView*, skratka *IV* znamená *ImageView*.

ako video. V prípade, že kamera nie je pripojená, práve sa pripája, alebo video je zatvorené je zobrazený práve tento prvok s príslušnou ikonkou a textom, čo zlepšuje používateľský zážitok.

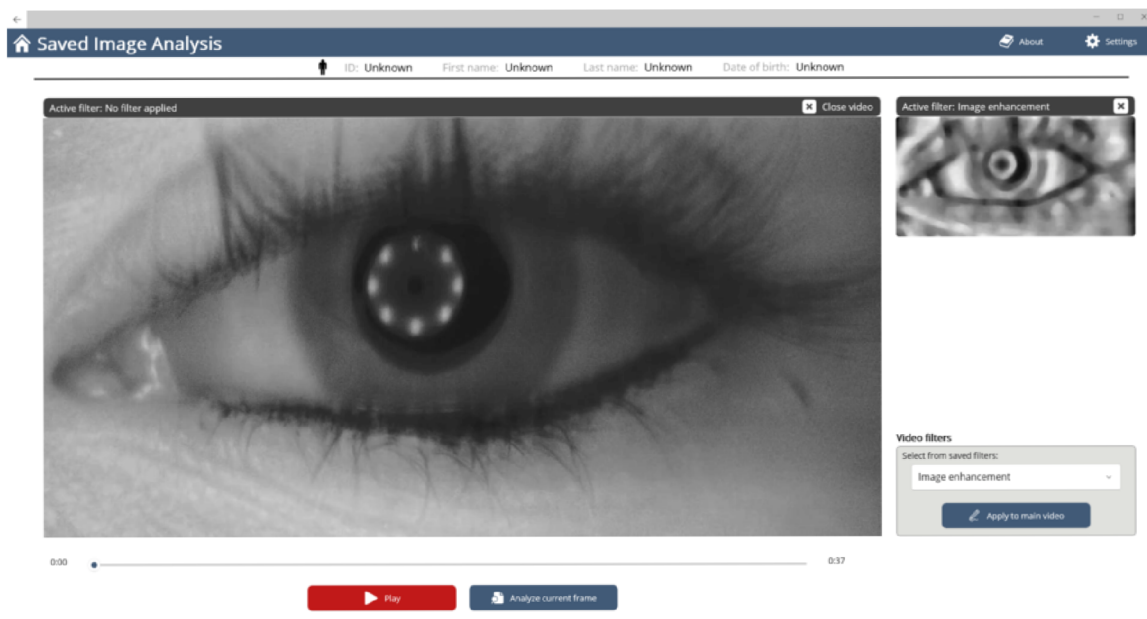


Obr. 5.5: Obrazovka analýzy uloženej snímky (*SavedImage Window*)

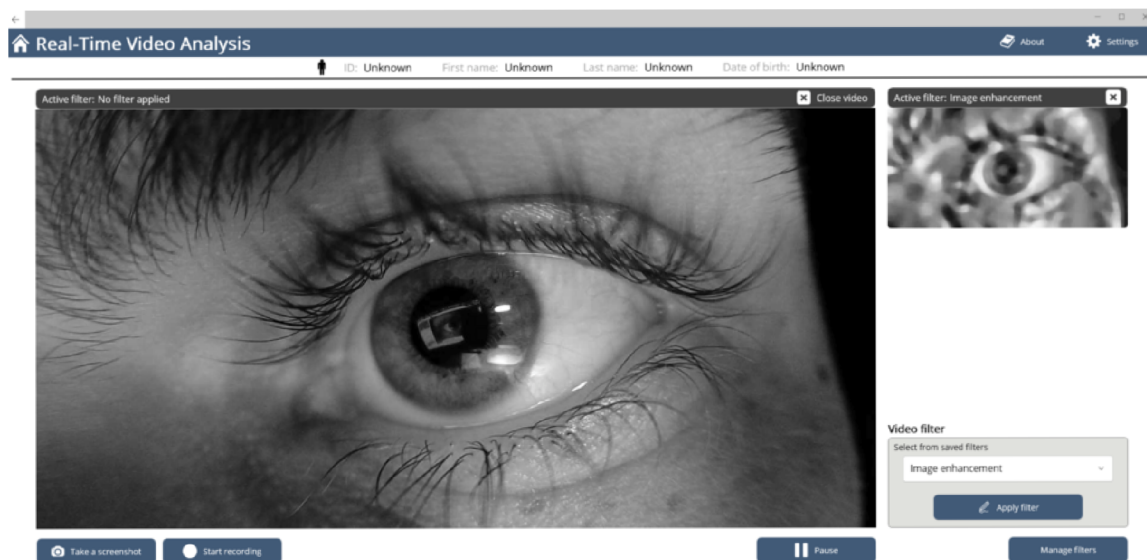
ImageView pozostáva z elementov *Image* a *Frame*, kde *Frame* s animáciou načítavania je zobrazený v prípade prebiehajúcej analýzy obrázka. Obrazovka analýzy uloženej snímky má

dve rozloženia. V prvom sa nachádzajú 4 *ImageView* rovnakej veľkosti. Po maximalizovaní jedného z nich sa tento zväčší a zvyšné tri sa usporiadajú pod seba vedľa neho, čím vznikne druhé rozloženie tejto obrazovky.

Podobným spôsobom je implementovaný aj *SideVideoView*, avšak s tým rozdielom, že video je zobrazované v elemente *ImageButton*. Po kliknutí na bočné video sa obsahy hlavného a bočného videa vymenia.



Obr. 5.6: Obrazovka analýzy uloženého videa (*SavedVideoWindow*)



Obr. 5.7: Obrazovka analýzy videa v reálnom čase (*RealTimeVideoWindow*)

Hlavnými funkčnými obrazovkami aplikácie je obrazovka analýzy uloženej snímky (*SavedImageWindow*), obrazovka analýzy uloženého videa (*SavedVideoWindow*) a obrazovka analýzy videa v reálnom čase (*RealTimeVideoWindow*). Vizualizáciu používateľského rozhrania pre tieto tri obrazovky je možné vidieť postupne na obrázkoch 5.5, 5.6 a 5.7.

Všetky obrazovky používateľského rozhrania spolu s používateľskou príručkou sa nachádzajú v prílohe A.

Backend

Každý z vyššie spomenutých *Views* obsahuje príslušný *ViewModel*. Ako je možné vidieť na obrázku 5.4, niektoré *Views* vo vnútri obsahujú ďalšie *Views*. Komunikáciu medzi nimi zabezpečujú *ViewModels* a to tak, že rodičovský *ViewModel* volá priamo metódy vnoreného *ViewModelu* a vnorený *ViewModel* odosiela rodičovskému eventy.

Identifikácia kamery a jej indexu je vykonaná s využitím knižnice *DirectShowLib*. Získavanie videa v reálnom čase z pripojenej kamery je zabezpečené pomocou triedy *Emgu.CV.VideoCapture*. Tá definuje event *ImageGrabbed*, ktorý je vystrelený pri každej prijatej snímke videa. V prípade úspešného pripojenia kamery je v novom vlákne spustená metóda na kontrolovanie jej prítomnosti v cykle. V prípade odpojenia to daná metóda zachytí, nastaví potrebné hodnoty a v novom vlákne spustí metódu, ktorá opäť v cykle kontroluje, či kamera nebola opätovne pripojená. Táto metóda sa spustí aj v prípade, že kamera nie je pripojená už od príchodu na obrazovku.

Po zachytení každého *ImageGrabbed* eventu je snímka videa načítaná do objektu *Emgu.CV.Mat* a orezaná do pomeru 2:1. Následne je konvertovaná do objektu typu *Emgu.CV.Image<Gray, byte>*, ktorý je vstupom pre prípadné transformácie obrazu aplikované na hlavné i bočné video. Výstup je transformovaný na pole bytov metódou *Emgu.CV.Image<Gray, byte>.ToJpegData()* a zobrazený.

Na nahrávanie videa je využitá trieda *Emgu.CV.VideoWriter* a na prehrávanie uložených videí opäť trieda *Emgu.CV.VideoCapture*.

Pre zabezpečenie hladšieho behu programu sú v rámci tried *RealTimeVideoViewModel*, *ManageFiltersViewModel* a *SavedVideoViewModel* implementované metódy *OnPageUnload()* a *OnPageReload()*. Metóda *OnPageUnload()* po odchode z obrazovky ukončí prímanie dát z kamery a taktiež všetky funkcie bežiacie v iných vláknach. Zníži sa tým pamäťová a výpočtová náročnosť aplikácie. Pri návrate na obrazovku z obrazovky potomka sú funkcie obrazovky obnovené zavolaním metódy *OnPageReload()*.

Každá z transformácií obrazu v aplikácii implementuje interface *IImageTransform*. Ten definuje dve metódy:

- *public Image<Gray, byte> Apply(Image<Gray, byte> inputImage)* - vstupom je snímka pred aplikáciou transformácie obrazu, výstupom je transformovaná snímka.
- *public string GetDescription()* - vracia textový popis transformácie obrazu s konkrétnymi hodnotami.

Názvy transformácií obrazu určené na analýzu videa v reálnom čase spolu s metódami ich implementácie dostupné v aplikácii sú nasledovné:

- Binarizácia - *Emgu.CV.CvInvoke.Threshold()*
- Zmena jasů - operátor +
- CLAHE - *Emgu.CV.CvInvoke.CLAHE()*
- Zmena kontrastu - *Emgu.CV.CvArray<byte>._Mul()*

- Detekcia hrán - *Emgu.CV.CvInvoke.Canny()*
- Gamma korekcia - *Emgu.CV.Image<Gray, byte>._GammaCorrect()*
- Vyrovnávanie histogramu - *Emgu.CV.Image<Gray, byte>._EqualizeHist()*
- Rozostrenie (mediánový filter) - *Emgu.CV.CvInvoke.MedianBlur()*
- Rozostrenie (Gaussovský filter) - *Emgu.CV.Image<Gray, byte>._SmoothGaussian()*
- Zaostrenie - aplikácia Laplaceovho filtra pomocou *Emgu.CV.CvInvoke.Filter2D()*

Používateľ má možnosť nastavovať parametre obrazových transformácií v obrazovke správy filtrov. Tieto transformácie tiež možno použiť pri analýze uloženého videa, či snímky.

Okrem nich je v aplikácii implementovaná redukcia šumu (*Emgu.CV.CvInvoke.FastNlMeansDenoising()*) a 4 segmentačné filtre prostredníctvom neurónových sietí.

Filtre sú ukladané do súborov typu json so štruktúrou definovanou v 5.2. Hodnota *FilterType* stanovuje typ filtra. Filtre typu *default* sú v aplikácii preddefinované a nie je ich možné odstrániť. Hodnota *FilterName* definuje meno filtra zobrazené používateľovi v aplikácii a *IsRealTimeApplicable* určuje typ filtra z hľadiska použitia na video v reálnom čase. Zoznam transformácií obrazu v poradí ako majú byť aplikované je uložený v hodnote *ImageTransforms*. Každá transformácia obsahuje svoj názov a 3 hodnoty, ktoré definujú vlastnosti danej transformácie. V prípade, že transformácia má menej vstupných parametrov ako je hodnôt v súbore filtra, sú tieto hodnoty ignorované.

```
{
  "FilterType": "default",
  "FilterName": "Eyelid segmentation – MobileNet",
  "IsRealTimeApplicable": false,
  "ImageTransforms": [
    {
      "Name": "Denoising",
      "Value1": "5",
      "Value2": null,
      "Value3": null
    },
    {
      "Name": "Clahe",
      "Value1": "18",
      "Value2": "8",
      "Value3": null
    },
    {
      "Name": "Smoothing",
      "Value1": "5",
      "Value2": null,
      "Value3": null
    },
    {
      "Name": "SegmentationMobileNet",
      "Value1": "eyelid",
      "Value2": null,
      "Value3": null
    }
  ]
}
```

Blok kódu 5.2: Príklad štruktúry súboru obsahujúceho údaje filtra

V aplikácii je 6 preddefinovaných filtrov. Prvým je vylepšenie snímky pomocou CLAHE a rozmazania Gaussovským filtrom a je aplikovateľný aj na video v reálnom čase. Ostatné filtre sú aplikovateľné iba na uložené snímky, či video. Jedným z nich je opäť filter slúžiaci predovšetkým na vylepšenie kontrastu pomocou odšumenia a CLAHE transformácie. Zvyšné 4 filtre slúžia na segmentáciu očného viečka a meibomových žliaz sieťami U-Net a MobileNetV3. Príklad filtra využívajúceho neurónovú sieť sa nachádza v ukážke súboru 5.2.

Na transformáciu snímky neurónovou sieťou je využitá trieda *Microsoft.ML.OnnxRuntime.InferenceSession*. Tá zabezpečuje načítanie ONNX modelu siete a vykonanie dopredného prechodu sieťou. Výstupom funkcie *Apply* je neupravený jednokanálový obrázok obsahujúci bielu farbu na miestach, ktoré určila sieť za oblasť záujmu a čiernu inde.

Úpravy vysegmentovanej masky sa vykonávajú metódami statickej triedy aplikácie *ImageUtils*. Pre zachovanie a vyplnenie iba najväčšej súvislej vysegmentovanej oblasti je použitá funkcia zobrazená v bloku kódu 5.3. Tá je aplikovaná na segmentačné masky očného viečka sietí U-Net i MobileNetV3.

```
public static Image<Gray, byte> GetBiggestSegmentedArea(
Image<Gray, byte> segmentationMap)
{
    // Find the contours in the segmenation map
    VectorOfVectorOfPoint contours = new();
    Mat hierarchy = new();
    CvInvoke.FindContours(segmentationMap, contours, hierarchy,
    RetrType.External, ChainApproxMethod.ChainApproxSimple);

    Image<Gray, byte> resultImage = new Image<Gray, byte>(
    segmentationMap.Width, segmentationMap.Height, new Gray(0));

    // Find the biggest contour index
    double biggestArea = 0.0;
    int contourIndex = -1;
    for (int i = 0; i < contours.Size; i++)
    {
        double area = CvInvoke.ContourArea(contours[i]);
        if (area > biggestArea)
        {
            contourIndex = i;
            biggestArea = area;
        }
    }

    CvInvoke.DrawContours(resultImage, contours, contourIndex,
    new MCvScalar(255, 0, 0), -1);

    return resultImage;
}
```

Blok kódu 5.3: Nájdenie a vyplnenie najväčšej vysegmentovanej oblasti

Pri segmentácii meibomových žliaz je vytvorená podľa vyššie spomenutého postupu aj maska očného viečka sieťou MobileNetV3. Následne je na výstupnú masku žliaz aplikovaná funkcia *Emgu.CV.CvInvoke.BitwiseAnd()* s maskou očného viečka pre odstránenie artefaktov segmentácie mimo oblasti očného viečka.

Na záver je pre každý segmentačný filter maska prevedená do farebnej trojkanálovej snímky a metódou *Emgu.CV.CvInvoke.AddWeighted()* je vytvorený prekryv s pôvodnou snímkou oka. Takto upravené výstupy pre segmentáciu viečka sieťou MobileNetV3 a segmentáciu žliaz sieťou U-Net s použitím prototypu kamery 4.1 je možné vidieť v časti Evaluácia 6 na obrázkoch 6.8 a 6.9.

Kapitola 6

Evaluácia

V tejto kapitole je zhrnutá evaluácia natrénovaných modelov neurónových sietí na testovacej časti dátovej sady MGD-1K spolu s porovnaním so súvisiacimi prácami. Rovnako sú zhodnotené i dosiahnuté výsledky na dátach prototypu kamery predstavenej v časti 4.1 a činnosť aplikácie.

6.1 Evaluácia segmentácie očného viečka

Uspešnosti sietí segmentujúcich očné viečko sú vyhodnotené dvomi spôsobmi. Prvým prístupom je výpočet metrík priamo z výstupov neurónových sietí, ďalej označovaným ako prístup "bez úprav". Druhým prístupom je úprava výstupnej segmentačnej masky tak, aby bola zachovaná iba najväčšia súvislá vysegmentovaná oblasť s vyplnením prípadných prázdnych miest vo vnútri masky. Tento prístup je ďalej označovaný ako prístup "s úpravami".

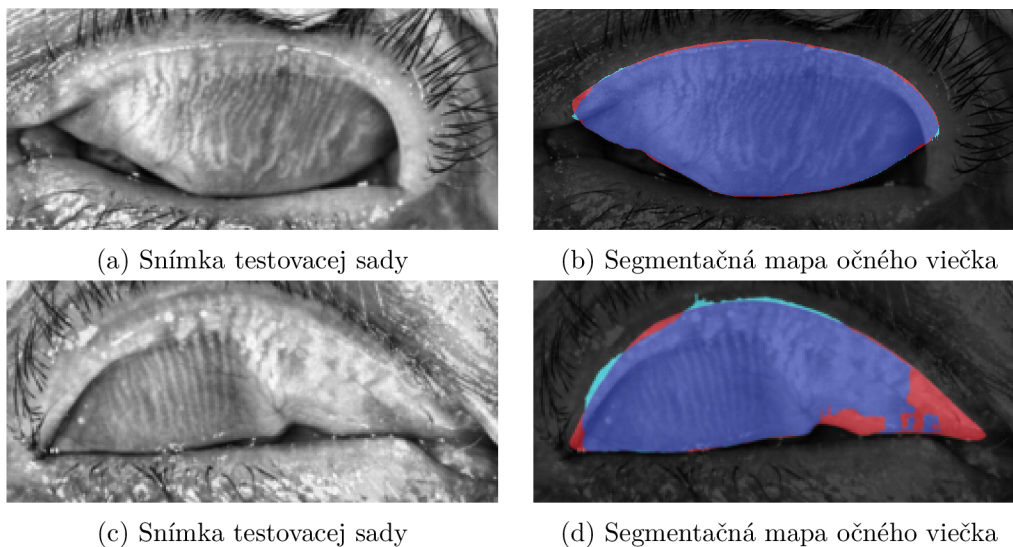
Model	Prekryv	Správnosť	Presnosť	Úplnosť	F1 skóre	Špecificita
U-Net bez úprav	90,46 %	98,28 %	96,05 %	93,97 %	94,86 %	99,21 %
U-Net s úpravami	91,63 %	98,50 %	97,27 %	94,09 %	95,52 %	99,46 %
MobileNetV3 bez úprav	93,19 %	98,80 %	96,89 %	96,08 %	96,43 %	99,35 %
MobileNetV3 s úpravami	93,61 %	98,88 %	97,34 %	96,09 %	96,66 %	99,44 %

Tabuľka 6.1: Vyhodnotenie tréningových metrík sietí pre segmentáciu očného viečka

V tabuľke 6.1 sa nachádza vyhodnotenie hlavných testovacích metrík. Z metrík vyplýva že ponechanie iba najväčšej segmentovanej oblasti má prízivý vplyv na všetky testovacie metriky. V úlohe segmentácie očného viečka dosiahla sieť MobileNetV3 lepšie výsledky ako modifikovaná architektúra U-Net.

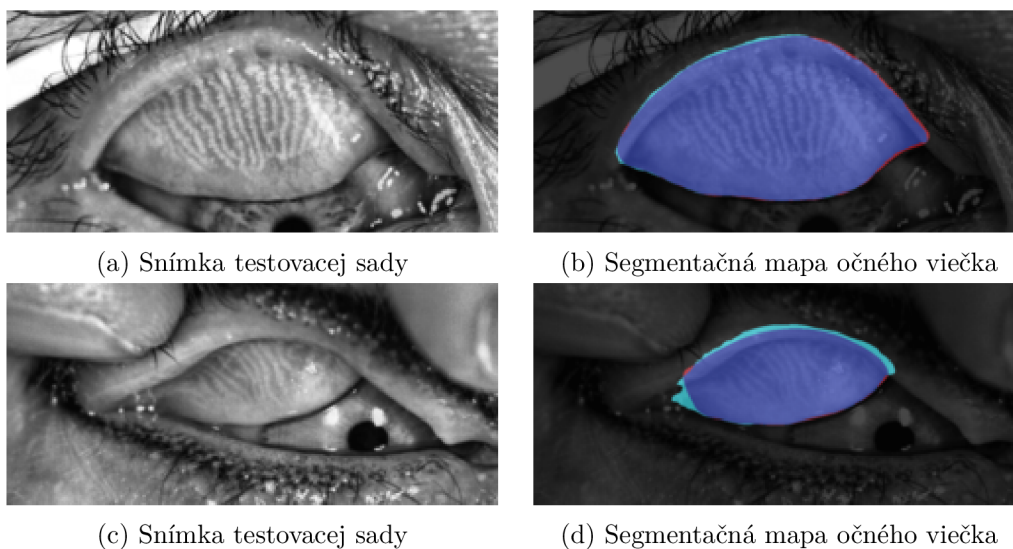
Pre lepšiu vizualizáciu výstupov boli pre každý typ siete vytvorené zobrazenia vykresľujúce segmentáciu najlepšej a najhoršej snímky testovacej množiny vzhľadom na hodnotu miery prekryvu. Pre segmentáciu očného viečka architektúrou U-Net je možné toto zobrazenie vidieť na obrázku 6.1. Obrázky 6.1a a 6.1b zobrazujú snímku a jej segmentačnú masku s najvyššou mierou prekryvu, obrázky 6.1c a 6.1d zase tie s najmenším prekryvom. Rovnaké rozloženie je použité aj v ďalších vizualizáciách spolu s rovnakou farebnou škálou. Tmavomodrou farbou sú označené oblasti, ktoré boli vysegmentované správne. Červené oblasti predstavujú časti, ktoré boli označené lekárom ako oblasť záujmu, ale sieť ich neurčila a svetlo-modré sú oblasti vysegmentované sieťou, ktoré však neboli označené lekárom.

Analýzou 10 najhorších výstupov podľa miery prekryvu pre architektúru U-Net s úpravami je možné zistiť, že sieť má najčastejšie problém s nedostatočne prevráteným, prípadne



Obr. 6.1: Segmentácia očného viečka adaptáciou architektúry U-Net s úpravami - najlepší a najhorší výstup

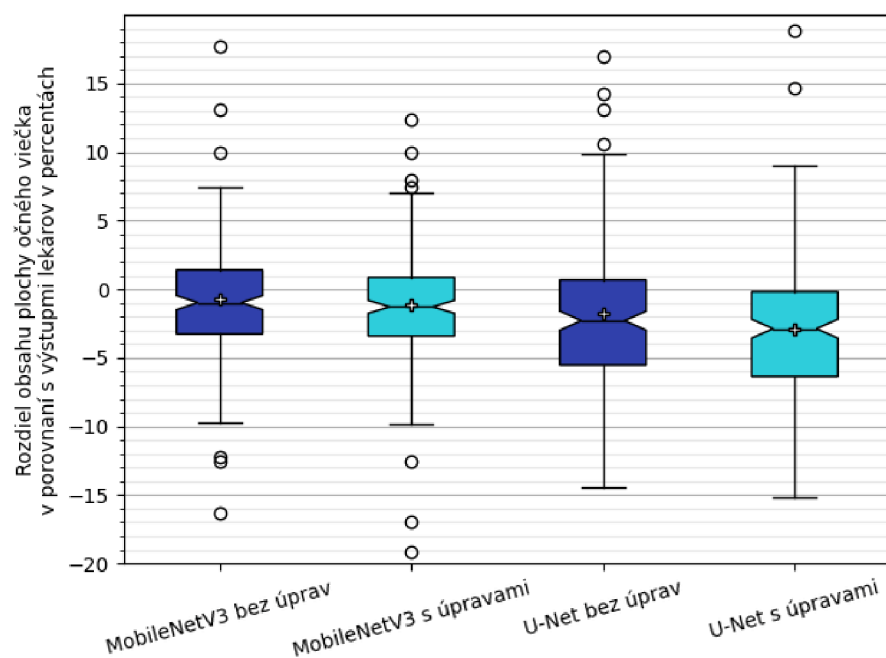
deformovaným vrchným viečkom. Iba jedna snímka z 10 obsahovala spodné viečko. Naopak sieť je najmenej chybová pri horných, či spodných viečkach s dostatočne veľkým obsahom plochy.



Obr. 6.2: Segmentácia očného viečka architektúrou MobileNetV3 s úpravami - najlepší a najhorší výstup

Zobrazenie najlepšej a najhoršej segmentácie viečka sieťou MobileNetV3 s úpravami je vidieť na obrázku 6.2. Sieť má podobné problémy ako model U-Net s úpravami. Z najlepších desiatich segmentácií však nie je ani jedna spodného viečka, z čoho možno dedukovať, že najlepšie sieť pracuje na snímkach dostatočne prevráteného horného viečka.

Okrem vyššie spomenutých metrik sú výstupy sietí porovnané aj vzhladom na obsah segmentačnej masky, nakoľko práve to zohráva najväčšiu úlohu pri správnom určení meibos-



Obr. 6.3: Percentuálne rozdiely v obsahu vysegmentovanej oblasti jednotlivých prístupov a architektúr sietí v porovnaní so segmentačnou maskou určenou lekárom

Model	Priemerný obsah meibomových žliaz (v pixeloch)	Priemerná chyba v absolútnej hodnote	Štandardná odchýlka chyby
Ohodnotenie lekárom	143 243,34	-	-
U-Net bez úprav	139 868,96	5,37 %	7,73 %
U-Net s úpravami	138 384,47	5,33 %	7,33 %
MobileNetV3 bez úprav	142 099,38	3,35 %	4,05 %
MobileNetV3 s úpravami	141 530,16	3,12 %	3,82 %

Tabuľka 6.2: Vyhodnotenie chýb obsahov segmentačných masiek sietí pre detekciu očného viečka

kóre. Z diagramu na obrázku 6.3 a tabuľky 6.2 je možné vidieť, že v priemere je výstupom siete maska s menším obsahom, ako maska určená lekárom. Najlepšie výsledky spomedzi všetkých prístupov dosahuje sieť MobileNetV3 s úpravami.

Najväčším nedostatkom sietí je aj vzhľadom na metriky presnosti a úplnosti nedostatočná detekcia viečka a teda označovanie oblastí patriacich viečku ako oblastí mimo oblasti záujmu.

Celkovo však možno zhodnotiť, že najmä sieť MobileNetV3 s úpravami dosahuje veľmi dobré výsledky blízke súčasnému stavu poznania v tejto oblasti.

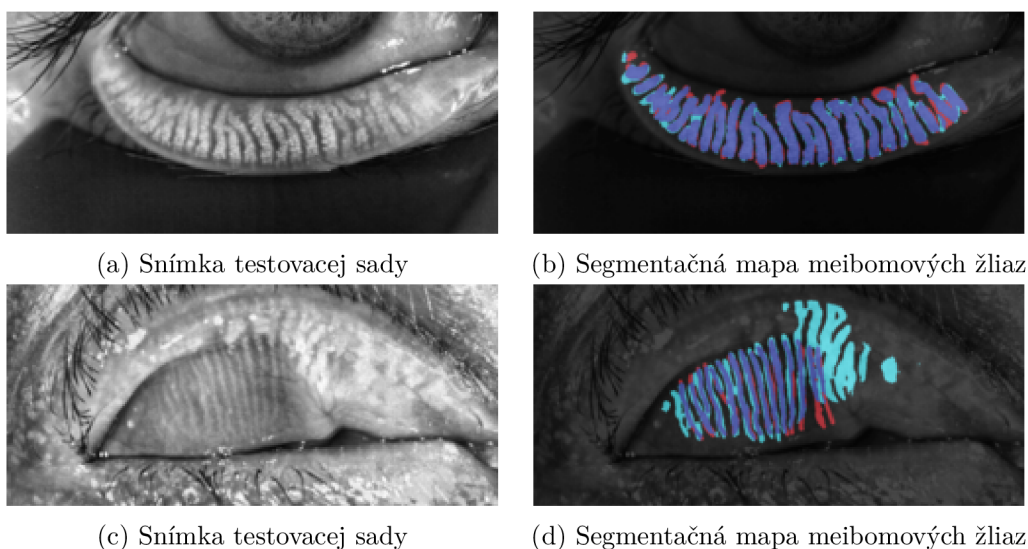
6.2 Evaluácia segmentácie meibomových žliaz

Tak ako pri segmentácii očného viečka, aj pre meibomove žľazy je použitých viacero prístupov pre každý typ siete. Prvým je použitie výstupu siete bez ďalších úprav, ďalej spomínaný ako prístup "bez úprav". Druhým je prístup, kedy sú ponechané iba oblasti prekryté výstupom siete U-Net segmentujúcou očné viečko s úpravami. Tento prístup je označený ako "prekryv s U-Net". Nakoniec posledný prístup využíva rovnaký proces, avšak so sieťou MobileNetV3 a je označený ako "prekryv s MobileNetV3".

Model	Prekryv	Správnosť	Presnosť	Úplnosť	F1 skóre	Špecificita
U-Net bez úprav	67,06 %	97,83 %	78,19 %	82,96 %	79,94 %	98,67 %
U-Net - prekryv s U-Net	68,26 %	97,95 %	80,07 %	82,90 %	80,89 %	98,80 %
U-Net - prekryv s MobileNetV3	68,33 %	97,96 %	80,12 %	82,92 %	80,95 %	98,81 %
MobileNetV3 bez úprav	64,75 %	97,67 %	77,09 %	80,91 %	78,35 %	98,65 %
MobileNetV3 - prekryv s U-Net	64,77 %	97,68 %	77,25 %	80,77 %	78,36 %	98,67 %
MobileNetV3 - prekryv s MobileNetV3	64,83 %	97,68 %	77,20 %	80,90 %	78,40 %	98,66 %

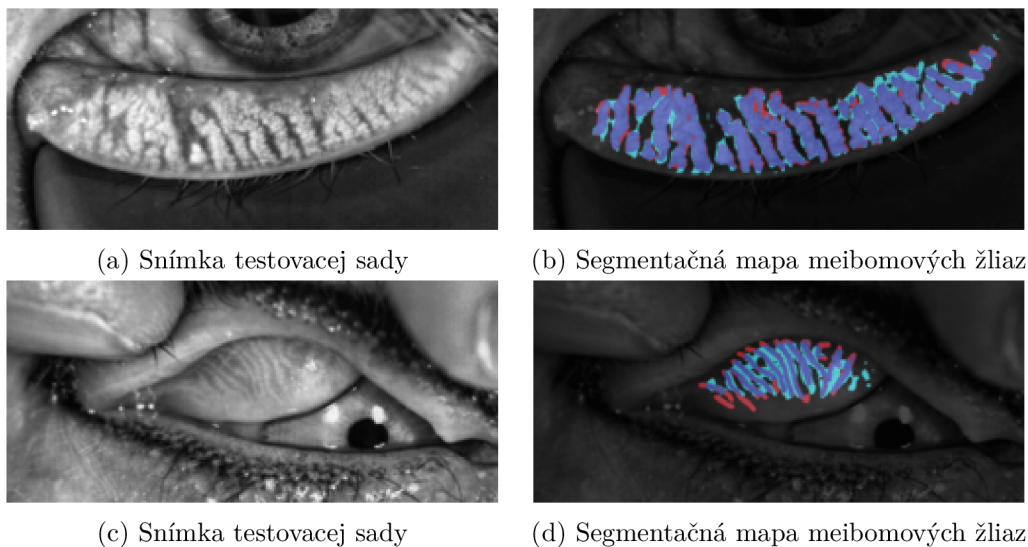
Tabuľka 6.3: Vyhodnotenie tréningových metrických sietí pre segmentáciu meibomových žliaz

Z tabuľky 6.3 je možné odpozorovať, že najlepšie výsledky sú dosiahnuté prístupom siete U-Net s prekryvom masky očného viečka siete MobileNetV3. Po prekryve mierne klesla hodnota úplnosti, čo značí nárast chyby *false negative*, teda neoznačenie pixelu obsahujúceho žľazu. S najväčšou pravdepodobnosťou sa jedná o pixely na okrajoch očného viečka. Percentuálny rozdiel je však zanedbateľný a predstavuje iba 0,04 %.

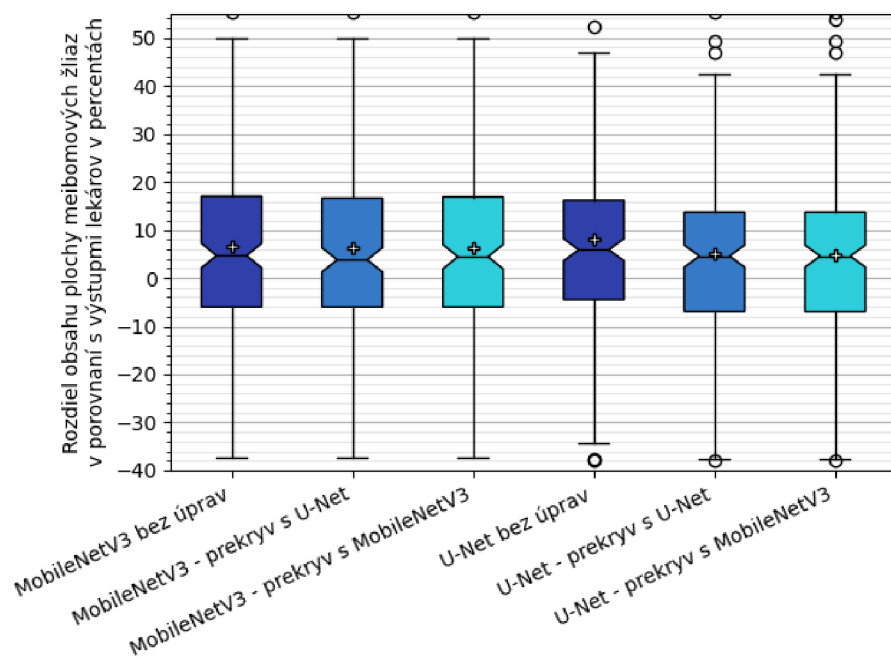


Obr. 6.4: Segmentácia meibomových žliaz adaptáciou architektúry U-Net s prekryvom MobileNetV3 - najlepší a najhorší výstup

Na obrázkoch 6.4 a 6.5 sú zobrazené najlepšie a najhoršie výstupy pre jednotlivé siete s prístupom prekryvu s MobileNetV3 vzhľadom na metriku prekryvu. Analýzou 10 najlepších a 10 najhorších výstupov sietí bolo zistené, že siete sa najlepšie vysporadúvajú s dolným viečkom, nakoľko v oboch prípadoch bolo medzi 10 najlepšími snímkami až 9 tvorených dolným viečkom. Naopak najhoršie výsledky boli dosiahnuté pre nedostatočne odklopené, prípadne deformované horné viečko. Segmentácia žliaz horného viečka tvorila 9 z 10 prípadov najhorších výsledkov siete U-Net a 8 z 10 prípadov pre sieť MobileNetV3.



Obr. 6.5: Segmentácia meibomových žliaz architektúrou MobileNetV3 s prekryvom MobileNetV3 - najlepší a najhorší výstup



Obr. 6.6: Percentuálne rozdiely v obsahu vysegmentovanej oblasti jednotlivých prístupov a architektúr sietí v porovnaní so segmentačnou maskou určenou lekárom

Z diagramu 6.6 a tabuľky 6.4 je možné zistiť, že obsah výstupov sietí je v priemere vyšší ako maska označená lekárom. Aj v tejto metrike dosahuje najlepšie výsledky prístup s použitím architektúry U-Net s prekryvom masky viečka vyprodukovanej modelom architektúry MobileNetV3.

Model	Priemerný obsah očného viečka (v pixeloch)	Priemerná chyba v absolútnej hodnote	Štandardná odchýlka chyby
Ohodnotenie lekárom	44 545,07	-	-
U-Net bez úprav	47 313,86	15,08 %	15,08 %
U-Net - prekryv s U-Net	46 288,90	14,17 %	13,27 %
U-Net - prekryv s MobileNetV3	46 225,24	13,95 %	12,04 %
MobileNetV3 bez úprav	46 383,04	15,36 %	13,50 %
MobileNetV3 - prekryv s U-Net	46 215,44	15,15 %	13,48 %
MobileNetV3 - prekryv s MobileNetV3	46 304,47	15,17 %	13,41 %

Tabuľka 6.4: Vyhodnotenie chýb obsahov segmentačných masiek sietí pre detekciu meibomových žliaz

Prístup segmentácie žliaz modelom U-Net s prekryvom s MobileNetV3 dosahuje v mnohých parametroch lepšie hodnoty ako iné práce vykonané na dátovej sade MGD-1K, či podobných dátových sadách.

6.3 Porovnanie s existujúcimi prácami

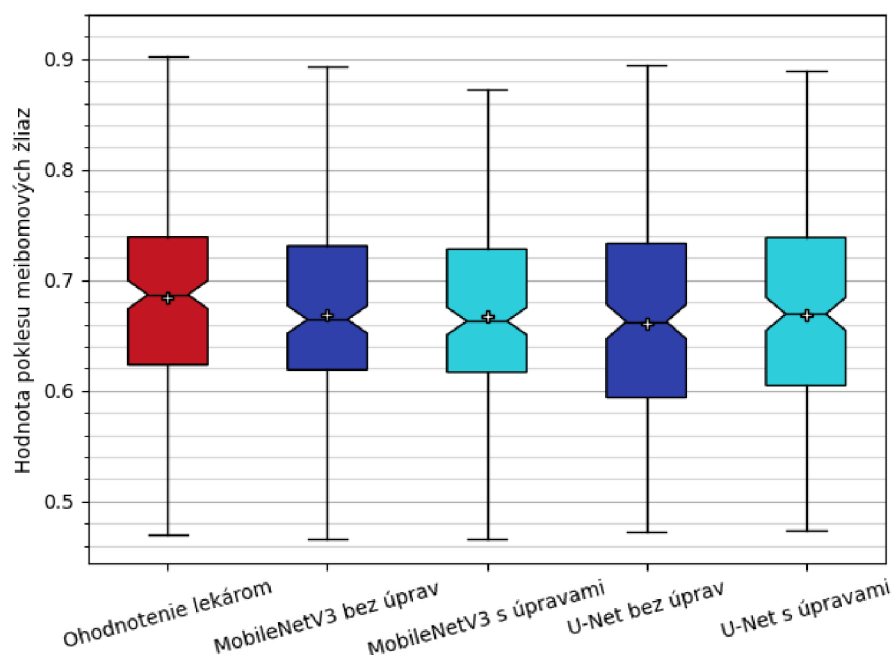
Aby bolo možné lepšie porovnanie s existujúcimi prácami, bola dvojjvýberovým t-testom s predpokladom rovnakých variancií pre hodnoty poklesu meibomových žliaz (vzorec 4.1) určená p-hodnota. Výsledky sa nachádzajú v tabuľke 6.5. Porovnané sú dva prístupy pre každú sieť. V prvom prístupe "bez úprav" je hodnota poklesu meibomových žliaz určená z obsahu segmentačnej masky žliaz danej siete bez úprav a obsahu segmentačnej masky očného viečka siete MobileNetV3 bez úprav. Druhý prístup "s úpravami" počíta s maskou žliaz po prekryve maskou očného viečka siete MobileNetV3 a maskou očného viečka siete MobileNetV3 s úpravami. Vizualizáciu rozloženia poklesu meibomových žliaz v rámci testovacej dátovej sady je možné vidieť na obrázku 6.7. Výsledky potvrdzujú, že najlepším prístupom je použitie siete U-Net pre segmentáciu žliaz s prekrytím upravenou maskou viečka vytvorenou sieťou MobileNetV3. Porovnanie so súvisiacimi prácami je vykonané práve na základe tohto prístupu. Pri hladine $\alpha = 0,05$ na základe výsledkov t-testu nie je možné zamietnuť nulovú hypotézu rovnakej strednej hodnoty.

Model	Pokles v percentách	P-hodnota
Ohodnotenie lekárom	68,39 % +- 8,25 %	-
U-Net bez úprav	66,14 % +- 8,83 %	0,0088
U-Net s úpravami	66,87 % +- 8,82 %	0,0761
MobileNetV3 bez úprav	66,81 % +- 7,87 %	0,0513
MobileNetV3 s úpravami	66,74 % +- 7,81 %	0,0412

Tabuľka 6.5: Výsledky dvojjvýberového t-testu s predpokladom rovnakých variancií

Porovnanie je vykonané s piatimi súvisiacimi prácami. Z nich však len jedna pracuje s dátovou sadou MGD-1K, a to práca autorov Saha a spol. [32]. V spomínanej práci bola dosiahnutá hodnota prekryvu pre meibomové žľazy 67,63 % v porovnaní so 68,33 % dosiahnutých v mojej práci. Dôležité je podotknúť, že hodnota prekryvu dosiahnutá segmentáciou dvoma nezávislými expertmi dosiahla výšku 64,03 %.

V práci od Prabhua a spol. [28] bola meraná p-hodnota pre pokles meibomových žliaz. Tá však bola meraná osobitne pre zdravé a poškodené očné viečka, pričom jedna sada dát obsahovala iba 50 snímok, čo prirodzene spôsobí vyššiu p-hodnotu. V priemere bola



Obr. 6.7: Hodnoty poklesu meibomových žliaz v testovacej časti dátovej sady

dosiahnutá p-hodnota 0,3485 oproti hodnote 0,0761 v mojej práci. Výsledky sú však z rozličných datasetov.

V práci od Setua a spol. [33] boli pri segmentácii meibomových žliaz dosiahnuté hodnoty presnosti, úplnosti a F1 skóre 83%, 81% a 84% v porovnaní s hodnotami 80,12 %, 82,92 % a 80,95 % dosiahnutými v mojej práci.

Práca od Wang a spol. [38] sa zaoberala segmentáciou očného viečka a atrofickej oblasti nepokrytej meibomovými žľazami. Metriky správnosti a prekryvu segmentácie viečka dosiahli hodnoty 97,6 % a 95,4 % v porovnaní s 98,88 % a 93,61 % dosiahnutými sieťou MobileNetV3 s úpravami.

V práci od Liua a spol. [22] bola testovacia množina rozdelená na dve časti - 40 zdravých viečok a 40 s dysfunkciou meibomových žliaz. Pre porovnanie s mojou prácou boli výsledky z oboch spriemerované a porovnanie sa nachádza v tabuľke 6.6.

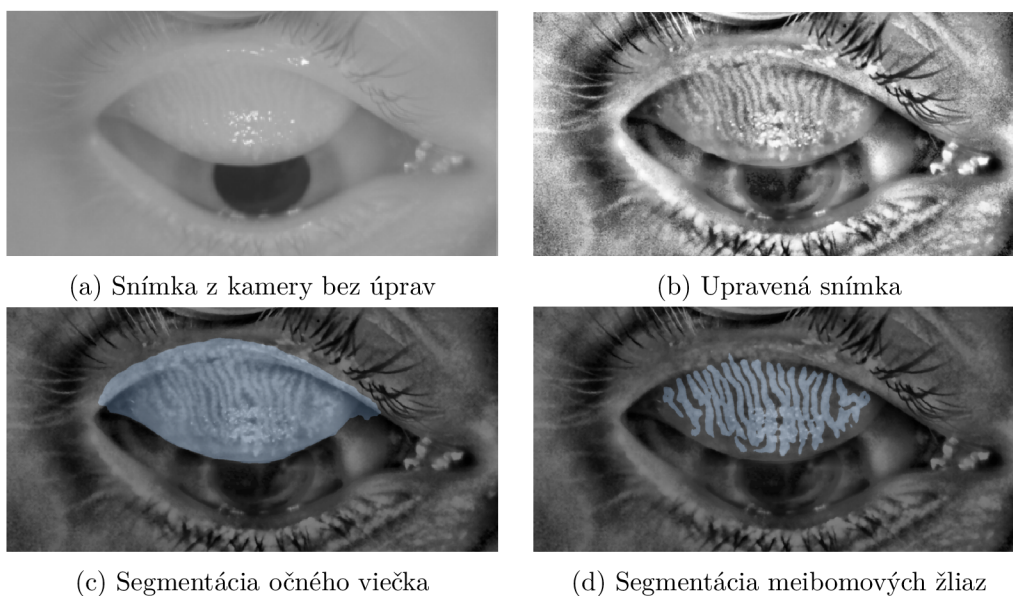
Metrika	Práca od Liua a spol.	Moja práca
Prekryv	67,55 %	68,33 %
Správnosť	81,95 %	97,96 %
Úplnosť	91,75 %	82,92 %
F1 skóre	80,95 %	80,95%
Špecifická	80,80 %	98,81 %

Tabuľka 6.6: Porovnanie s prácou od Liu a spol.

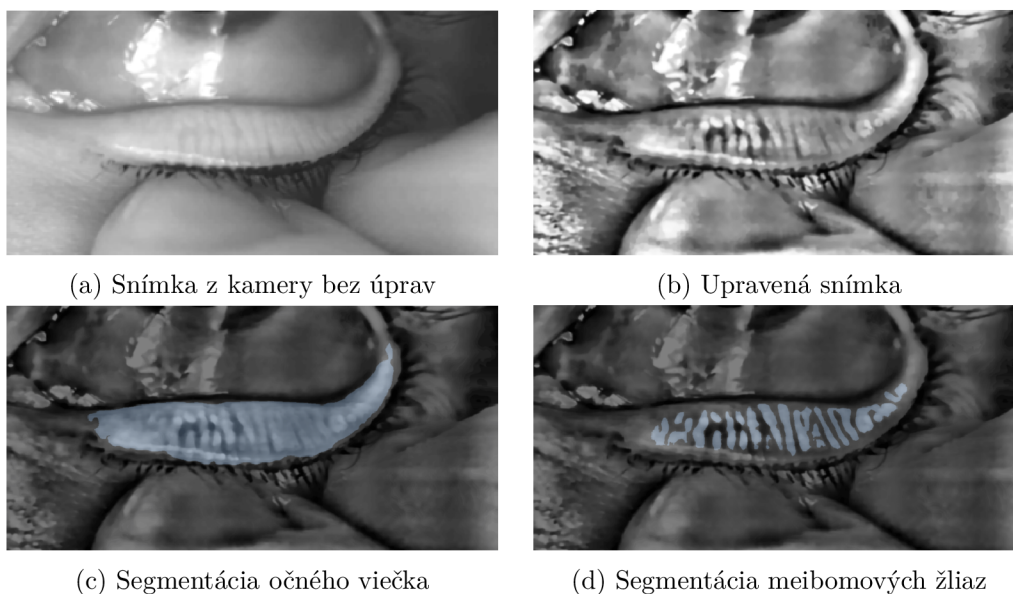
S výnimkou porovnania s prácou od Sahu a spol. [32] je dosiahnuté výsledky ťažké interpretovať v kontexte porovnania so súviciami prácami vykonanými na odlišných dátových sádach. Možno však konštatovať že sú s nimi prinajmenšom na podobnej úrovni.

6.4 Zhodnotenie funkčnosti aplikácie

Vytvorená aplikácia spĺňa všetky požadované nároky uvedené v zadaní práce a pracuje bez problémov. Jej správna funkcionálnosť bola otestovaná na piatich zariadeniach s operačným systémom Windows 10 alebo Windows 11 a nevyskytli sa žiadne závažnejšie komplikácie. V niektorých prípadoch nastal problém s prekresľovaním obrazových elementov, čo je však známa chyba frameworku .NET MAUI v operačnom systéme Windows a na jej odstránení sa pracuje.



Obr. 6.8: Analýza snímky horného viečka z prototypu kamery



Obr. 6.9: Analýza snímky dolného viečka z prototypu kamery

Analýza videa v reálnom čase je aj napriek pôvodnému zámeru aplikovania siete Mobile-NetV3 vykonávaná iba prostredníctvom obrazových filtrov, nakoľko čas segmentácie jednej snímky touto sieťou dosahuje približne 150 milisekúnd. Ide však iba o čas segmentácie. Spolu s predspracovaním dát a následným vytvorením užívateľsky príjemnej snímky sa čas predĺži na viac ako 500 milisekúnd, čo nie je použiteľné pre video v reálnom čase. Spracovanie snímky modelom U-Net trvá v aplikácii bez predspracovania a úprav po spracovaní približne 600 milisekúnd.

Výstupy analýzy snímky videa je možné vidieť na obrázkoch 6.8 pre horné viečko a 6.9 pre dolné viečko.

Výstupy z prototypu kamery obsahujú značné množstvo šumu, s čím si však vedia modely neuróných sietí poradiť. Tak isto siete dobre zvládajú aj odlesky v obraze, ktoré vznikli pri snímaní. Problém nastáva pri niektorých snímkach, ktoré obsahujú viditeľné vodorovné rôzne osvetlené pásy spôsobené infračerveným osvetlením a snímaním kamery po riadkoch. Tieto pásy sa vyskytujú s vyššou intenzitou iba v niektorých častiach videosekvencií a na analýzu je teda lepšie zvážiť použitie takých pasáží, kedy sú menej výrazné.

Aplikácia bola predstavená i oftamologickému expertovi MUDr. Tomášovi Mňukovi. Ten bol s dostupnou funkcionalitou aplikácie spokojný a spĺňala všetky jeho potreby. Jeho pripomienky sa týkali najmä kvality obrazu kamery a ovládania infračerveného osvetlenia, čo nebolo súčasťou tejto diplomevej práce.

Aplikácia obsahuje niektoré predpripravené elementy, ktorých implementácia nebola súčasťou zadania diplomovej práce a budú dopracované v budúcnosti. Rovnako nie sú doimplementované ani niektoré časti, ktorých presná špecifikácia nie je nateraz známa, nakoľko prototyp kamery neustále prechádza zmenami.

Ide o funkcionalitu obrazovky nastavení, pre ktorú je aktuálne implementovaná iba voľba pričinka pre ukladanie dát pacientov. V prípade nevyhnutnej potreby zmeny parametrov kamery je možné ich meniť priamo v konfiguračnom súbore aplikácie nachádzajúcom sa v aplikačnom priečinku na ceste */Config/AppSettings.json*.

Okrem toho zatiaľ nie je implementované ani vyhľadávanie pacientov a ich dát priamo z prostredia aplikácie, nakoľko je v budúcnosti plánovaná implementácia prostredníctvom MySQL databázy. Aktuálne má používateľ možnosť importovať súbory zo súborového systému počítača.

V obrazovke analýzy uloženej snímky nie je implementovaný výpočet plochy očného viečka a meibomových žliaz spolu s určením stupňa poškodenia, čo je ďalším plánovaným rozšírením aplikácie.

Rovnako bude v budúcnosti aplikácia pozmenená tak, aby bola kompatibilná s novým prototypom kamery pre snímanie slzných žliaz.

Kapitola 7

Záver

Cieľom tejto práce bolo preskúmať súčasný stav poznania v oblasti detekcie meibomových žliaz a segmentácie očného viečka a na jej základe vytvoriť s použitím metód strojového učenia segmentačné algoritmy zakomponované do funkčného prototypu aplikácie s používateľským rozhraním, v rámci ktorej je snímanie vykonávané prostredníctvom infračervenej kamery.

Okrem konvenčnej metódy segmentácie zdravotníckych dát architektúrou neurónovej siete U-Net bola vytvorená aj alternatívna verzia segmentácie modelom architektúry MobileNetV3_large so segmentačnou hlavicou LR_ASPP. Trénovanie neurónových sietí prebiehalo na dátovej sade MGD-1K a vytvorené boli celkovo štyri modely - dva určené na segmentáciu očného viečka a dva na segmentáciu meibomových žliaz. Výstupné segmentačné masky boli upravené extrakciou najväčšej súvislej oblasti alebo vzájomným prekryvom za účelom dosiahnutia lepších výsledkov.

Najlepší z modelov pre tvorbu masky očného viečka dosiahol hodnotu miery prekryvu s dátami anotovanými expertom na úrovni 93,61 %. Metriky správnosti, presnosti, úplnosti, F1 skóre a špecificity dosahovali postupne hodnoty 98,88 %, 97,34 %, 96,09 %, 96,66 % a 99,44 %. Najlepší model segmentácie meibomových žliaz dosiahol pre spomínané metriky postupne hodnoty 68,33 %, 97,96 %, 80,12 %, 82,92 %, 80,95 % a 98,81 %. V porovnaní s jedinou prácou [32] vykonanej na rovnakej dátovej sade bola dosiahnutá mierne vyššia hodnota prekryvu v úlohe segmentácie meibomových žliaz. Natrénované modely vykazujú na základe vizuálneho vyhodnotenia dobré výsledky aj na prototypu kamery použitej vo výslednej aplikácii.

Aplikácia bola vyvinutá s použitím frameworku .NET 7.0 a grafického rozhrania .NET MAUI. Pre účely práce s obrazom a videom slúžila najmä knižnica OpenCV s jej obálkou pre jazyk C# - Emgu CV.

Pokračovaním tejto práce by mohlo byť experimentovanie s ďalšími relevantnými segmentačnými architektúrami neurónových sietí, prípadne rozšírenie algoritmov o vyhodnotenie meiboskóre, či meiboznámky na základe výstupných masiek meibomových žliaz a očného viečka. Rovnako je plánované postupné rozširovanie vyvinutej aplikácie najmä o softvérové ovládanie vlnovej dĺžky infračerveného osvetlenia a zlepšovanie kvality výstupného obrazu prototypu kamery.

Literatúra

- [1] ARITA, R. *Meibography: A Japanese Perspective*. 2018. Dostupné z: <https://pubmed.ncbi.nlm.nih.gov/30481806/>.
- [2] ARITA, R., ITOH, K., INOUE, K. a AMANO, S. *Noncontact Infrared Meibography to Document Age-Related Changes of the Meibomian Glands in a Normal Population*. 2008. Dostupné z: <https://www.sciencedirect.com/science/article/abs/pii/S0161642007007282>.
- [3] ARITA, R., ITOH, K., MAEDA, S., MAEDA, K. a AMANO, S. *A Newly Developed Noninvasive and Mobile Pen-Shaped Meibography System*. 2012. Dostupné z: https://journals.lww.com/corneajrnl/Abstract/2013/03000/A_Newly_Developed_Noninvasive_and_Mobile.4.aspx.
- [4] BADRINARAYANAN, V., KENDALL, A. a CIPOLLA, R. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. 2017. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7803544>.
- [5] CALL, C. B., WISE, R. J., HANSEN, M. R., CARTER, K. D. a ALLEN, R. C. *In Vivo Examination of Meibomian Gland Morphology in Patients With Facial Nerve Palsy Using Infrared Meibography*. 2012. Dostupné z: https://journals.lww.com/op-rs/Abstract/2012/11000/In_Vivo_Examination_of_Meibomian_Gland_Morphology.3.aspx.
- [6] CELIK, T., LEE, H. K., PETZNICK, A. a TONG, L. *Bioimage informatics approach to automated meibomian gland analysis in infrared images of meibography*. 2013. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1888429613000629>.
- [7] DRIVER, P. J. a LEMP, M. A. *Meibomian Gland Dysfunction*. 1996. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0039625796800646>.
- [8] DUCHI, J., HAZAN, E. a SINGER, Y. *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*. 2011. Dostupné z: https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf?source=post_page.
- [9] EVERINGHAM, M., VAN GOOL, L., K. I. WILLIAMS, C., WINN, J. a ZISSERMAN, A. *The PASCAL Visual Object Classes (VOC) Challenge*. 2010. Dostupné z: <https://link.springer.com/article/10.1007/s11263-009-0275-4>.
- [10] GHOSH, S., DAS, N., DAS, I. a MAULIK, U. *Understanding Deep Learning Techniques for Image Segmentation*. 2019. Dostupné z: <https://dl.acm.org/doi/abs/10.1145/3329784>.

- [11] GIRSHICK, R. *Fast R-CNN*. 2015. Dostupné z: https://openaccess.thecvf.com/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf.
- [12] GIRSHICK, R., DONAHUE, J., DARELL, T. a MALIK, J. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. Dostupné z: https://openaccess.thecvf.com/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf.
- [13] GOODFELLOW, I., BENGIO, Y. a COURVILLE, A. *Deep Learning*. MIT Press, 2016. Dostupné z: <http://www.deeplearningbook.org>.
- [14] HE, K., GKIOXARI, G., DOLLAR, P. a GIRSHICK, R. *Mask R-CNN*. 2017. Dostupné z: https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf.
- [15] HE, K., ZHANG, X., REN, S. a SUN, J. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*. 2015. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7005506>.
- [16] HOWARD, A., SANDLER, M., CHU, G., CHEN, L.-C., CHEN, B. et al. *Searching for MobileNet V3*. 2019. Dostupné z: <https://arxiv.org/pdf/1905.02244v5.pdf>.
- [17] KINGMA, D. P. a BA, J. L. *ADAM: A Method For Stochastic Optimization*. 2015. Dostupné z: <https://arxiv.org/pdf/1412.6980.pdf>.
- [18] KOH, Y. W., CELIK, T., LEE, H. K., PETZNIK, A. a TONG, L. *Detection of meibomian glands and classification of meibography images*. 2012. Dostupné z: <https://www.spiedigitallibrary.org/journals/journal-of-biomedical-optics/volume-17/issue-8/086008/Detection-of-meibomian-glands-and-classification-of-meibography-images/10.1117/1.JBO.17.8.086008.full>.
- [19] KRAHENBUHL, P. a KOLTUN, V. *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*. 2011. Dostupné z: https://proceedings.neurips.cc/paper_files/paper/2011/file/beda24c1e1b46055dff2c39c98fd6fc1-Paper.pdf.
- [20] KRIZHEVSKY, A., SUTSKEVER, I. a E. HINTON, G. *ImageNet Classification with Deep Convolutional Neural Networks*. 2017. Dostupné z: <https://dl.acm.org/doi/pdf/10.1145/3065386>.
- [21] LITJENS, G., TOTH, R., VEN, W. van de, HOEKS, C., KERKSTRA, S. et al. *Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge*. 2014. Dostupné z: <https://www.sciencedirect.com/science/article/abs/pii/S1361841513001734>.
- [22] LIU, X., WANG, S., ZHANG, Y. a YUAN, Q. *Scribble-Supervised Meibomian Glands Segmentation in Infrared Images*. 2022. Dostupné z: https://dl.acm.org/doi/full/10.1145/3497747?casa_token=Bf8Zi33reVEAAAAA%3AgoQ62xz28xlfyz3RGSy-gLvlevQEHA52k8MIZM0IPfrjdgVHmZ8n21lg4TFqN-4cEJmrWD1VU02hMw.
- [23] LONG, J., SHELHAMER, E. a DARRELL, T. *Fully Convolutional Networks for Semantic Segmentation*. 2015. Dostupné z:

- https://openaccess.thecvf.com/content_cvpr_2015/papers/Long_Fully_Convolutional_Networks_2015_CVPR_paper.pdf.
- [24] MALHOTRA, P., GUPTA, S., KOUNDAL, D., ZAGUIA, A. a ENBEYLE, W. *Deep Neural Networks for Medical Image Segmentation*. 2022. Dostupné z: <https://www.hindawi.com/journals/jhe/2022/9580991/>.
- [25] MATHERS, W. D., SHIELDS, W. J., SACHDEV, M. S., PETROLL, W. M. a JESTER, J. V. *Meibomian Gland Dysfunction in Chronic Blepharitis*. 1991. Dostupné z: https://journals.lww.com/corneajrnl/Abstract/1991/07000/Meibomian_Gland_Dysfunction_in_Chronic.1.aspx.
- [26] MILETARI, F., NAVAB, N. a AHMADI, S.-A. *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation*. 2016. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/7785132>.
- [27] NICHOLS, K. K., FOULKS, G. N., BRON, A. J., GLASGOW, B. J., DOGRU, M. et al. *The International Workshop on Meibomian Gland Dysfunction: Executive Summary*. 2011. Dostupné z: <https://iovs.arvojournals.org/article.aspx?articleid=2126267>.
- [28] PRABHU, S. M., CHAKIAT, A., S, S., VUNNAVA, K. P. a SHETTY, R. *Deep learning segmentation and quantification of Meibomian glands*. 2020. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S174680941930357X>.
- [29] REN, S., HE, K., GIRSHICK, R. a SUN, J. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015. Dostupné z: https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.
- [30] RIPLEY, D. L. a POLITZER, T. *Vision Disturbance after TBI*. 2010. Dostupné z: https://www.researchgate.net/profile/Thomas-Politzer/publication/49630104_Vision_disturbance_after_TBI/links/0fcfd50b752045b209000000/Vision-disturbance-after-TBI.pdf.
- [31] RONNEBERGER, O., FISCHER, P. a BROX, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.
- [32] SAHA, R. K., CHOWDHURY, A. M., NA, K.-S., HWANG, G. D., EOM, Y. et al. *Automated quantification of meibomian gland dropout in infrared meibography using deep learning*. 2022. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1542012422000519>.
- [33] SETU, M. A. K., HORSTMANN, J., SCHMIDT, S., STERN, M. E. a STEVEN, P. *Deep learning-based automatic meibomian gland segmentation and morphology assessment in infrared meibography*. 2021. Dostupné z: <https://www.nature.com/articles/s41598-021-87314-8>.
- [34] SIMARD, P. Y., STEINKRAUS, D. a PLATT, J. C. *Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis*. 2003. Dostupné z: https://www.researchgate.net/profile/John-Platt-2/publication/2880624_Best_Practices_for_Convolutional_Neural_Networks/links/00b49524c79b1afb07000000/Best-Practices-for-Convolutional-Neural-Networks.pdf.

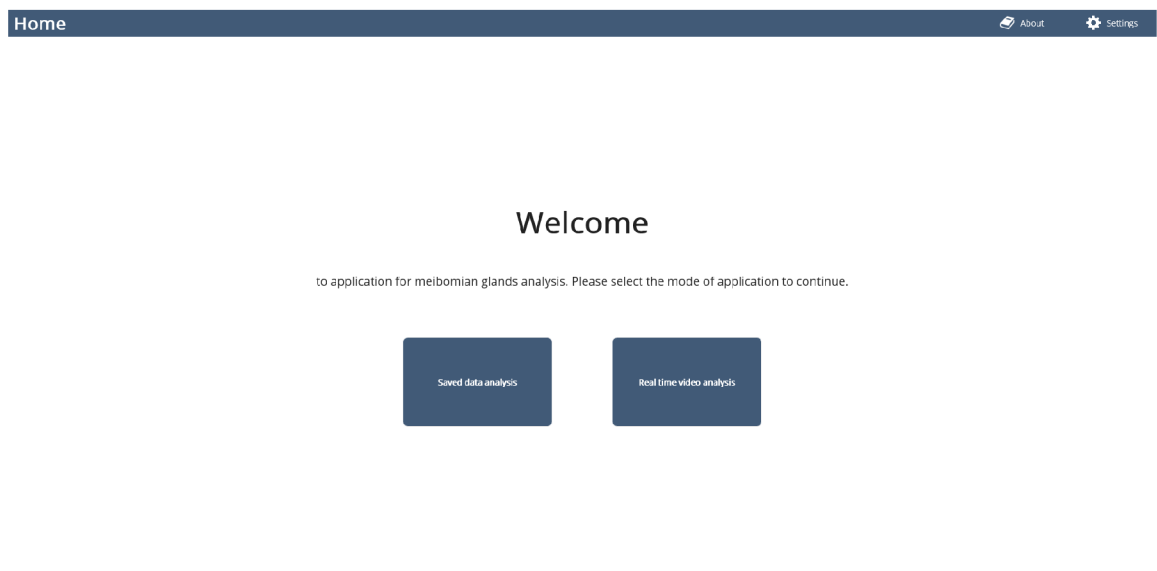
- [35] SIMONYAN, K. a ZISSERMAN, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. Dostupné z: <https://arxiv.org/pdf/1409.1556.pdf>.
- [36] SWIDERSKA, K., READ, M. L., BLACKIE, C. A., MALDONADO CODINA, C. a MORGAN, P. B. *Latest developments in meibography: A review*. 2022. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1542012422000477>.
- [37] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S. et al. *Going Deeper with Convolutions*. 2015. Dostupné z: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf.
- [38] WANG, J., YEH, T. N., CHAKRABORTY, R., YU, S. X. a LIN, M. C. *A Deep Learning Approach for Meibomian Gland Atrophy Evaluation in Meibography Images*. 2019. Dostupné z: <https://tvst.arvojournals.org/article.aspx?articleid=2757833>.
- [39] WISE, R. J., SOBEL, R. K. a ALLEN, R. C. *Meibography: A review of techniques and technologies*. 2012. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1319453412001130>.
- [40] XIAO, P., LUO, Z., DENG, Y., WANG, G. a YUAN, J. *An automated and multiparametric algorithm for objective analysis of meibography images*. 2021. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7930676/>.
- [41] ZHANG, J., FENG, H., LUO, Q., WEI, J. a LI, J. *Oil Spill Detection in Quad-Polarimetric SAR Images Using an Advanced Convolutional Neural Network based on SuperPixel Model*. 2020. Dostupné z: https://www.researchgate.net/publication/340020703_Oil_Spill_Detection_in_Quad-Polarimetric_SAR_Images_Using_an_Advanced_Convolutional_Neural_Network_Based_on_SuperPixel_Model.
- [42] ZHAO, H., SHI, J., QI, X., WANG, X. a JIA, J. *Pyramid Scene Parsing Network*. 2017. Dostupné z: https://openaccess.thecvf.com/content_cvpr_2017/papers/Zhao_Pyramid_Scene_Parsing_CVPR_2017_paper.pdf.

Príloha A

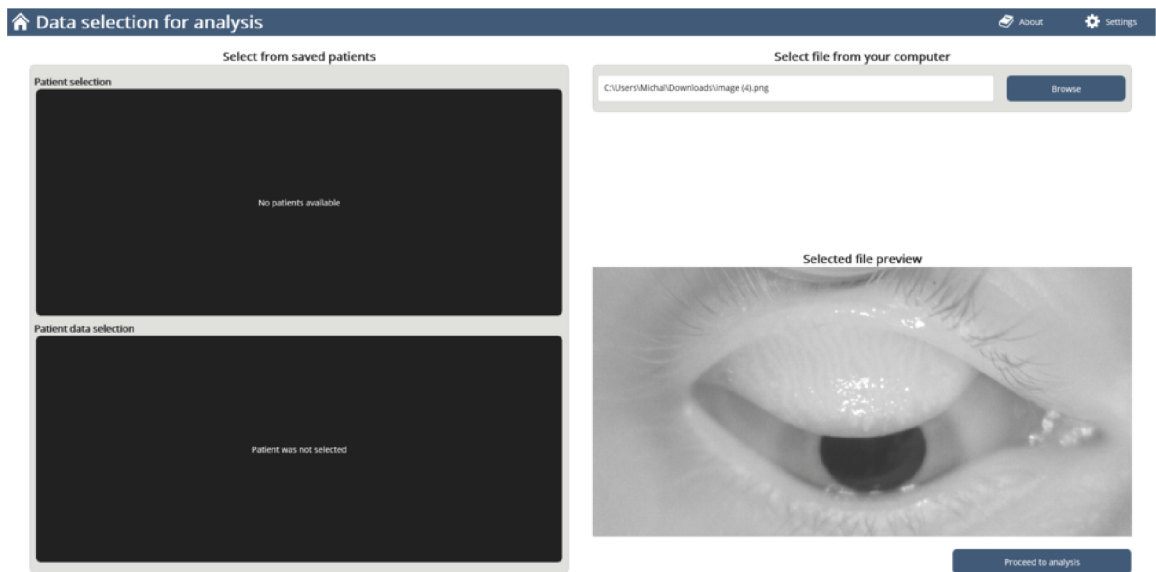
Prehľad vlastností aplikácie

A.1 Používateľské rozhranie aplikácie

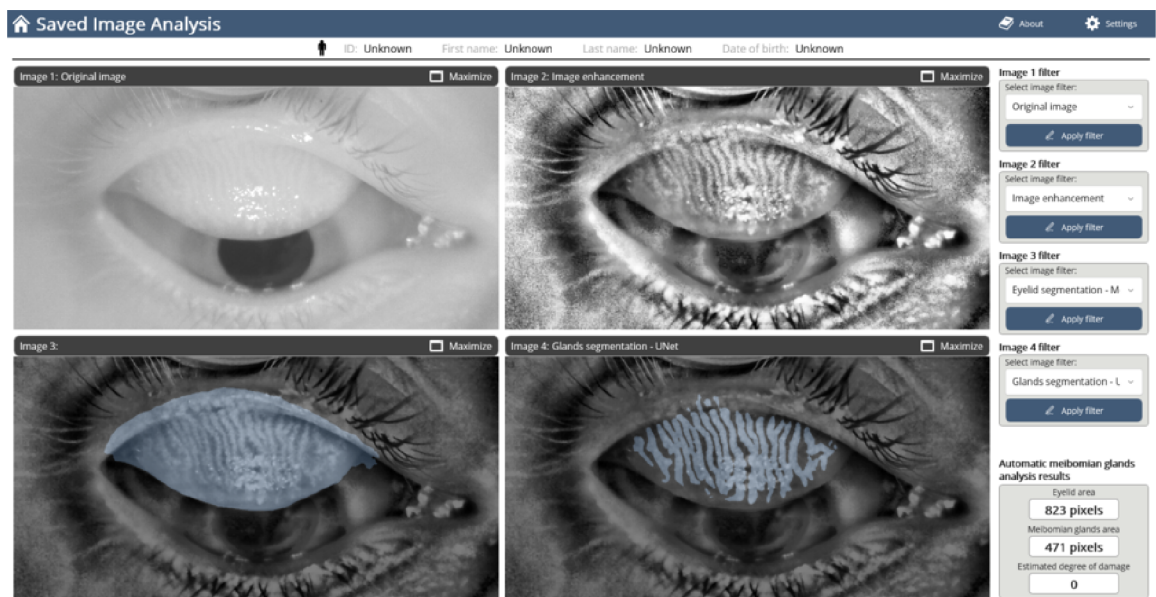
Používateľské rozhranie aplikácie pozostáva z 9 obrazoviek, ktoré sú znázornené na obrázkoch nižšie.



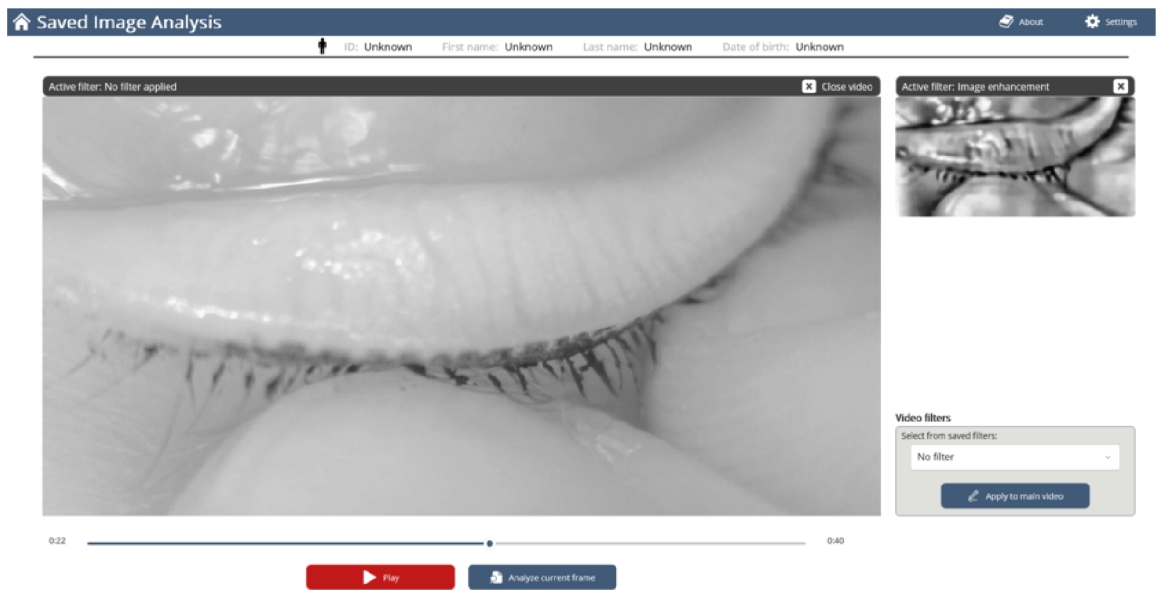
Obr. A.1: Domovská obrazovka



Obr. A.2: Obrazovka výberu z uložených dát

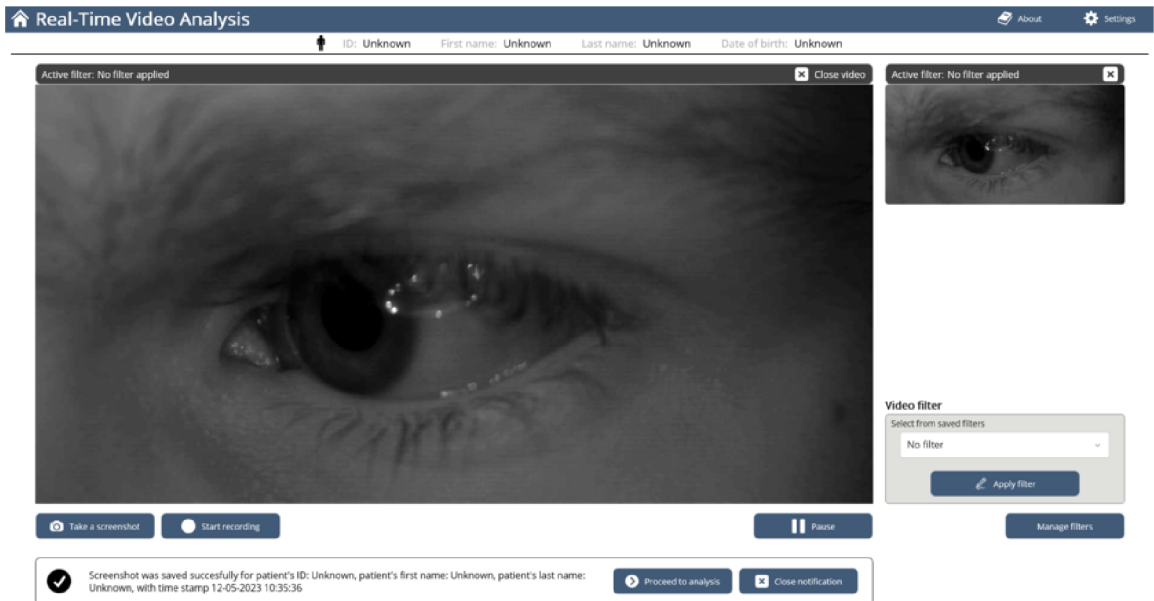


Obr. A.3: Obrazovka analýzy uloženej snímky

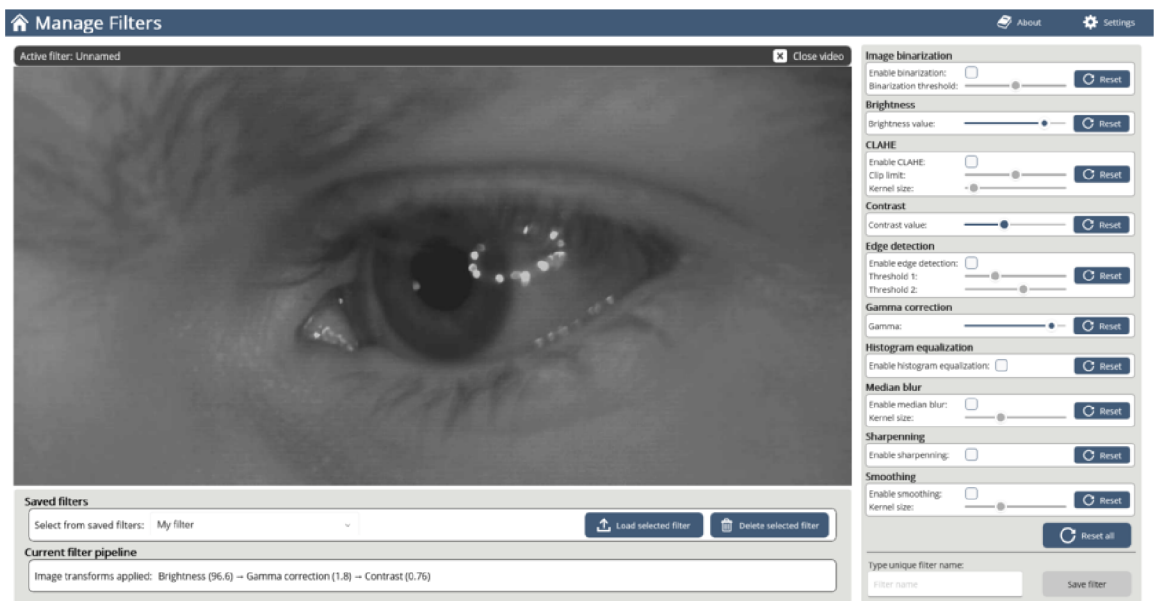


Obr. A.4: Obrazovka analýzy uloženého videa

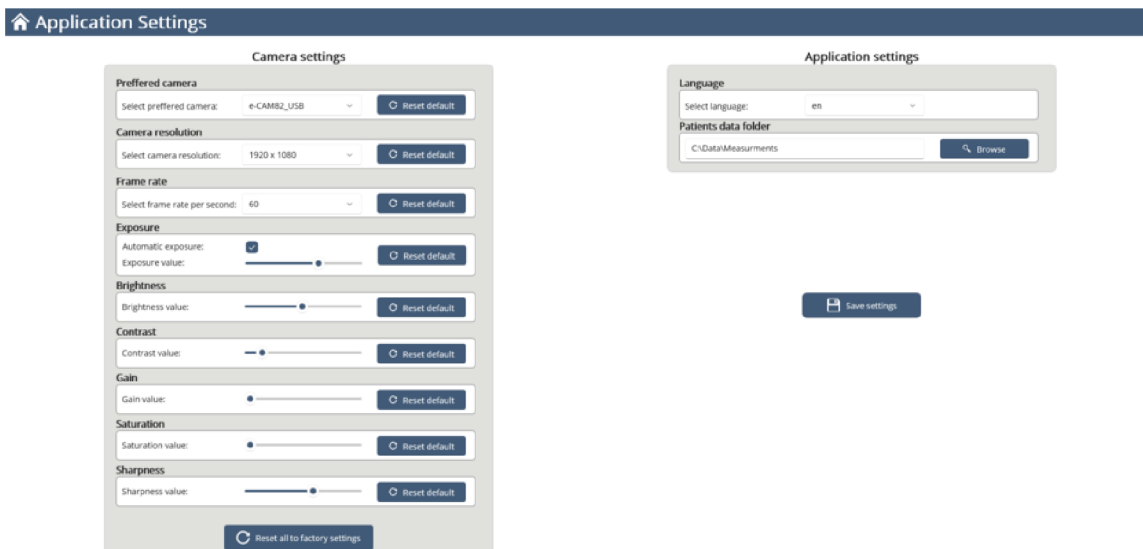
Obr. A.5: Obrazovka identifikácie pacienta



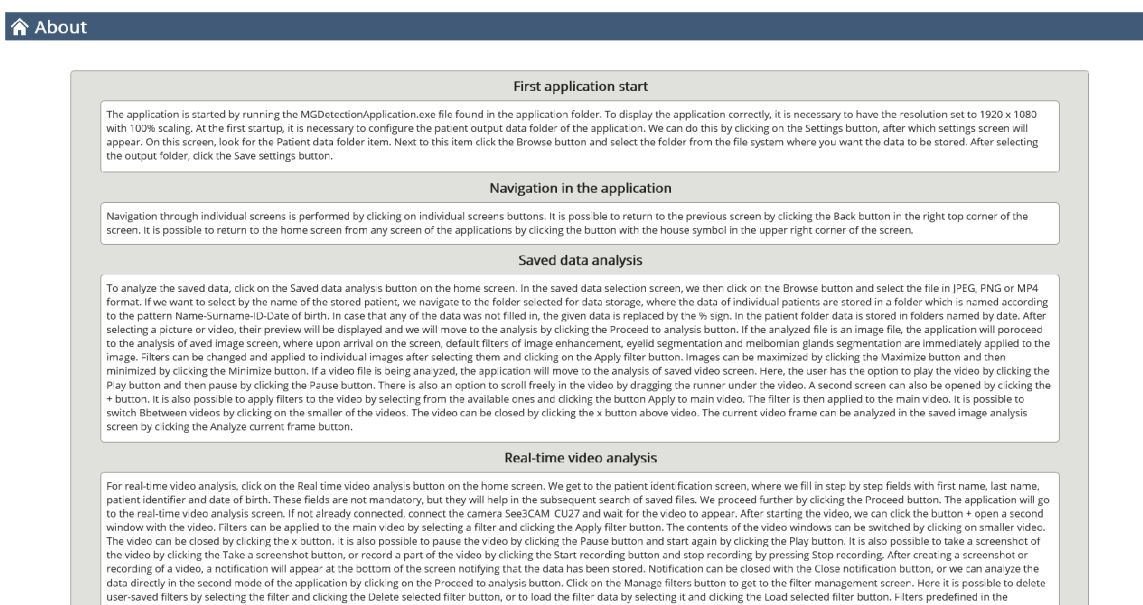
Obr. A.6: Obrazovka analýzy videa v reálnom čase



Obr. A.7: Obrazovka správy filtrov



Obr. A.8: Obrazovka nastavení



Obr. A.9: Obrazovka O aplikácii

A.2 Používateľská príručka aplikácie

Prvé spustenie aplikácie

Aplikácia sa spúšťa spustením súboru *MGDetectionApplication.exe* nachádzajúcom sa v priečinku aplikácie. Pre správne zobrazenie aplikácie je potrebné mať nastavené rozlíšenie obrazovky 1920 x 1080 s mierkou 100 %. Pri prvom spustení je nutné nakonfigurovať výstupný priečinok aplikácie. To vykonáme kliknutím na tlačidlo *Settings*, po čom sa zobrazí obrazovka nastavení. Na tejto obrazovke vyhľadáme položku *Patient data folder*. Pri nej klik-

neme na tlačidlo *Browse* a vyberieme priečinok zo súborového systému, kde si želáme dáta ukladať. Po zvolení výstupného priečinku kliknutím na tlačidlo *Save settings* nastavenia uložíme.

Navigácia v aplikácii

Navigácia jednotlivými obrazovkami je vykonávaná klikaním na jednotlivé tlačidlá obrazoviek. Na predchádzajúcu obrazovku je možný návrat po kliknutí na tlačidlo *Spät* v pravom hornom rohu obrazovky. Na domovskú obrazovku je možný návrat z každej obrazovky aplikácie kliknutím na tlačidlo so symbolom domčeka v pravom hornom rohu obrazovky.

Analýza uložených dát

Pre analýzu uložených dát klikneme na domovskej obrazovke na tlačidlo *Saved data analysis*. V obrazovke výberu z uložených dát následne klikneme na tlačidlo *Browse* a zvolíme súbor vo formáte JPEG, PNG alebo MP4.

V prípade, že chceme zvoliť podľa mena uloženého pacienta, odnavigujeme sa do priečinka zvoleného pre ukladanie dát, kde sú dáta jednotlivých pacientov ukladané do priečinkov pomenovaných podľa vzoru ID-Meno-Priezvisko-Dátum narodenia. V prípade, že nejaký z údajov nebol vyplnený, je daný údaj nahradený znakom %. V priečinku pacienta sú dáta ukladané do priečinkov podľa dátumu.

Po zvolení snímky, či videa sa zobrazí ich náhľad a k analýze sa presunieme kliknutím na tlačidlo *Proceed to analysis*.

V prípade, že analyzovaný súbor je snímka, aplikácia prejde na obrazovku analýzy uloženej snímky, kde sú pri príchode na obrazovku okamžite na snímku aplikované preddefinované filtre vylepšenia obrázku, segmentácie očného viečka a segmentácie meibomových žliaz. Filtre je možné meniť a aplikovať na jednotlivé obrázky po ich výbere a kliknutí na tlačidlo *Apply filter*. Snímky je možné maximalizovať kliknutím na tlačidlo *Maximize* a následne minimalizovať kliknutím na tlačidlo *Minimize*.

V prípade, že je analyzovaný súbor video sa aplikácia presunie do obrazovky analýzy uloženého videa. Tu má používateľ možnosť video spustiť kliknutím na tlačidlo *Play* a následne zastaviť kliknutím na tlačidlo *Pause*. Vo videu je tiež možnosť sa ľubovoľne posúvať ťahaním bežca pod videom. Kliknutím na tlačidlo + je možné otvoriť aj druhú obrazovku videa. Na video je tiež možné aplikovať filtre výberom z dostupných a kliknutím na tlačidlo *Apply to main video*. Filter je vtedy aplikovaný na hlavné video. Medzi videami je možné prepínať kliknutím na menšie z videí. Video je možné zatvoriť kliknutím na tlačidlo x nad videom. Aktuálnu snímku videa je možné analyzovať v obrazovke analýzy uloženej snímky kliknutím na tlačidlo *Analyze current frame*.

Analýza videa v reálnom čase

Pre analýzu videa v reálnom čase klikneme na domovskej obrazovke na tlačidlo *Real time video analysis*. Dostaneme sa na obrazovku identifikácie pacienta, kde postupne vyplníme polia s identifikátorom, menom, priezviskom a dátumom narodenia pacienta. Tieto polia nie sú povinné ale pomôžu pri následnom vyhľadávaní uložených súborov. Ďalej postupujeme kliknutím na tlačidlo *Proceed*.

Aplikácia prejde do obrazovky analýzy videa v reálnom čase. Ak ešte nie je pripojená kamera *See3CAM_CU27*, pripojíme ju a počkáme, kým sa zobrazí jej obraz. Po začatí premietania môžeme kliknutím na tlačidlo + otvoriť aj druhé okno s videom. Na hlavné

video je možné aplikovať filtre výberom filtra a kliknutím na tlačidlo *Apply filter*. Obsahy okien videí je možné prepínať kliknutím na menšie video. Video je možné zavrieť kliknutím na tlačidlo *x*. Video je tiež možné v tomto režime pozastaviť kliknutím na tlačidlo *Pause* a znovu spustiť kliknutím na tlačidlo *Play*. Je tiež možné vytvoriť snímku videa kliknutím na tlačidlo *Take a screenshot*, či nahráť časť videa kliknutím na tlačidlo *Start recording* a ukončiť nahrávanie stlačením *Stop recording*.

Po vytvorení snímky, či nahrávky videa sa v spodnej časti obrazovky objaví notifikácia o uložení dát, ktorú je možné zavrieť tlačidlom *Close notification*, alebo je možné uložené dáta priamo analyzovať v druhom režime aplikácie kliknutím na tlačidlo *Proceed to analysis*.

Kliknutím na tlačidlo *Manage filters* sa dostaneme na obrazovku správy filtrov. Tu je možné mazať používateľom uložené filtre výberom filtra a kliknutím na tlačidlo *Delete selected filter*, či načítať údaje filtra jeho výberom a kliknutím na tlačidlo *Load selected filter*. Filtre preddefinované v aplikácii nie je možné odstrániť. Je tiež možné vytvoriť vlastný filter aplikáciou transformácií obrazu v pravej časti obrazovky. Filter uložíme zvolením pomenovania filtra a kliknutím na tlačidlo *Save filter*.

Poznámka k aktuálnej funkcionalite aplikácie

Aplikácia obsahuje niektoré predpripravené elementy, ktorých implementácia nebola súčasťou zadania diplomovej práce a budú dopracovávané v budúcnosti. Rovnako nie sú doimplementované ani niektoré časti, ktorých presná špecifikácia nie je nateraz známa, nakoľko prototyp kamery neustále prechádza zmenami.

Ide o funkcionalitu obrazovky nastavení, pre ktorú je aktuálne implementovaná iba voľba príčinka pre ukladanie dát pacientov. V prípade nevyhnutnej potreby zmeny parametrov kamery je možné ich meniť priamo v konfiguračnom súbore aplikácie nachádzajúcom sa v aplikačnom priečinku na ceste */Config/AppSettings.json*.

Okrem toho zatiaľ nie je implementované ani vyhľadávanie pacientov a ich dát priamo z prostredia aplikácie, nakoľko je v budúcnosti plánovaná implementácia prostredníctvom MySQL databázy. Aktuálne má používateľ možnosť importovať súbory zo súborového systému počítača.

V obrazovke analýzy uloženej snímky nie je implementovaný výpočet plochy očného viečka a meibomových žliaz spolu s určením stupňa poškodenia, čo je ďalším plánovaným rozšírením aplikácie.

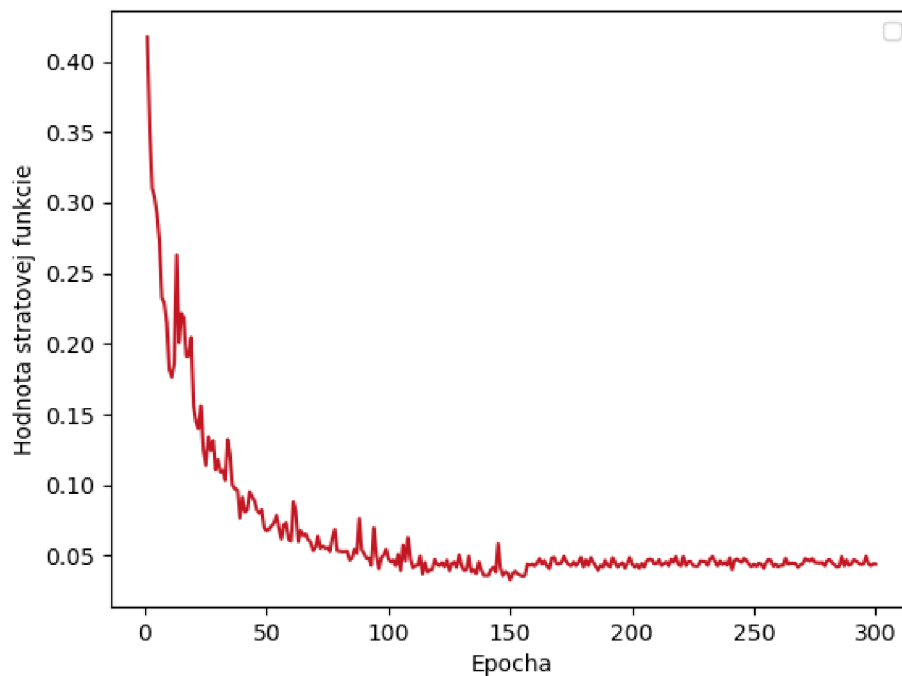
Rovnako bude v budúcnosti aplikácia pozmenená tak, aby bola kompatibilná s novým prototypom kamery pre snímanie slzných žliaz.

Príloha B

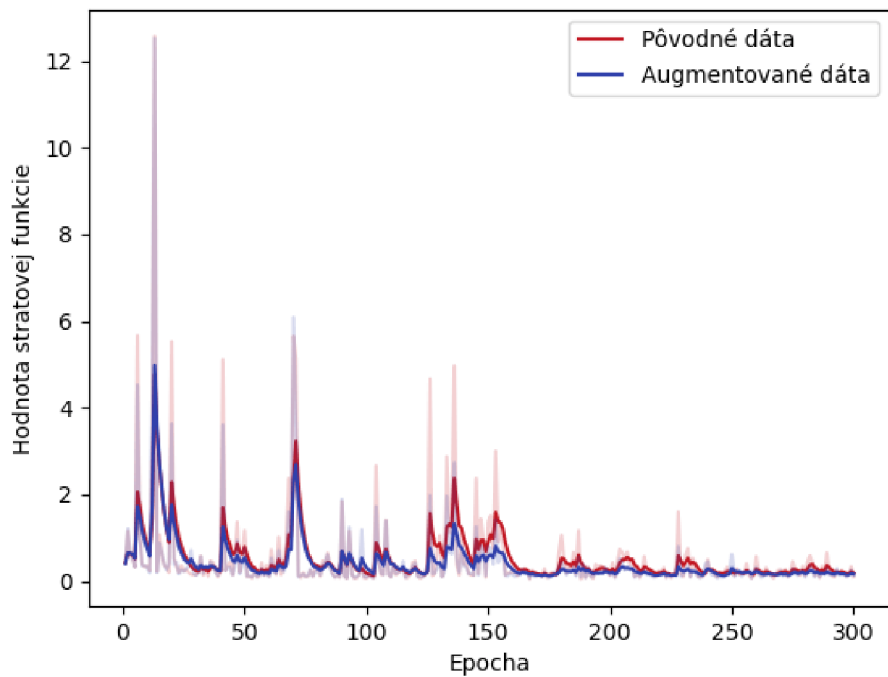
Podrobné výsledky trénovania neurónových sietí

V tejto prílohe sa nachádzajú trénovacie metriky najlepšieho behu každého typu segmentácie vykonanej v tejto práci. Zobrazený je vývoj stratovej funkcie na trénovacej dátovej sade, vývoj stratovej funkcie na validačnej dátovej sade pre pôvodné dáta validačnej sady a pre dáta, na ktoré sú aplikované fixné transformácie obrazu a rovnako je zobrazený vývoj hodnoty miery prekryvu pre pôvodné i transformované validačné dáta.

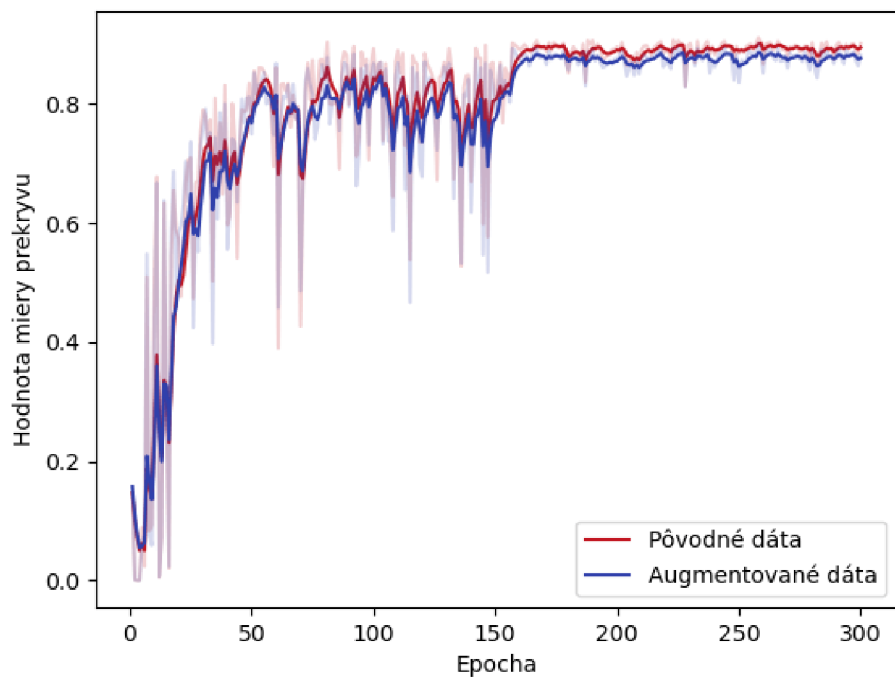
Vývoj trénovacích metrík siete U-Net pre segmentáciu očného viečka



Obr. B.1: Vývoj hodnôt stratovej funkcie na trénovacích dátach

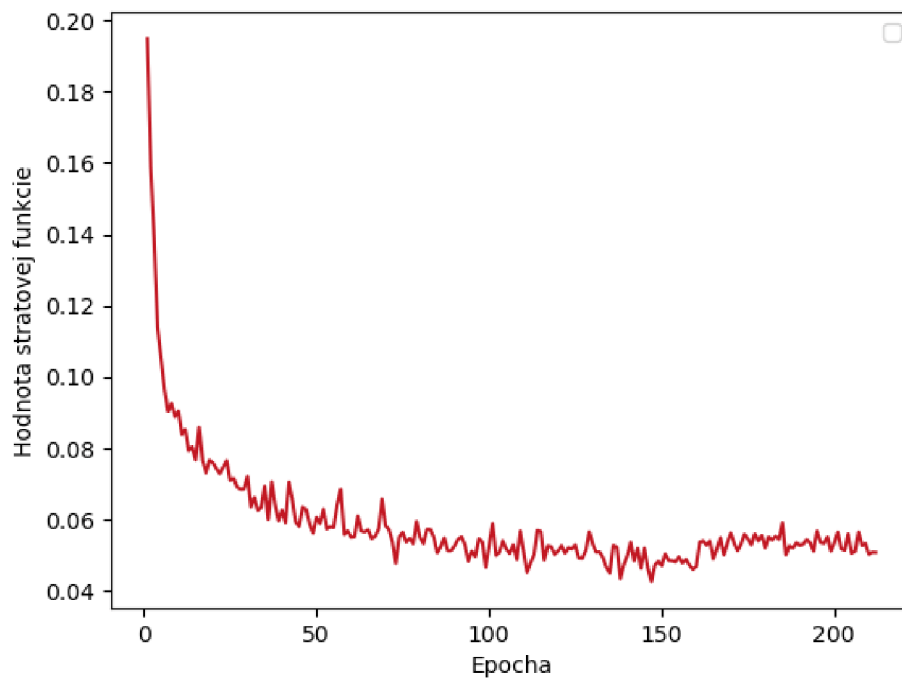


Obr. B.2: Vývoj hodnôt stratovej funkcie na validačných dátach

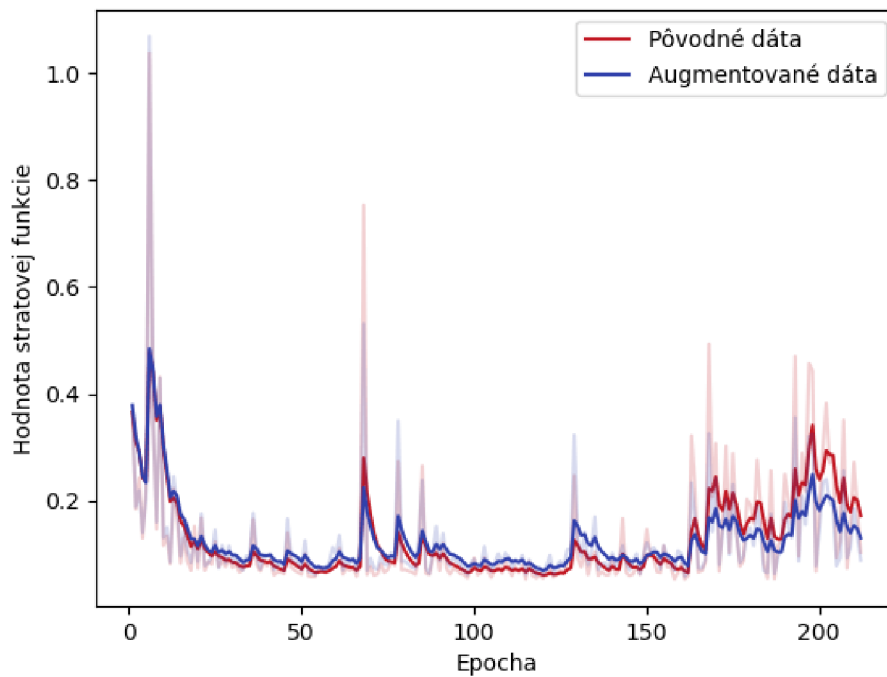


Obr. B.3: Vývoj hodnôt miery prekryvu na validačných dátach

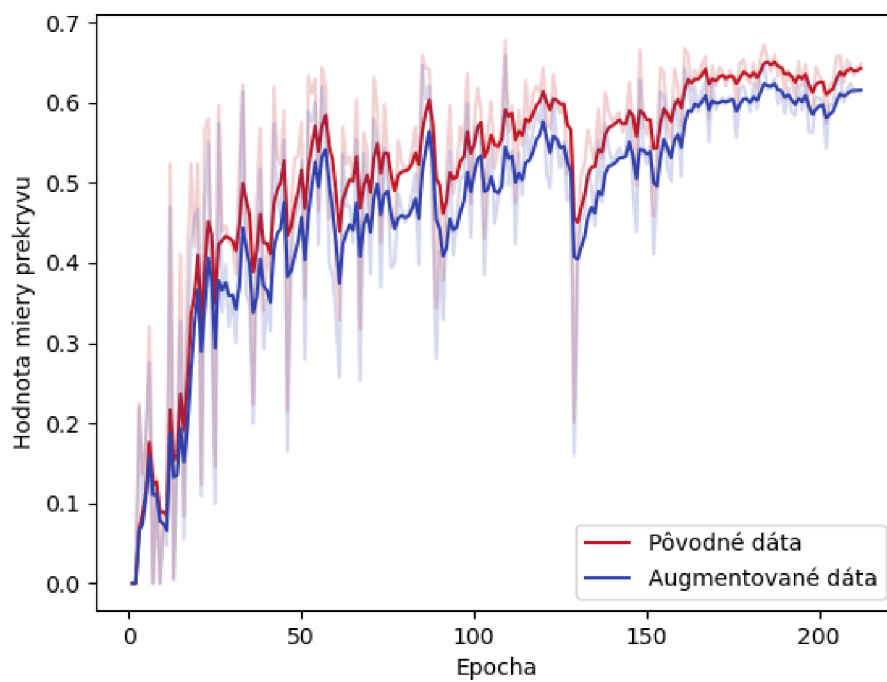
Vývoj tréningových metrick sítě U-Net pre segmentáciu meibomových žliaz



Obr. B.4: Vývoj hodnôt stratovej funkcie na tréningových dátach

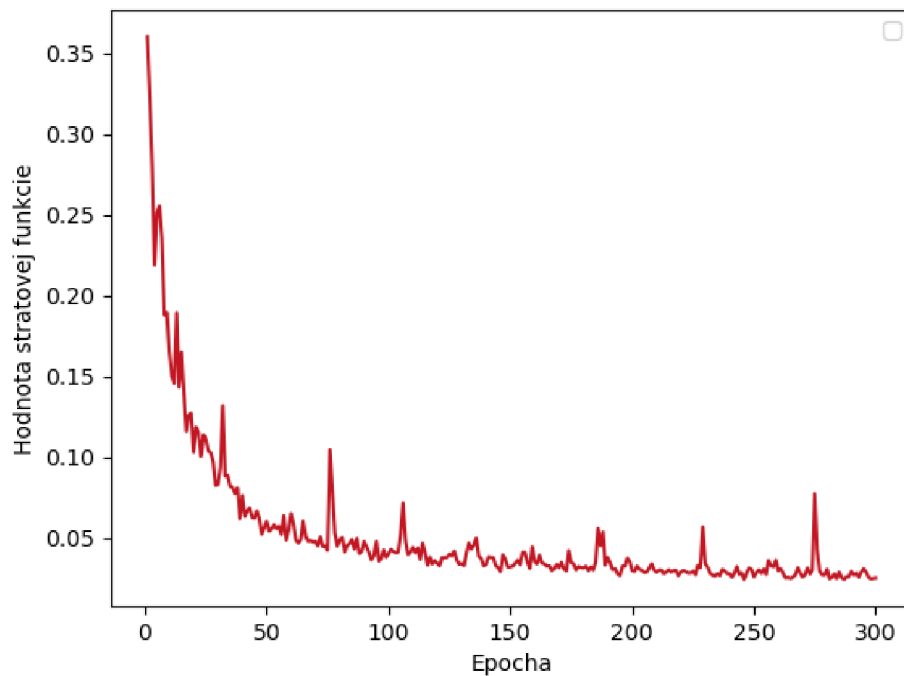


Obr. B.5: Vývoj hodnôt stratovej funkcie na validačných dátach

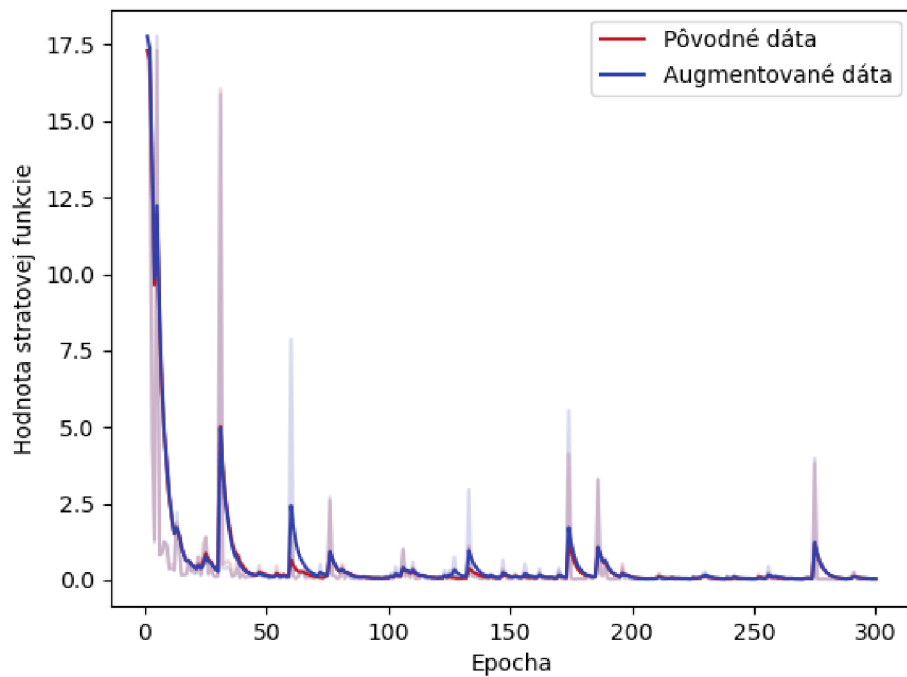


Obr. B.6: Vývoj hodnôt miery prekryvu na validačných dátach

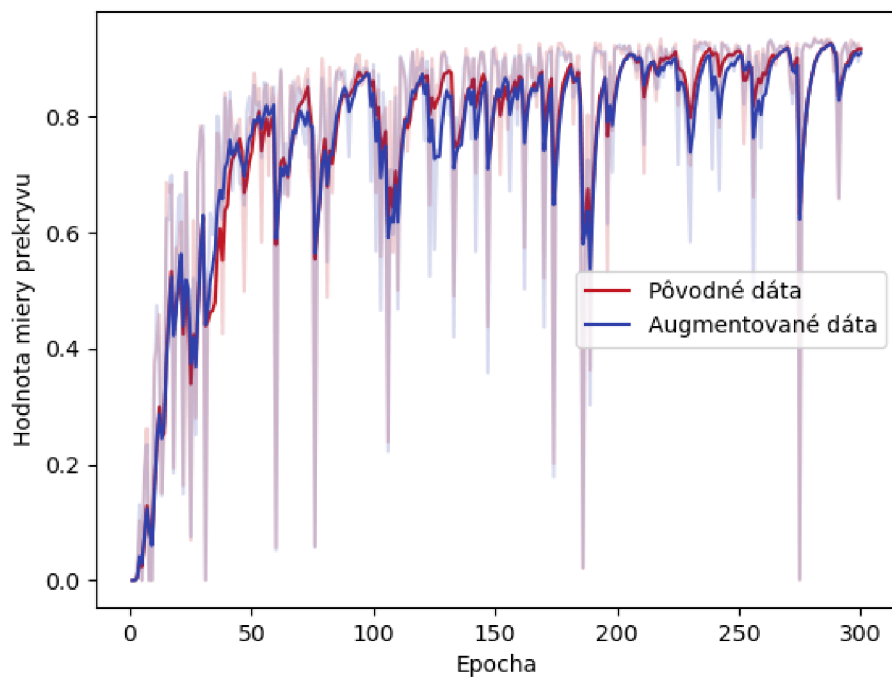
Vývoj tréovacích metrík siete MobileNetV3 pre segmentáciu očného viečka



Obr. B.7: Vývoj hodnôt stratovej funkcie na tréovacích dátach

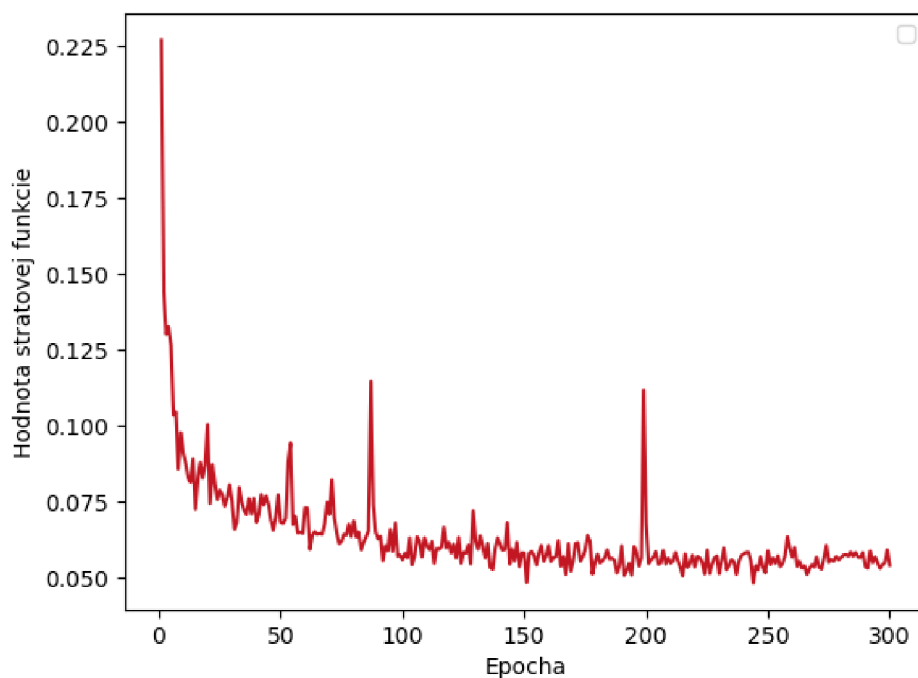


Obr. B.8: Vývoj hodnôt stratovej funkcie na validačných dátach

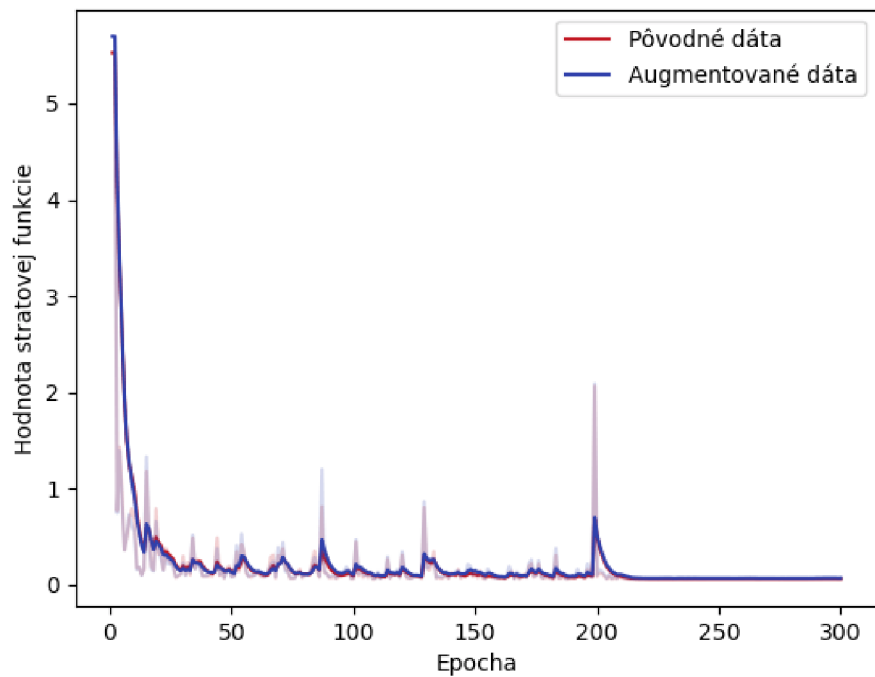


Obr. B.9: Vývoj hodnôt miery prekryvu na validačných dátach

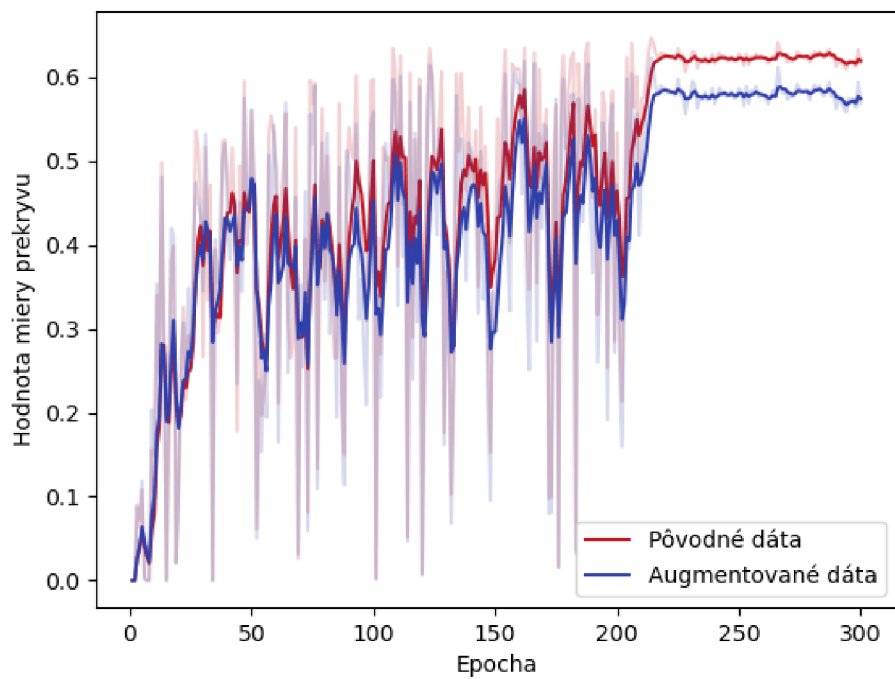
Vývoj tréovacích metrik siete MobileNetV3 pre segmentáciu meibomových žliaz



Obr. B.10: Vývoj hodnôt stratovej funkcie na tréovacích dátach



Obr. B.11: Vývoj hodnôt stratovej funkcie na validačných dátach



Obr. B.12: Vývoj hodnôt miery prekryvu na validačných dátach

Príloha C

Obsah pamäťového média

Priložené elektronické médium obsahuje dva súbory a tri priečinky:

- *DP_xsanda03_text.pdf* - Text diplomovej práce vo formáte PDF.
- *DP_xsanda03_user_manual.pdf* - Používateľská príručka aplikácie a návod na spustenie zdrojových kódov vo formáte PDF.
- *DP_xsanda03_text_src* - priečinkok obsahujúci zdrojové súbory pre text záverečnej práce pre Overleaf.
- *DP_xsanda03_code_src* - priečinkok obsahujúci zdrojové súbory kódu vypracovaného pre účely diplomovej práce a dátovú sadu pre účely tréningovania neurónových sietí.
- *DP_xsanda03_app* - priečinkok obsahujúci publikovanú aplikáciu s príponou .exe a súbormi potrebnými pre jej správny beh.

Priečinkok *DP_xsanda03_code_src* má nasledovnú štruktúru:

- */python* - priečinkok obsahujúci skripty vytvorené v jazyku Python 3.8 na predspracovanie dátovej sady, tréningovanie a testovanie neurónových sietí, vykresľovanie diagramov a pôvodnú i predspracovanú dátovú sadu
 - */dataset* - priečinkok obsahujúci dátovú sadu
 - * */raw_data* - priečinkok obsahujúci pôvodnú dátovú sadu MGD-1K
 - * */preprocessed_data* - priečinkok obsahujúci predspracované dátové sady pre segmentáciu očného viečka a pre segmentáciu meibomových žliaz
 - */test_results* - priečinkok obsahujúci výsledky testovania pre jednotlivé natréňované modely neurónových sietí
 - */train_results* - priečinkok obsahujúci priebehy tréningovania jednotlivých modelov neurónových sietí spolu s diagramami tréningovania
 - jednotlivé skripty vytvorené v jazyku Python 3.8 na predspracovanie dátovej sady, tréningovanie a testovanie neurónových sietí a vykresľovanie diagramov
- */dotnet* - priečinkok obsahujúci zdrojové súbory aplikácie s používateľským rozhraním

- */MGDetectionApplication* - priečinok obsahujúci celé riešenie (solution) aplikácie
 - * */Config* - priečinok obsahujúci konfiguračné súbory aplikácie a zdrojové kódy pre prácu s nimi
 - * */Filters* - priečinok obsahujúci preddefinované filtre aplikácie
 - * */ImageTransformations* - priečinok obsahujúci zdrojové kódy pre jednotlivé transformácie obrazu dostupné v aplikácii
 - * */Models* - priečinok obsahujúci zdrojové kódy modelov v aplikácii
 - * */NeuralNetworks* - priečinok obsahujúci natrénované modely neurónových sietí vo formáte ONNX
 - * */Resources* - priečinok obsahujúci zdroje pre aplikáciu ako obrázky, ikonky, texty, štýly či fonty
 - * */ViewModels* - priečinok obsahujúci zdrojové kódy view-modelov v aplikácii
 - * */Views* - priečinok obsahujúci zdrojové kódy view (obrazoviek) v aplikácii
 - * ostatné súbory a priečinky, ktoré sú súčasťou riešenia
- */MGDetection.sln* - súbor riešenia aplikácie pre spustenie v programe Visual Studio 2022

Priečinok *DP_xsanda03_app* má nasledovnú štruktúru:

- */MGDetectionApplication.exe* - spustiteľný súbor aplikácie s používateľským rozhraním
- ostatné súbory a priečinky potrebné pre beh aplikácie