



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**SEGMENTACE ŽIL A TEPEN V SÍTNICOVÝCH
SNÍMCÍCH**

VEIN-ARTERY SEGMENTATION OF BLOOD VESSELS IN RETINAL IMAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RADEK SEDLÁŘ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ANDRII KAVETSKYI

BRNO 2021

Abstrakt

Tato práce se zaměřuje na úvod do problematiky segmentace žil a tepen ze sítnicových snímků. Práce obsahuje srovnání nejpoužívanějších metod s jejich klady a zápory. Dále je navržena vlastní metoda pro segmentaci a rozdělení na žíly a tepny. Práce také obsahuje detailní popis implementace navržené metody a shrnutí jejich výsledků.

Abstract

This work focuses on an introduction to the issue of segmentation of veins and arteries from retinal images. The work contains a comparison of the most used methods with their pros and cons. Furthermore, a proprietary method for segmentation and division into veins and arteries is proposed. The work also contains a detailed description of the implementation of the proposed method and a summary of their results.

Klíčová slova

Segmentace, OpenCV, Python, DRIVE, STARE, HRF, Přizpůsobené filtrování

Keywords

Segmentation, OpenCV, Python, DRIVE, STARE, HRF, Matched filtering

Citace

SEDLÁŘ, Radek. *Segmentace žil a tepen v sítnicových snímcích*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Andrii Kavetskyi

Segmentace žil a tepen v sítnicových snímcích

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Andrii Kavetskeho. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Radek Sedlář
9. května 2022

Poděkování

Děkuji panu Andrii Kavetskemu za pomoc při vedení bakalářské práce. Mé poděkování patří též mé rodině a kamarádům, kteří mi byli po dobu studia oporou.

Obsah

1	Oko	6
1.1	Anatomie oka	6
1.2	Anatomie sítnice	7
1.2.1	Onemocnění sítnice	8
1.3	Snímky sítnice	8
2	Techniky segmentace	10
2.1	Techniky segmentace krevního řečiště	10
2.2	Techniky rozlišení žil a tepen	11
2.3	Vybrané nejvýkonnější přístupy	12
3	Navrhovaná metoda	13
3.1	Předzpracování	13
3.2	Segmentace cévního řečiště	14
3.3	Nalezení žil a tepen	15
3.3.1	Skeletonizace	15
3.3.2	Nalezení optického disku	16
3.3.3	Výběr počátků cév	16
3.3.4	Sledování	16
3.3.5	Rozlišení na žíly a tepny	16
4	Implementace	18
4.1	Předzpracování	18
4.2	Segmentace cévního řečiště	19
4.3	Skeletonizace	22
4.4	Nalezení optického disku	23
4.5	Výběr počátku cév	23
4.6	Sledování	24
4.7	Rozlišení na žíly a tepny	26
4.7.1	První přístup	26
4.7.2	Druhý přístup	28
4.8	Dokončení	29
5	Zhodnocení výsledků	30
5.1	Hodnocení metod	30
5.1.1	Měřená kritéria	30
5.2	Nejvýkonnější metody	31
5.3	Výsledky navrhované metody pro segmentaci	31

5.3.1	DRIVE	31
5.3.2	HRF	32
5.3.3	STARE	34
5.4	Výsledky lokalizace optického disku	36
5.5	Výsledky sledování	37
5.6	Výsledky rozdělení na žíly a cévy	38
6	Závěr	42

Zadání bakalářské práce



24912

Student: **Sedlář Radek**
Program: Informační technologie
Název: **Segmentace žil a tepen v sítnicových snímcích**
Vein-Artery Segmentation of Blood Vessels in Retinal Images
Kategorie: Bioinformatika

Zadání:

1. Nastudujte literaturu týkající se sítnice lidského oka, techniky segmentace sítnicových krevních cév a rozdělení krevních cév na žíly a tepny.
2. Navrhněte vlastní algoritmus pro segmentaci krevních cév s rozdělením krevních cév na žíly a tepny.
3. Implementujte algoritmus z předchozího bodu, otestujte jeho výkonnost a přesnost.
4. Zhodnoťte dosažené výsledky a navrhněte možnosti dalšího rozšíření práce.

Literatura:

- R. Hemelings, B. Elen, I. Stalmans, K. Van Keer, P. De Boever, and M. B. Blaschko. Artery-Vein Segmentation in Fundus Images Using a Fully Convolutional Network. *Computerized Medical Imaging and Graphics*, 2019.
- A.P. Schachat. *Ryan's Retina*. Sixth edition. Edinburgh; New York: Elsevier, 2018.
- T. A. Soomro, A. J. Afifi, L. Zheng, S. Soomro, J. Gao, O. Hellwich, and M. Paul. Deep Learning Models for Retinal Blood Vessels Segmentation: A Review. *IEEE Access*, 2019.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 a 2.

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

Vedoucí práce: **Kavetskyi Andrii**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 3. listopadu 2021

Seznam obrázků

1.1	Lidské oko [16].	7
3.1	První snímek rozložený na RGB složky a jeho příslušná maska.	14
4.1	Příklad výstupů procesu předzpracování	19
4.2	Příklad získání hodnot pro profil z hodnot normálního rozdělení s parametry $\mu = 7, \sigma = 0.9$	20
4.3	Příklad masky otočené o 0° , 15° a 90°	21
4.4	Výstup filtrovací funkce	22
4.5	Výsledky přístupu skeletonizace.	23
4.6	Práce s optickým diskem a hledání počátků cév.	24
4.7	Okolí použité pro sledování cév. Zelené pixely reprezentují okolí a modrý je středový pixel.	25
4.8	Ukázka rozdělení cévní struktury na menší segmenty.	27
4.9	Výběr pixelů pro výpočet ODR. Černé pixely jsou pozadí a bílé céva. Střed cévy pak představuje modrý pixel a okraje jsou vyznačeny zeleně. Červené jsou vybrané pixely pro reprezentaci pozadí.	28
4.10	Ukázka rozdělení cévní struktury na menší segmenty.	29
5.1	Tvorba masek pro STARE.	34
5.2	Porovnání výsledku a manuálně získaného výsledku pro snímek číslo 11 z množiny STARE.	36
5.3	Příklad lokalizace optického disku a vyznačených startovacích bodů pro sledování.	37
5.4	Srovnání výsledků obou metod a očekávaného výsledku.	41

Úvod

Vyšetření očního pozadí je velice důležité pro velké množství zdravotnických odvětví. Oční lékař je z něj schopen zjistit mnoho informací o zdravotním stavu pacienta a sledovat tak jeho stav. Vyšetření je relativně jednoduché, neinvazivní a výsledky jsou k dispozici téměř okamžitě. Vyšetření se provádí speciální kamerou, která pořídí snímek očního pozadí pacienta, který se následně uloží do připojeného počítače. Lékař pak může se snímkem v počítači libovolně manipulovat a zkoumat ho bez další potřeby pacienta.

Jakožto diabetik často chodím na toto vyšetření, a proto mi záleží na technikách a kvalitě vyhodnocování těchto snímků, protože s jejich pomocí lze například určit míru poškození cév cukrovkou.

V minulosti se snímky nepořizovaly a lékaři proto museli používat speciální zrcátko, což bylo pro pacienta značně nepříjemné. V dnešní době se kamerou pořízené snímky mohou vyhodnocovat výpočetní techniky a díky tomu je možné zjistit skutečnosti, které nelze na první pohled poznat, například poměr žil a tepen na obrázku. Především v posledních letech se v tomto odvětví značně projevil trend v podobě umělé inteligence.

Cílem této práce je nastudovat běžně používané techniky pro toto téma, navrhnout vlastní postup segmentace cévního řečiště a jeho rozdělení na tepny a žíly, tento postup následně implementovat a poté zhodnotit výsledky.

V první kapitole je krátké shrnutí anatomie oka, sítnice a nemocí, které se ze sítnicových snímků dají určit. Na konci této kapitoly se nachází výčet databází veřejně přístupných dat, které se budou v této práci používat.

V kapitole **Techniky segmentace** se zaměřuji na nejznámější techniky často používané k těmto úlohám a porovnávám jejich výhody a nevýhody. V první části rozebírám segmentace cévního řečiště a v části druhé jejich rozdělení na tepny a žíly.

Kapitola **Navrhovaná metoda** vysvětluje principy fungování navrženého postupu.

Implementace se věnuje procesu implementace navrhované metody v jazyce Python, za pomoci knihovny OpenCV.

Kapitola **Zhodnocení výsledků** pak shrnuje všechny výsledky dosažené použitím navrhované metody a jejich srovnání s ostatními metodami.

Kapitola 1

Oko

1.1 Anatomie oka

Oko se skládá z oční koule, zrakového nervu a z pomocných ústrojí oka. Část s veškerou funkcí důležitou pro zrak je oční koule, která má tvar nepravidelné koule. Z vnějšku je patrná pouze přední část oční koule - rohovka a přilehlé části bělimy. Větší část oční koule je uložena v očnici, což je dutina v lebce. Na oční kouli lze rozlišovat dva póly, a to pól přední, který se nachází v nejméně vypouklé části rohovky, a pól zadní, nacházející se v centru zadní části, kousek od vyústění optického nervu. Rozměry oční koule se značně liší u každého jedince, ale jako největší vzdálenost je považována právě vzdálenost předního a zadního pólu koule oční, jedná se v průměru o 24 mm. Povrch oční koule je tvořen ze 4/5 bělimou a z 1/5 rohovkou. Celá anatomie je naznačena na obrázku 1.1.

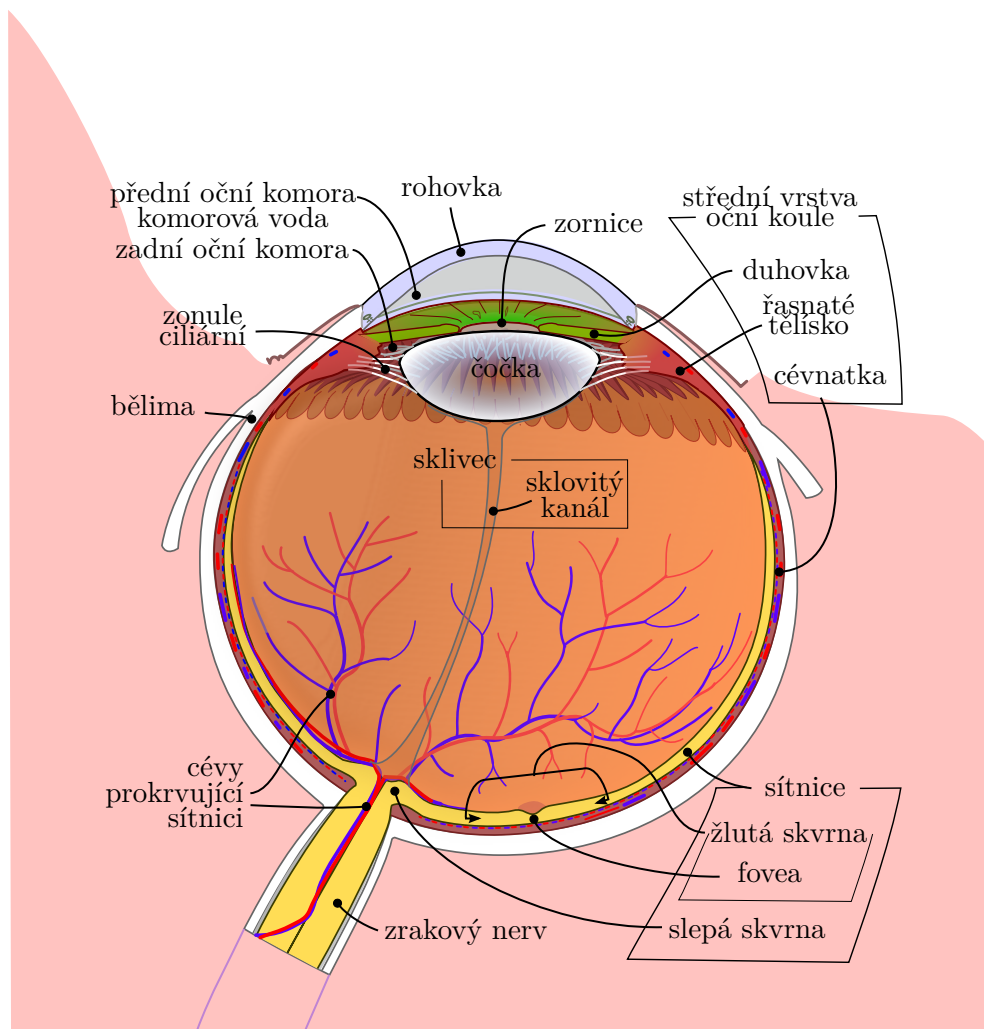
Bělina je vazivová tkáň nacházející se na zadní části oční koule. Díky šlachovitému charakteru této tkáně a nízkému prokrvení má mléčnou barvu, která pak ve stáří může přecházet v nažloutlou, kvůli zrnkům tuku, která se zde ukládají.

Rohovka se pak nachází v přední části oční koule. Je bezbarvá, zcela bez cév a průhledná. Přechod bělimy v rohovku je postupný.

Cévnatka je největší část střední vrstvy oka, protíná zadní část střední vrstvy v rozsahu dvou třetin celkového povrchu. Její tloušťka se pochybuje okolo 0,4 mm v zadní části a postupně se ztenčuje až na 0,2 mm. Cévnatka je velmi dobře prokrvená a díky tomu zajišťuje výživu většině vnitřních částí oka.

Duhovka se nachází v přední části a to v prostoru mezi rohovkou a čočkou. Má tvar mezikruží s vykrojeným otvorem - zorničkou. Přední část duhovky je viditelná přes rohovku a má charakteristickou jak barvu, tak kresbu. Tyto charakteristiky se pak mohou používat k identifikaci jedince, podobně jako otisky prstů. Duhovka je na svých okrajích spojena se svaly, které dokážou měnit její tvar a přizpůsobit ji světelným podmínkám. Typicky se při nedostatku světla rozšíří, aby umožnila většímu množství světla dopadnout na sítnici.

Čočka je orgán nacházející se za duhovkou, tvarově připomínající disk. U dospělého a zdravého jedince je čočka průhledná a jasná. Je zodpovědná za správný lom dopadajícího světla na sítnici. To jí umožňují svaly, kterými je zachycena po stranách. Tyto svaly pak



Obrázek 1.1: Lidské oko [16].

mění její tvar a umožňují jedinci zaostřit na různé vzdálenosti.

Sklivec tvoří 2/3 oční koule a nachází se mezi čočkou a sítnicí. Skládá se převážně z vody, a to až z 98 %.

1.2 Anatomie sítnice

Vnitřní vrstva oka, neboli sítnice, vystýlá vnitřní prostor oka až k okraji duhovky. Tkáň sítnice je pokryta světločivnými buňkami. Barva sítnice je podmíněna prokrvením a množstvím pigmentu daného jedince, ale většinou bývá sytě červeně oranžová. Nejvýraznější částí je takzvaný optický disk, který je vyústěním optického nervu a všech viditelných cév na očním pozadí. Optický disk je jediné místo, které není pokryté světločivnými buňkami, a proto je tento bod slepý. Asi 4 mm od optického disku leží žlutá skvrna. Je to místo s největší koncentrací světločivných buněk, a tím pádem je to také místo nejostřejšího vidění. Žlutá skvrna je tmavší než ostatní části. Svě jméno dostala, protože u mrtvol zežloutne.

Světločivné buňky sítnice se dělí na dvě hlavní skupiny - tyčinky a čípky. Tyčinky dokážou vnímat intenzitu světla a čípky pak jeho vlnovou délku (barvu).[14]

1.2.1 Onemocnění sítnice

Právě ze snímků sítnice lze diagnostikovat některá onemocnění nebo zjistit jejich závažnost. Jelikož je sítnice prokrvena vlásečnicemi, tak je náchylná na různé záněty a změny tlaku.

Zelený zákal (dále jen anglicky *Glaucoma*) je onemocnění způsobené zvýšeným tlakem v oční kouli. Zvýšení tlaku má největší následek na optickém disku a cévách, které z něj vystupují. Tyto cévy mají tedy snížený průtok a nedokážou dostatečně prokrvovat sítnici. Na snímcích lze nemoc poznat z rozšířeného středu optického disku a podle zvýraznění jeho okraje. V posledních stádiích lze pozorovat onemocnění i zvenčí pouhým okem, a to jako změnu barvy oka z průhledné na mléčně bílou. Glaucoma se dělí na více typů dle příčiny vzniku.

Retinopatie je souhrnné označení pro patologické změny cév. Nejčastěji jsou tyto změny spojeny s diabetem. Na snímcích se retinopatie projevuje více způsoby. Nejvýraznější jsou nepravidelné šířky cév, podlitiny a místa s uloženým tukem. Neléčení příčin retinopatie vede k neodvratnému poškození a následné ztrátě zraku.[24]

1.3 Snímky sítnice

Pro tuto práci byly využity databáze DRIVE[6], HRF[15] a STARE[28]. Tyto databáze byly založeny za účelem srovnávání automatických metod pro segmentaci cévního řečiště, detekci nemocí a měření vlastností cév, jako například délky cév nebo poměru žil a tepen.

DRIVE - Snímky pochází z programu, který sleduje vývoj diabetické retinopatie u 400 pacientů ve věkovém rozmezí 25 až 90 let. Ze všech pořízených snímků bylo vybráno 40, které se skládají z 33 snímků bez náznaku nemoci a 7 snímků s nemocí v raném stádiu. Pro pořízení snímků bylo použito zařízení *Canon CR5 non-mydratic 3CCD camera*, což je speciální přístroj pro pořizování snímků očního pozadí se zorným polem (dále jen anglickou zkratkou *FOV*) 45°. Snímky jsou pak v rozlišení 768x584 pixelů s barevnou hloubkou 256 barev. Následně jsou snímky oříznuté kolem *FOV* a ke každému je přiřazena maska, která reprezentuje hranice *FOV*. Snímky jsou rozděleny na dvě skupiny, trénovací a testovací, obě s 20 snímky. Trénovací množina má navíc ke každému snímku přiřazený výsledný snímek nazývaný jako *zlatý standard*.

HRF - Snímky obsažené v HRF obsahují 45 snímků. Tyto snímky jsou dále rozděleny na tři skupiny po 15 snímcích. První skupina obsahuje snímky zdravých pacientů, druhá snímky s rozvinutou diabetickou retinopatií a třetí pak snímky s viditelným zeleným zákalem. Všechny snímky pak byly pořízeny pomocí přístroje *Canon CR-1 fundus camera*, který disponuje *FOV* 45°. Všechny snímky mají vysoké rozlišení 3504x2336 pixelů. Ke každému snímku existuje maska a *zlatý standard*.

STARE - Celkem obsahuje 400 snímků s různými deformacemi, ale také bez nich. Ze všech snímků je vybrána množina 20, u kterých specialisté ručně segmentovali cévní řečiště. Ne všechny snímky obsahují optický disk. Rozlišení všech snímků je 700x605

pixelů. Tato množina neobsahuje vytvořené masky pro maskování okolí instrumentu, proto je potřeba je vytvořit.

Kapitola 2

Techniky segmentace

2.1 Techniky segmentace krevního řečiště

Technik segmentace krevního řečiště z sítnicových snímků je celá řada. Shrnutí je inspirováno [8]. Většina přístupů kombinuje dvě a více metod pro dosažení nejlepšího výsledku. Každá metoda má své výhody a nevýhody, je tedy důležité zvolit správnou kombinaci více metod. Všechny přístupy mají za úkol rozlišit mezi pozadím a cévou.

Učení s učitelem (anglicky *Supervised learning*) takto jsou označovány algoritmy, které používají strojového učení a předem připravených tréninkových dat. Data jsou vytvořena zkušeným oftalmologem, který ručně vyznačí krevní řečiště z obrázku a toto vyznačení se pak označuje jako zlatý standard. Algoritmy pak porovnávají původní snímky s příslušným zlatým standardem a učí se závislosti mezi nimi. Po učení se algoritmus spustí na další množině stejně připravených snímků a sleduje se jeho přesnost a další vlastnosti. Mezi práce využívající tyto postupy můžeme řadit například: síť zpětného šíření (anglicky *back propagation network*)[18], neuronové sítě (anglicky *neural networks*)[26]

Učení bez učitele (anglicky *Unsupervised learning*) se snaží nalézt vlastnosti nebo jejich kombinaci, díky které lze jednoznačně určit, zda pixel náleží cévě, nebo ne. Na rozdíl od metod pod dohledem nepoužívají předem segmentovaná data k učení. Jako metody učení bez učitele mohou být označovány například: RACAL (RADIUS based CLustering ALgorithm)[25], FCM (fuzzy C-means clustering)[31]

Přizpůsobené filtrování (anglicky *Matched filtering*) je metoda využívající speciálně vytvořeného filtru (anglicky *kernel*), který je pak aplikován na každý pixel daného obrázku. Sestavení správného filtru je klíčové a k jeho zhotovení se používá podobnost s některými funkcemi, nejčastěji funkcí normálního rozdělení. Práce využívající Matched Filtering[4][20].

Morfologické zpracování obrazu (anglicky *Morphological processing*) jsou techniky využívající takzvané morfologické operátory. Základní jsou *dilatace*, *eroze* a od nich odvozené *otevření* a *uzavření*. Morfologické zpracování se používá především na binární obrazy (jen černé a bílé), ale může být použito také na gray-scale (obraz obsahující pouze stupně šedi)[10].

Sledování cév (anglicky *Vessel tracking*) spočívá v určení relativně malé množiny bodů, nejčastěji kořenových bodů cév, od kterých se dále hledá další pokračování cévy. Vy-

hledávání dalšího bodu většinou spočívá v sledování určité vlastnosti nebo skupiny vlastností z okolí aktuálního pixelu. Tyto postupy bývají značně výpočetně náročné. Největší problém této metody je určení výchozích bodů pro začátek sledování. Sledování je použito například v [5].

Vícestupňové přístupy (anglicky *Multiscale approaches*) používají profil založený na normálním rozdělení podobně jako "Přizpůsobené filtrování". Hlavní změna je v tom, že je brána v potaz vzdálenost od optického disku a profil je postupně upravován, aby simuloval postupně se zužující cévy.[7]

Parametrické modely (anglicky *Parametric models*) jsou modely založené na aktivních obrysových modelech představených v [13]. Tyto přístupy využívají množinu bodů, která tvoří obrys. Tento obrys pak hledá další body v okolí pro svou expanzi. To zda se model rozšíří, je určováno jeho vnitřní energií, která je vypočítána z vlastností jeho bodů. Tyto přístupy jsou často nazývány jako hadi (anglicky *snakes*). Nevýhoda je nutnost nalezení výchozích bodů. Tento přístup je například použit v práci [22].

Geometrické modely (anglicky *Geometric models*) přístup založen na teorii geometrického vývoje křivky. Využívá LSM (anglicky *Level-set method*), která se používá pro sledování křivek a povrchů. Výhodou LSM je, že lze provádět numerické výpočty zahrnující křivky a povrchy na pevné kartézské mřížce, aniž by bylo nutné tyto objekty parametrizovat. V článku [29] byl navržen přístup rozdělení původního snímku na části, podle úrovně osvětlení.

2.2 Techniky rozlišení žil a tepen

Rozlišení segmentovaných cév na tepny a žíly se dá provádět na základě několika vlastností.

Tloušťka cév - žíly jsou oproti tepnám širší, ve stejné vzdálenosti od optického disku.

Centrální reflex - tato vlastnost ukazuje poměr intenzity středového pixelu cévy a intenzity pixelu reprezentující stěnu cévy.

Kontrast s pozadím - vlastnost určuje, jaký je rozdíl mezi cévou a jejím přilehlým pozadím.

Množství - poměr žil a tepen na snímku úzce souvisí s tloušťkou. Tím, že všechny cévy vystupují z optického disku a postupně se s zvyšující se vzdáleností dělí a zužují, tak cévy končí dříve než žíly. Díky těmto vlastnostem je pak jasné, že v poměru je tepen méně. Tento údaj se pak vypočítává z výsledku.

Tyto poznatky se pak dají uplatnit hned v několika přístupech. Originální snímek se rozloží podle příslušného barevného modelu na jeho základní složky a poté jsou sledovány rozdíly mezi nimi. Většina těchto přístupů využívá rozdíly v různých vlnových délkách barevného modelu RGB a to konkrétně ve složkách představujících vlnové délky 700 nm (červená složka) a 546,1 nm (zelená složka).

Například v článku [2] se vypočítává vlastnost ODR (anglicky *optical density ratio*), která se vypočítá ze zelené a červené složky, které se extrahují z původního RGB snímku. Červená složka je citlivá na obsah kyslíku v cévách, proto jsou v tomto spektru žíly výraznější a tepny lépe splývají s pozadím. RGB však není jediný barevný prostor, který se

používá pro rozlišování. Například v článku [12] se používá složka Hue (česky *odstín*) z barevného modelu HSV.

S vypočítanými vlastnostmi je pak potřeba určit, jestli se daná céva řadí mezi žíly, nebo tepny. Pro finální rozhodování se následně používá například prahování, ale pokud je vlastností více, tak se používá algoritmus K-means. V poslední době jsou nejpůvodnější přístupy založené na plně konvolučních sítích [11] a hlubokém učení [30]. Tyto přístupy, ačkoliv jsou úspěšné, potřebují pro optimální fungování velké množství dat pro proces učení.

Další přístupy například vyberou množinu startovacích míst, které se vyhodnotí jako žíla či céva. Od těchto startovacích míst se pak odvíjí klasifikace zbylých cév. U těchto přístupů je problém určit startovací množinu bez předem vysegmentované mapy cév příslušného snímku.

Některé přístupy využívají rozdělení segmentovaného cévního řečiště pro sestavení grafu. Každá samostatně vyhodnocená část je pak zanesena do grafu. Celý graf je pak znova přepočítán a segmenty jsou poté znovu vyhodnoceny s ohledem na umístění v grafu. Tento přístup byl využit ve spojení s neuronovými sítěmi v článku [9].

2.3 Vybrané nejvýkonnější přístupy

V této sekci jsou stručně popsány vybrané přístupy, které budou v sekci s hodnocením použity jako vzorové řešení této problematiky.

- Wang et al. 2017 [33] Jedná se o přístup využívající učení s učitelem. Konkrétně používá dva klasifikátory: konvoluční neuronovou síť (CNN) a náhodný les (RF). Tato metoda je velice efektivní a je téměř vždy používána k srovnání výkonnosti nových algoritmů.
- Moghimirad et al. 2012 [17] Tento článek navrhuje metodu víceúrovňové segmentace, založenou na dvourozměrné funkci mediality. Ta je pak navíc vynásobena sestavenou Hessovskou maticí. Pak jsou aplikovány metody na redukci šumu, po níž zůstanou pouze segmenty, které je potřeba opětovně připojit.
- Odstrcilik et al. 2013 [20] V tomto článku se používá k segmentaci cévního řečiště sada přizpůsobených filtrů. Filtry jsou sestaveny na základě Gaussovy křivky. Oproti ostatním přístupům využívajícím podobný postup je tento přizpůsoben na snímky s větším rozlišením. Sada používaných filtrů je rozdělena na tři, podle tloušťky cév, které mají za úkol detekovat. Výsledné mapy cév pro každou tloušťku jsou spojené do jednoho výsledku.

Pro srovnání rozdělení žil a tepen nebyl nalezen žádný důvěryhodný zdroj, tím pádem je složité uvést nejvýkonnější a nejpřesnější metody pro klasifikaci cév.

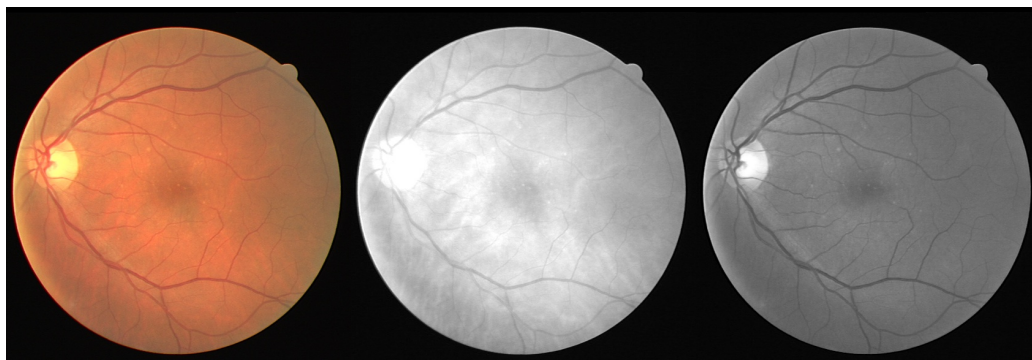
Kapitola 3

Navrhovaná metoda

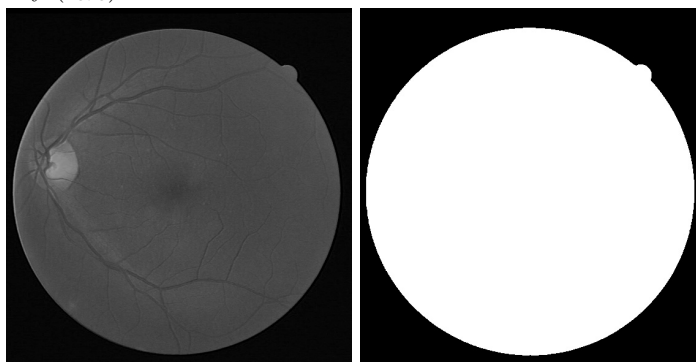
3.1 Předzpracování

Vstupní data budou označována jako I_{src} . Před samotnou segmentací je potřeba snímek upravit. Úprava spočívá v odstranění nepotřebných částí, zvýraznění hledaných vlastností a celkové úpravě, která je nutná před použitím vlastního algoritmu pro segmentaci. Předně je důležité podotknout, že vstupní data jsou v barevné formátu BRG. Po rozložení na základní barevné spektrum nám vzniknou I_{red} pro červené, I_{green} pro zelené a I_{blue} pro modré spektrum. Z 3.1 je zřejmé, že největší vypovídající hodnotu má zelená složka. K tomuto závěru došla většina prací uvedených v 2.1.

Před aplikování dalšího kroku je nutné I_{green} invertovat a použít příslušnou masku pro odstranění okrajů. Výstup tohoto kroku je $I_{greenInv}$.



(a) první snímek z DRIVE[6] (b) červené spektrum (I_{red}) (c) zelené spektrum (I_{green}) testovací množiny (I_{src})



(d) modré spektrum (I_{blue}) (e) příslušná maska snímku

Obrázek 3.1: První snímek rozložený na RGB složky a jeho příslušná maska.

Pro zvýraznění cév je dále vhodné aplikovat morfologickou operaci, známou jako *horní část klobouku* [32]. Ta se dá reprezentovat jako:

$$I_{top_hat} = I_{greenInv} - I_{greenInv} \circ kernel, \quad (3.1)$$

kde \circ reprezentuje morfologické otevření a $kernel$ vhodně zvolenou matici.

Pro vyrovnání rozdílů je pak potřeba použít adaptivní vyrovnání histogramu (dále jen jako zkratka CLAHE). CLAHE funguje jako standardní vyrovnání histogramu s tím rozdílem, že nebere v potaz celý snímek, ale pouze určité okolí. Po aplikování těchto kroků by výsledek měl odstranit rozdíly v celkové světlosti, jako je například oblast optického disku v porovnání s protějším okrajem snímku.

3.2 Segmentace cévního řečiště

Samotná metoda navržená pro segmentaci se řadí mezi metody z sekce **Přizpůsobené filtrování** (viz 2.1) s tím rozdílem, že budou použity filtry založené na normálním rozdělení a na jeho druhé derivaci. Filtry se vždy sestavují na míru pro každou databázi, v našem případě bude hlavní databáze DRIVE, ale databáze STARE má velmi podobné vlastnosti. Při sestavování sady filtrů je potřeba brát v potaz řadu faktorů [1]:

- Většina cév, které se v datech nachází, jsou v rozsahu 2-10 pixelů.
- Cévy se dají charakterizovat po segmentech, které mají antiparalelní charakter.

- Cévy absorbují větší množství světla než zbylé okolní tkáň, takže mají tmavší charakter oproti okolí.
- Šířka cévy se postupně zmenšuje se zvyšující se vzdáleností od optického disku.
- Cévní profil lze aproximovat křivkou normálního rozložení.

Sestavení filtru pomocí Gaussovy křivky lze zjednodušit jako:

$$K_{gauss}(x, y) = \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right), \quad (3.2)$$

kde μ určuje střed a σ šířku. Podobně jako u 3.2 lze použít i druhou derivaci této funkce.

$$K_{gauss_second}(x, y) = \left(\frac{d}{dx}\right)^2 K_{gauss}(x, y) \quad (3.3)$$

Filtr se pak aplikuje na každý pixel zvlášť a vygeneruje hodnotu v intervalu 0 až 255. Filtr sestavený pomocí výše uvedeného postupu je schopný generovat vysokou odezvu pouze pro cévy orientované stejně jako je on. Tento problém lze vyřešit rotací filtru o 180° . Po překročení 180° by se totiž filtry opakovaly. Není nutno rotovat filtr po jednom stupni, v mnoha publikacích jako například [1][20] se ukázalo, že krok 15° je dostačující. Z jednoho profilu tedy vznikne 12 filtrů, které jsou aplikovány na snímek a tím je vygenerována výsledná množina dat, kde každý prvek náleží jednomu filtru.

Po filtraci se výstupy všech filtrů sloučí. Daný pixel pak bude mít hodnotu příslušnou nejvyšší hodnotě ze všech filtrů. Pro výsledné vyhodnocení, zda má mít pixel hodnotu 0 (pozadí), nebo 255 (céva), je potřeba aplikovat prahování. Pro definici navrhovaného prahování je potřeba si definovat pomocné funkce:

$$\begin{aligned} Val(x, y) &= \text{Hodnota pixelu,} \\ Max(I) &= \text{Maximální hodnota,} \\ Edge(x, y, t) &= \frac{Max(x, y, t)}{t}. \end{aligned} \quad (3.4)$$

Prahování pak lze vyjádřit jako:

$$I_{thresholded}(x, y) = \begin{cases} 0 & \text{pro } Val(x, y) < Edge(x, y, t), \\ 255 & \text{pro } Val(x, y) \geq Edge(x, y, t). \end{cases} \quad (3.5)$$

$I_{thresholded}$ je pak očekávaný výsledek.

3.3 Nalezení žil a tepen

Následně je pak potřeba $I_{thresholded}$ upravit a rozdělit na žíly a tepny. Pro rozdělení je potřeba stanovit množinu bodů, u které se bude výpočet provádět. Proto je důležité nalézt středy cév a následně je rozdělit do menších segmentů, které budou představovat vstup rozhodovacího procesu.

3.3.1 Skeletonizace

Jelikož $I_{thresholded}$ obsahuje množinu všech pixelů, které byly vyhodnoceny jako cévy, tak je potřeba odvodit pouze středové pixely. Navržená funkce, spuštěná na každém pixelu cévy z $I_{thresholded}$, vyhodnotí střed a výsledek zapíše do $I_{skeleton}$.

3.3.2 Nalezení optického disku

Tento krok je rozhodující pro správné určení výchozích bodů pro odstartování sledování cév. V používaných datech není optický disk vždy na stejném místě, nemá stejný tvar a ani stejnou intenzitu. Ačkoliv intenzita v oblasti optického disku není stejná napříč snímky, je vždy výrazně vyšší oproti zbytku cévního pozadí. Lze tudíž použít čtvercový průměrovací filtr bez normalizace. Při vhodně zvolené velikosti filtru tak po aplikování filtru získáme histogram. Z tohoto histogramu je vybráno pouze horních $n\%$ hodnot. Z výsledné množiny je pak proveden průměr souřadnic a tím nalezen střed optického disku.

3.3.3 Výběr počátků cév

S nalezeným středem optického disku je potřeba zvolit správnou vzdálenost, ve které chceme vybrat počáteční body. Vzdálenost by měla být větší než poloměr samotného optického disku. S danou vzdáleností je sestrojena kružnice okolo optického disku. Optický disk pak musí celý ležet uvnitř kružnice. Kružnici je následně přenesena na $I_{skeleton}$, a v místech, kde protne cévu, budou ležet počátky cév pro sledování.

3.3.4 Sledování

Algoritmus na sledování cévy bude spuštěn od každého vstupního bodu zvlášť. V každém kroku se použije čtvercová mřížka se středem v aktuálním bodu na $I_{skeleton}$. V této mřížce se vyhledá pokračování cévy a vypočítá se její orientace. Pokud není nalezeno žádné pokračování, tak se vyhledávací mřížka rozšíří a krok se opakuje až do určité velikosti mřížky. Pokud je v mřížce nalezeno větvení, zapamatuje se a po skončení aktuální větve je odstartován nový začátek od něj. Výstupem sledování pak budou navazující segmenty představující cévní systém.

3.3.5 Rozlišení na žíly a tepny

Pro rozlišení cévních segmentů na žíly a tepny byly vybrány dva přístupy.

První přístup:

Pro rozlišení cévních segmentů na žíly a tepny byl použit postup uveden v článku [2]. Tento postup využívá faktu, že červené spektrum je náchylné na změny kyslíku. To lze vidět na I_{red} , kde jsou žíly tmavší, protože absorbují větší množství světla než tepny. Díky těmto znalostem lze vypočítat optickou hustotu v daném bodě, dále jen jako OD (z anglického *optical density*).

$$OD = \ln\left(\frac{I_{vessel}}{I_{background}}\right). \quad (3.6)$$

I_{vessel} zastupuje průměrnou intenzitu cévních pixelů v okolí a $I_{background}$ průměrnou intenzitu pixelů vyhodnocených jako pozadí. Hodnoty OD se pak vypočítají pro zelené a červené spektrum zvlášť. Tyto hodnoty zastoupené v proměnných OD_{red} a OD_{green} jsou následně využity pro výpočet poměru optické hustoty, dále jen jako ODR (z anglického *optical density ratio*).

$$ODR = \ln\left(\frac{OD_{red} - OD_{green}}{OD_{green}}\right) \quad (3.7)$$

ODR je vypočítáno pro všechny segmenty. Pokud je *ODR* segmentu nižší než průměrné *ODR* všech segmentů, tak je označen jako žíla, jinak se jedná o tepnu.

Druhý přístup:

Další přístup je adaptace z [12]. Tento postup využívá 3 metriky: střední odchylka okolí pixelu v zeleném kanálu (dále jen **STDE_GREEN**), směrodatná odchylka okolí pixelu v zeleném kanálu (dále jen **MEAN_GREEN**) a směrodatná odchylka okolí pixelu v HUE kanálu z barevného prostoru HVS (dále jen **MEAN_HUE**). Původní množina všech výchozích bodů je opět rozdělena na tři podskupiny podle šířky cévy. Tyto metriky se vypočítají pro každý pixel podmnožiny. Na množinu výsledků se pak aplikuje shlukovací algoritmus **K-means**. Výstup algoritmu jsou pak body určující středy shluků a každý bod bude přidělen k nejbližšímu středovému bodu. Jeden shluk pak bude označen jako žíly a druhý jako tepny. Výsledky všech podmnožin jsou následně spojeny do jednoho výsledného snímku.

Kapitola 4

Implementace

Pro implementaci byl zvolen jazyk Python, přesněji ve verzi 3.9.12. Pro práci s obrázky je většinou použita knihovna OpenCV[21], pro různé matematické operace a struktury knihovna NumPy[19]. Všechn kód je dostupný na adrese <https://github.com/RadekSedlar/ArteriesVesselsSegmentation>.

4.1 Předzpracování

Pro získání I_{src} z datového souboru je použita funkce `cv2.imread(path)`, která vrací objekt jako dvourozměrné pole, kde má každá položka typ **tuple** tří čísel. Čísla jsou ve tvaru **uint8**, což jsou bezznaménkové číselné hodnoty vyjádřeny v osmi bitech, tedy hodnoty 0 až 255. Je vhodné podotknout, že funkce vrací matici seřazenou po řádcích a intenzity barev v buňkách jsou ve formátu BGR. Pro získání konkrétních barevných složek je použita funkce `cv2.split(I_src)`, jejíž výstup lze pozorovat na obrázku 3.1. Tato funkce vrací **tuple** oddělených barevných složek, ze kterých se dále bude používat pouze prostřední hodnota reprezentující zelenou složku. Dále je potřeba I_{green} invertovat. Invertování je přímočará operace, kdy se nová hodnota snímku vypočítá jako rozdíl maximální hodnoty a hodnoty aktuální. Tuto operaci implementuje funkce `I_greenInv = cv2.bitwise_not(I_green)`, jejíž výsledek lze vidět na obrázku 4.1a. Následující krok aplikuje morfologické operace popsány sekci **Předzpracování** (viz 3.1). Knihovna OpenCV má všechny hlavní morfologické operace implementované v rámci funkce `cv2.morphologyEx()`. Pro aplikování různých operací je zapotřebí zvolit správně parametry pro volání funkce. Pro získání kýženého výsledku potom vypadá volání následovně:

```
1 I_top_hat = cv2.morphologyEx(I_green_inverted, cv2.MORPH_TOPHAT,  
2 cv2.getStructuringElement(cv2.MORPH_RECT, (11,11)))
```

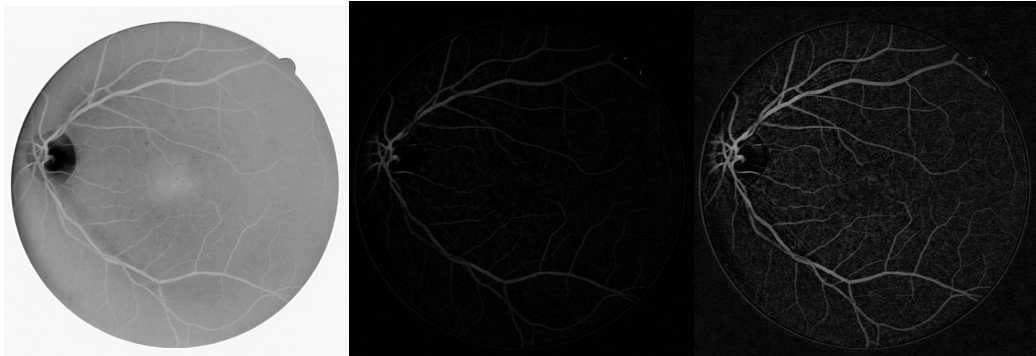
Volání `cv2.getStructuringElement(cv2.MORPH_RECT, (11,11))` vrací čtvercovou jednotkovou matici s rozměry 11x11. Jako poslední krok je potřeba aplikovat vyrovnání histogramu. Problém s klasickými vyrovnáním histogramu je ten, že bere v potaz celý vstupní snímek najednou. Pro zahrnutí vlivu nejbližšího okolí je potřeba použít přístup se zkratkou **CLAHE** (Contrast Limited Adaptive Histogram Equalization). Tento přístup vytváří posuvné okno okolo aktuálního pixelu a tím pádem se nebere v potaz celý snímek, ale pouze nejbližší hodnoty. Knihovna OpenCV opět poskytuje tuto funkcionalitu. Použití vypadá takto:

```

1 clahe = cv2.createCLAHE(clipLimit=3, tileGridSize=(10, 10))
2 I_clahe = clahe.apply(I_top_hat)

```

Parametry pak nastavují práh pro omezení kontrastu a velikosti okolí. Hodnota I_{clahe} je pak očekávaný výsledek.



(a) invertované zelené spektrum (I_{green}) (b) aplikování morfologických operací (I_{top_hat}) (c) aplikování CLAHE (I_{clahe})

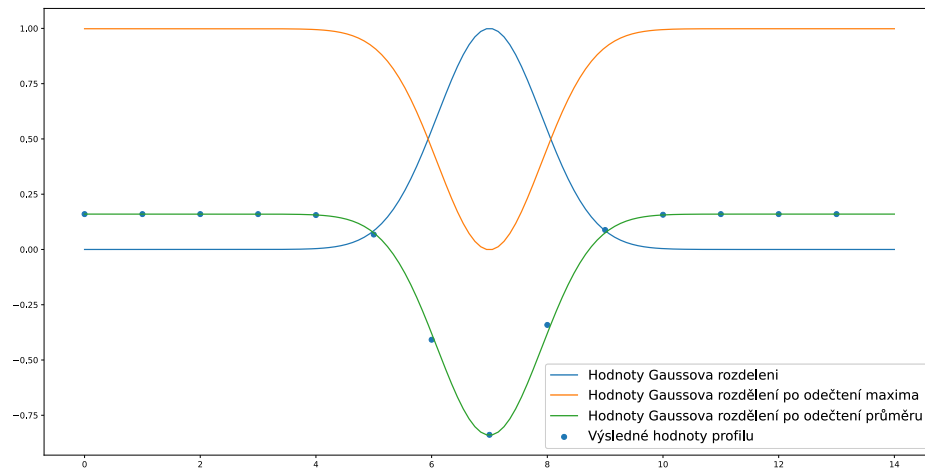
Obrázek 4.1: Příklad výstupů procesu předzpracování

4.2 Segmentace cévního řečiště

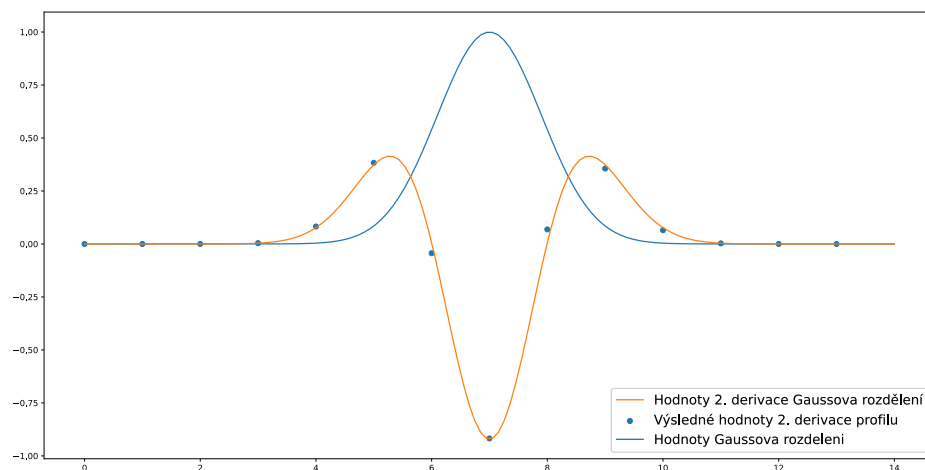
Nejdůležitější část celé segmentace je vytvoření správného cévního profilu. Tento profil se sestavuje na základě několika parametrů:

- μ - posunutí profilu na ose X,
- σ - šířka rozprostření funkce,
- *profilSize* - výsledná šířka čtvercové matice reprezentující masku.

Profil se vypočítá pomocí vztahu uvedeného v kapitole 3.2. Tím vznikne pole hodnot o délce *profilSize*. Hodnotu je potřeba invertovat a následně normalizovat. Příklad výstupních hodnot tohoto postupu je ukázán na obrázcích 4.2a a 4.2b.



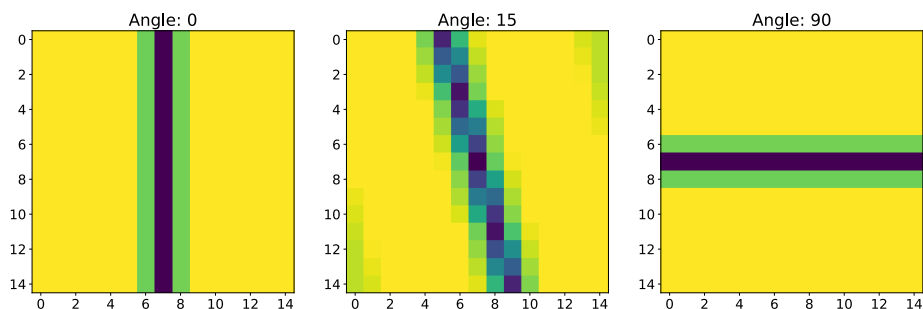
(a) Normální rozdělení.



(b) Druhá derivace normálního rozdělení.

Obrázek 4.2: Příklad získání hodnot pro profil z hodnot normálního rozdělení s parametry $\mu = 7, \sigma = 0.9$.

Profil je následně duplikován tolikrát, kolik má hodnot. Duplikované profily se naskládají do listu za sebou a tím se vytvoří matice, na kterou se dále budu odkazovat jako na masku. Vytvořenou masku je potřeba otočit okolo svého středu o 180° s krokem 15° . Tím vznikne 12 masek, které jsou použity jako filtry. Pro otočení masky je použita funkce `rotate(image, angle=angle)` z modulu `imutils[23]`. Ukázkou masky a některých úhlů je možné vidět na obrázku 4.3.



Obrázek 4.3: Příklad masky otočené o 0° , 15° a 90° .

Veškeré filtrování pak probíhá ve funkci:

```
1 apply_kernel_on_image_in_angles(grayScaleImage, kernel, show_steps=False, threshold_limit =
  5)
```

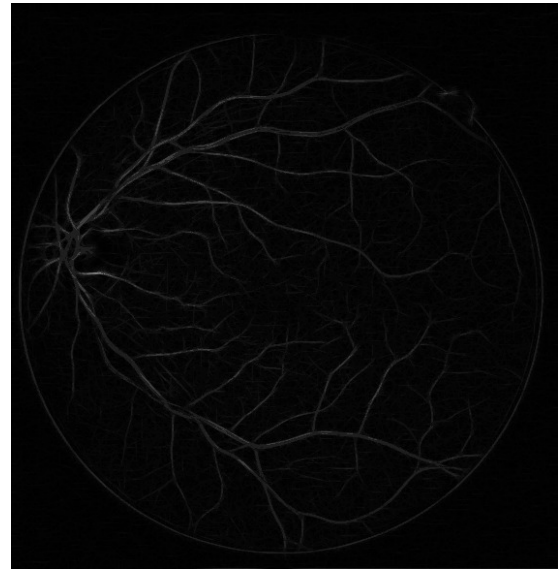
Prvním parametrem je výstup ze sekce **Předzpracování** (viz 4.1), druhý je výše zmiňovaná maska, třetí je parametr pro ladění, který, pokud je nastavený na **True**, tak ukazuje průběžný výsledek každého kroku a poslední parametr je prahovací limit. Pro každý krok je tedy jeden výsledek, který se vygeneruje za pomoci funkce:

```
1 cv2.filter2D(grayScaleImage, -1, filter)
2
```

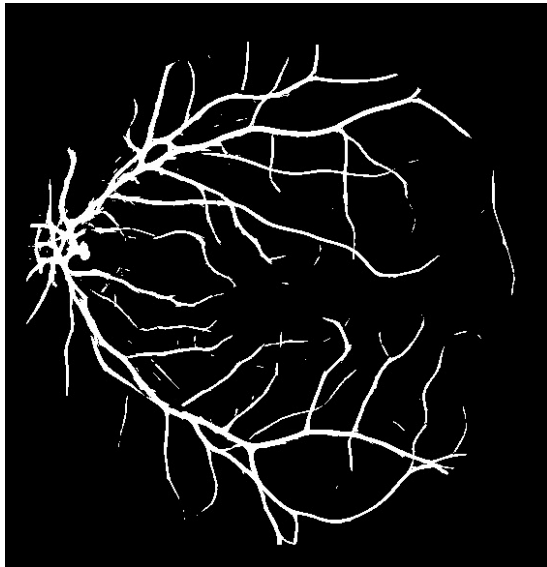
Výsledky průběžného filtrování se pak spojují do finálního obrázku pomocí funkce `cv2.bitwise_or(finalResult, stepResult)`. Funkce vrací **tuple** dvou hodnot a to výsledky filtrování s prahováním a bez prahování. Jak je vidět na obrázku 4.4, filtry vytvořené z druhé derivace Gaussova rozdělení mají menší odezvu, lépe ale reagují na menší detaily. Nakonec je zvolena kombinace dvou filtrů, přičemž jeden je sestaven na základě Gaussova rozdělení a druhý na jeho druhé derivaci. Pro zjištění nejlepší kombinace je potřeba zjistit parametry pro vytvoření filtrů.



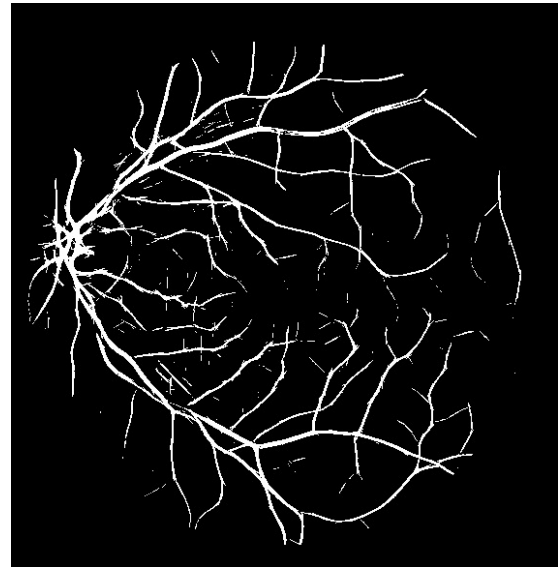
(a) Výstup filtru sestaveného na základě Gaussova rozdělení před prahováním.



(b) Výstup filtru sestaveného na základě druhé derivace Gaussova rozdělení před prahováním.



(c) Výstup filtru sestaveného na základě Gaussova rozdělení po prahování.



(d) Výstup filtru sestaveného na základě druhé derivace Gaussova rozdělení po prahování.

Obrázek 4.4: Výstup filtrovací funkce

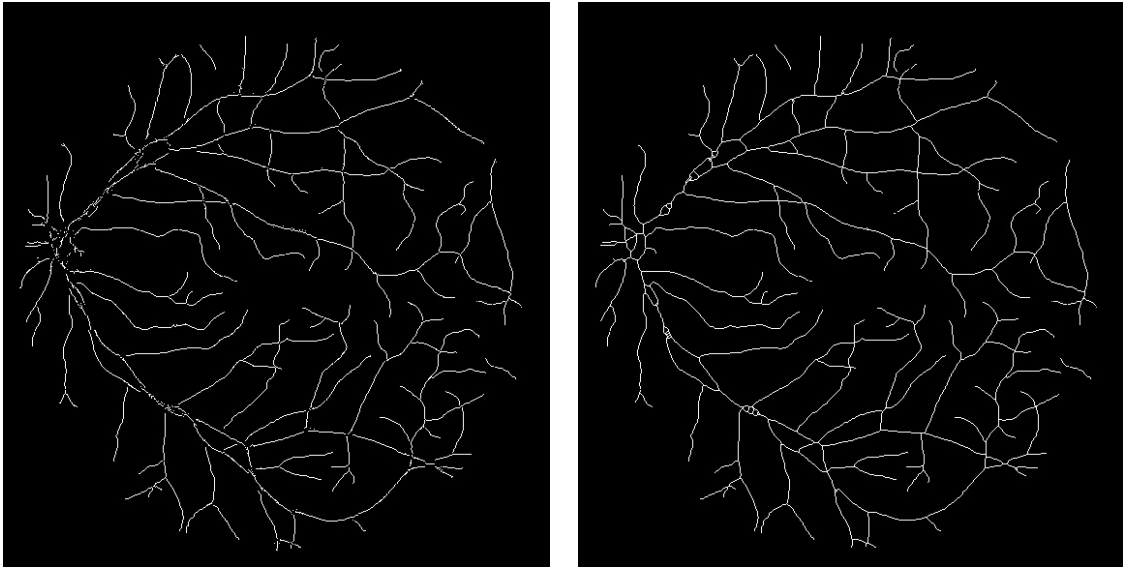
Pro zjištění nejlépe vyhovujících parametrů algoritmu byly vytvořeny filtry s parametrem σ v rozmezí 0.8 až 1.5 s velikostí kroku 0.1, rozmezí parametru pro prahování bylo stanoveno na hodnoty od 2 do 6.

4.3 Skeletonizace

Prvotní pokus o implementaci byla vlastní funkce. Tato funkce byla spuštěna nad každým pixelem segmentovaného obrazu. Pokud byl pixel vyhodnocen jako céva, tak bylo v několika směrech testováno, kterým směrem leží nejméně pixelů vyhodnocených jako cévy. Směr s

nejmenším počtem za sebou ležících cévních pixelů pak byl zvolen jako poloměr cévy. Z poloměru a směru se následně vyhodnotil středový pixel cévy. Ačkoliv tento přístup na první pohled vypadal jako dostačující, tak po podrobném prozkoumání výsledku se ukázal jako nedostačující. Hlavní problém byl v místech, kde se střetávaly nebo rozdělovaly cévy.

Proto byla pro skeletonizaci zvolena funkce `skeletonize` z modulu `skimage`[27]. Tato funkce přijímá pouze hodnoty 0 a 1, takže před a po použití je nutno data upravit. Funkce `skeletonize` nabízí volbu metody skeletonizace jako volitelný argument. Pro nejlepší výsledek byl volitelný argument nastaven na hodnotu `lee`. Srovnání navrhované metody a následně použité metody lze pozorovat na obrázcích 4.5a a 4.5b.



(a) Výstup vlastní implementace

(b) Výstup `skimage.morphology.skeletonize()`

Obrázek 4.5: Výsledky přístupů skeletonizace.

4.4 Nalezení optického disku

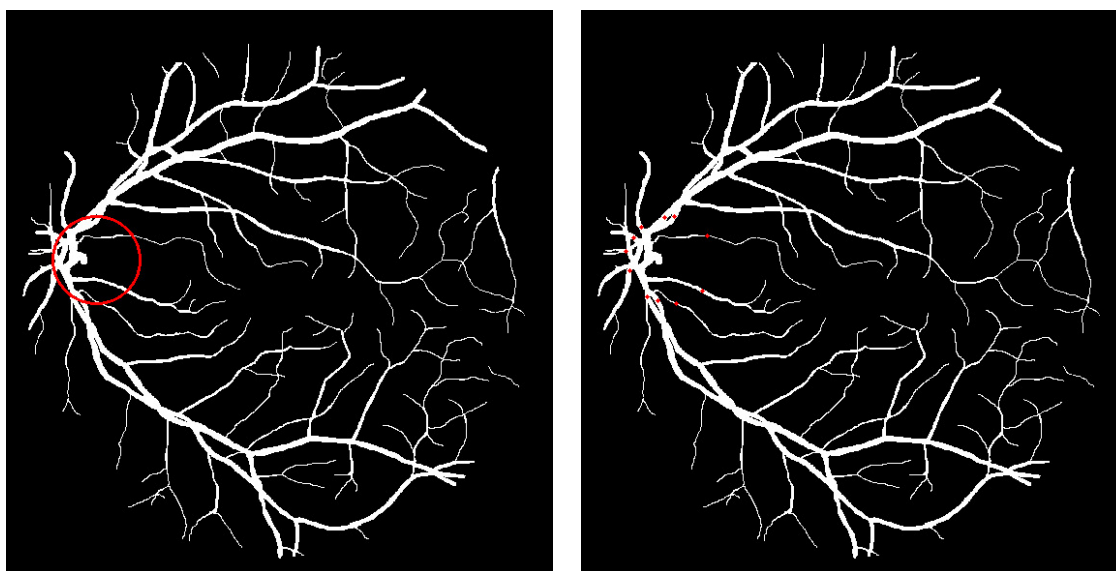
Postup navrhovaný v sekci **Nalezení optického disku** (viz 3.3.2) se ukázal být efektivní pouze pro první snímek, proto bylo potřeba přístup změnit. Další postup byl založený na posuvném okně, které sečetlo hodnoty všech pixelů okolo aktuálního středového pixelu. Toto okno bylo jako filtr použito na každý pixel I_{clae} . Pixel s největší odezvou pak byl považován za středový pixel optického disku. Tento postup přinesl výrazně lepší výsledky, takže byl dále rozvíjen. Další úprava byla změna tvaru filtru. Od předchozího čtvercového tvaru bylo upuštěno a nahradil ho filtr kruhovitěho tvaru. Tato poslední modifikace upřesnila výsledky na použitelnou mez.

4.5 Výběr počátku cév

Pro výběr počátečních bodů je potřeba vytvořit kružnici kolem optického disku. Střed této kružnice je propagován do výstupu z podkapitoly **Segmentace cévního řečiště** (viz 4.2). Měřením optických disků a dalším experimentováním byla stanovena hodnota 60 pixelů

jako poloměr kružnice. Tato hodnota je vyšší, než z měření vyplývá, ale je potřeba zahrnout chybu vytvořenou při hledání středu kružnice. Pro nalezení souřadnic bodů ležících na kružnici byla vytvořena funkce

`listOfCoordinates(radius, x_of_middle=0, y_of_middle=0)`, která vrací seřazené souřadnice ležící na kružnici. Správné uspořádání souřadnic je klíčové pro zjištění středových pixelů cév, kterými kružnice prochází. Střed cév je nalezen pomocí procházení pole souřadnic, a když je souřadnice vyhodnocena jako céva, tak je zařazena do pomocného pole. Pole bude naplňováno, dokud nebude další souřadnice vyhodnocena jako pozadí. V tom okamžiku se z pomocného pole vybere prostřední prvek a ten bude reprezentovat počátek cévy. Je také nutné, aby se při procházení pole brala v potaz jeho cykličnost. Tím je myšlen fakt, že první pixel, který je v poli, musí být pozadí. Toho je docíleno tím, že pokud jako první bude pixel cévy, tak bude přesunut na konec pole. Pro nalezení počátečních bodů byl použit I_{clahc} a ne navrhovaný $I_{skeleton}$, protože při použití $I_{skeleton}$ byly některé cévy ignorovány. Příklad vizualizace očekávaného výsledku je uveden na obrázku 4.6.



(a) Okolí optického disku

(b) Nalezené počátky cév

Obrázek 4.6: Práce s optickým diskem a hledání počátků cév.

4.6 Sledování

S nalezenými počátky cév a středem optického disku je možné začít algoritmus pro sledování cév. Prvotní pokus o implementaci využíval výsledek segmentace. V každém počátečním bodě se určí, kterým směrem céva může pokračovat. Tento směr je zjištěn jako úhel, který tvoří bod reprezentující střed optického disku a bod reprezentující počátek cévy, kde je střed optického disku brán jako střed kružnice a počátek cévy je pak bod na kružnici. S nalezeným směrem byl vypočítán maximální a minimální úhel, kterým může céva pokračovat. Tyto úhly byly vypočítány přičtením a odečtením $\frac{\pi}{2}$. Poté následoval cyklus, který prošel všechny možné úhly s krokem $\frac{\pi}{12}$ a zjistil, kolik pixelů cév tímto směrem leží. Úhel s největším počtem cévních pixelů byl zvolen jako směr pokračování. Tímto směrem byl pak zvolen nový bod a společně s minulým bodem, jakožto bodem výchozím, algoritmus pokračoval, dokud nenarazil na bod, kde v každém možném úhlu nebyly žádné další pokračující

pixely. Tento postup fungoval správně, dokud nedošlo k rozdělení cévy. V těchto případech se pokračovalo pouze směrem s největší shodou. V této části byla snaha o implementování detekce tohoto rozdělení pomocí variabilní hranice, nad kterou by počet pokračujících pixelů znamenal pokračování cévy. Tato hranice byla vypočítávána na základě tloušťky cévy. Pokud existovalo více úhlů nad touto hranicí, byl zvolen hlavní směr, kterým pokračovala céva. Další vyhovující směry pak musely být od hlavního minimálně $\frac{\pi}{4}$. Tento přístup se však ukázal jako nevhovující, takže musel být přepsán.

Velká změna oproti první implementaci je použití výstupu skeletonizace jako prostor pro sledování. Tento přístup si uchovává list bodů, dále označován jako **nodes**, zastupujících začátky cévního segmentu a jejich směr. Prvotní nodes jsou naplněny výstupem ze sekce 4.5 a směr je vždy určen jako úhel vytvořený na kružnici, kde střed optického disku je střed kružnice a aktuální bod pak leží na kružnici. S těmito výchozími hodnotami v **nodes** pak můžeme spustit samotné mapování cévního segmentu.

Každá iterace algoritmu vyhledá body v okolí aktuálního pixelu, které jsou v určeném rozsahu od aktuálního směru. Okolí aktuálního bodu je určeno jako pixely v šachovnicové vzdálenosti 1 a 2 bez vnitřních pixelů.



(a) Šachovnicové okolí 1 bez středových pixelů. (b) Šachovnicové okolí 2 bez středových pixelů.

Obrázek 4.7: Okolí použité pro sledování cév. Zelené pixely reprezentují okolí a modrý je středový pixel.

Ve vzdálenosti 1 jsou pak určeny pixel, které spadají do intervalu určeného aktuálním směrem. Dále jsou zvažovány pouze pixely, které nejsou vyhodnoceny jako pozadí. Další krok závisí na počtu nalezených vyhovujících pixelů:

Žádný - Pokud není nalezen žádný vyhovující pixel, tak se jedná o konec cévy. Začátek zmapovaného cévního segmentu je odstraněn z **nodes** a přidán k ostatním už zmapovaným segmentům.

Jeden - V tomto případě je minulý pixel nastavený na pixel aktuální a aktuální je nastaven na jediný nalezený. Je přepočítán aktuální směr a mapování cévního segmentu pokračuje dál s novými hodnotami.

Více než jeden - Mapování aktuálního cévního segmentu končí. Všechny nově nalezené pokračování jsou přidány do **nodes**. Začátek zmapovaného cévního segmentu je odstraněn z **nodes** a přidán k ostatním už zmapovaným segmentům.

Algoritmus pak pokračuje, dokud **nodes** není prázdný. Po dokončení algoritmu získáme list všech počátků cévních segmentů.

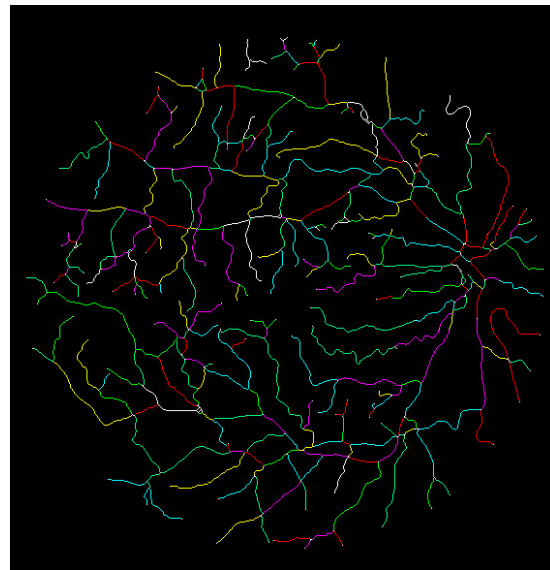
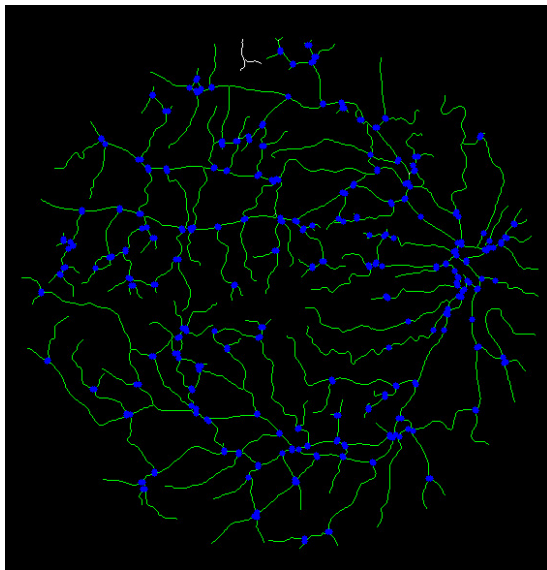
4.7 Rozlišení na žíly a tepny

4.7.1 První přístup

Původní záměr byl provést rozlišení pouze u kořenů cév vycházejících z optického disku. Tento přístup by teoreticky fungoval, pokud by se ve snímku nenacházelo křížení cév. Prakticky se ale cévy kříží až moc často, proto musel být přístup upraven.

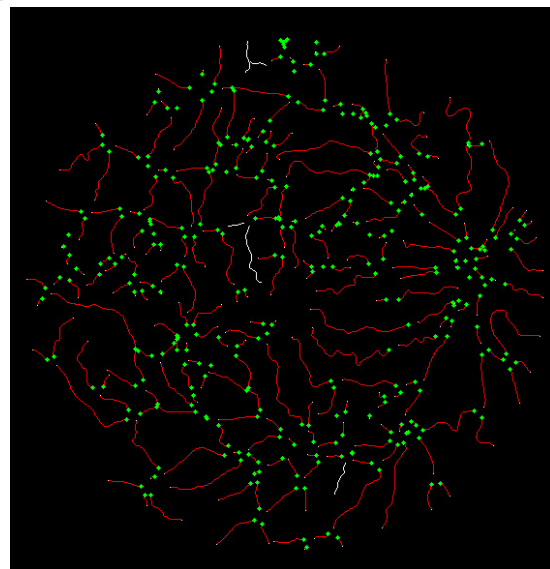
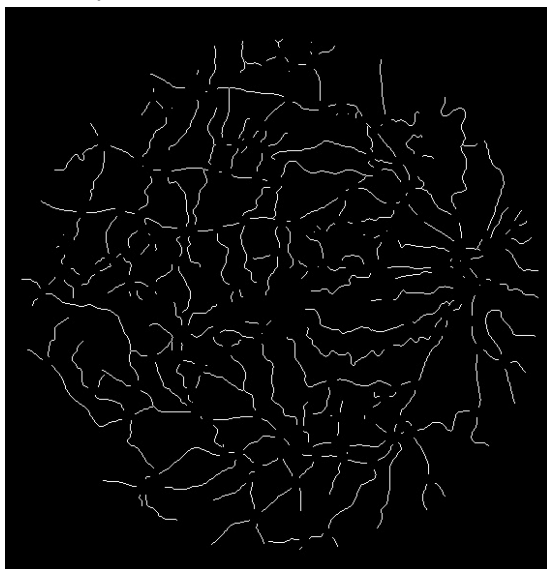
Výstup posledního kroku **Sledování** (viz 4.6) je každý cévní segment, který nepřerušovaně pokračuje nejméně dva pixely. Původně se průměrný *ODR* vypočítával globálně ze všech segmentů. To ale mělo za následek nesmyslné výsledky, kde všechny segmenty blíže k optickému disku byly vyhodnoceny jako tepny a ostatní jako žíly. Tento výsledek byl způsoben zanedbáním vzdálenosti od optického disku. Pro zohlednění světlosti u optického disku bylo potřeba segmenty rozdělit na tři skupiny podle vzdálenosti od něj.

Rozdělení počátečních bodů do tří skupin podle vzdálenosti od optického disku přineslo lepší výsledky, ty ale stále nebyly uspokojivé. Největší problém byl, že při rozdělení nebo křížení cév byly počáteční body segmentů příliš blízko u sebe. Tento problém byl vyřešen maskováním všech modře vyznačených bodů na obrázku 4.8a. Pro nalezení nových počátečních bodů bylo spuštěno vyhledávání konce z původních bodů a při nalezení prvního vyhovujícího pixelu, se pixel přidal do množiny nových počátků cévních segmentů. Výsledná množina nových počátků segmentů je pak vyznačena na obrázku 4.8d.



(a) Cévní segmenty s modře vyznačenými začátečními body.

(b) Cévní segmenty obarvené na náhodné barvy pro ilustraci.



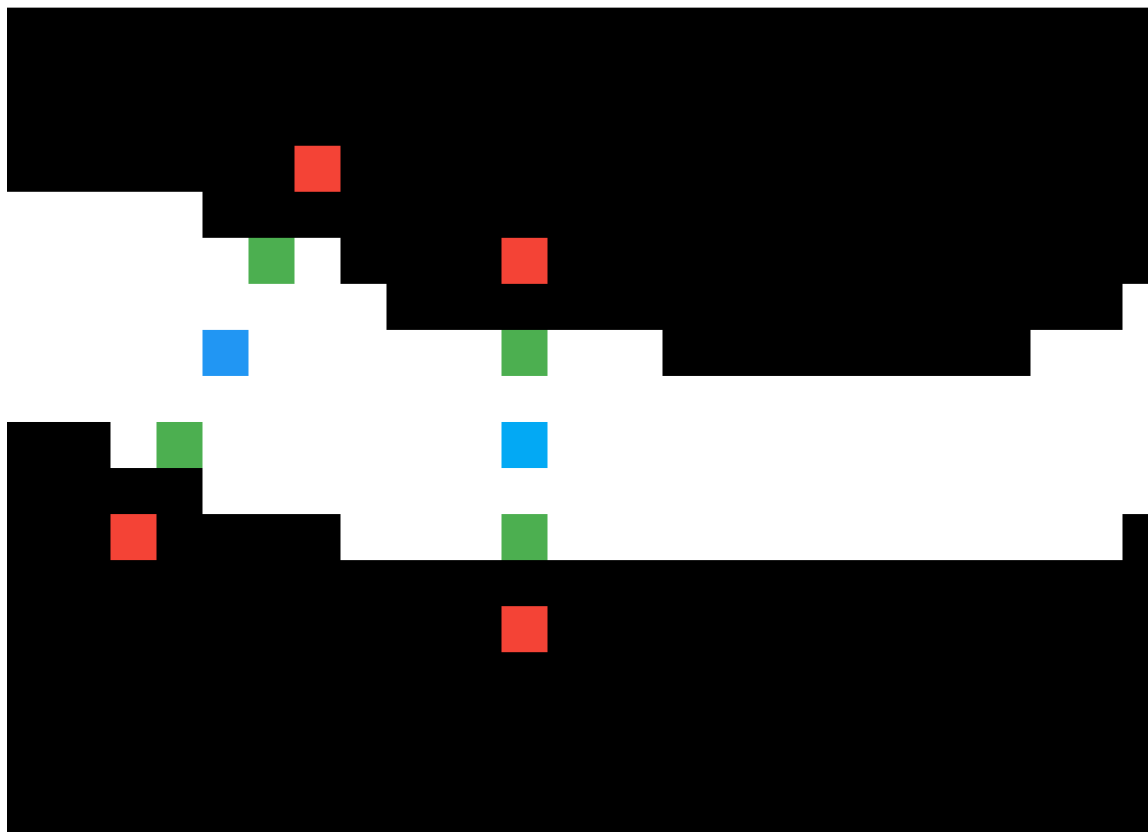
(c) Středové pixely bez bifurkace a míst kde se cévy kříží.

(d) Nově vyznačené body znázorňující počátky cévních segmentů.

Obrázek 4.8: Ukázka rozdělení cévní struktury na menší segmenty.

Rozdělení cév na žíly a tepny podle *ODR* a vzdálenosti od optického disku se bohužel ukázalo být stále neefektivní, proto byla vzdálenost od optického disku nahrazena šířkou aktuální cévy. Segmenty jsou tedy rozděleny na malé, střední a velké podle šířky. *ODR* je pak vypočteno podle vzorce 3.7. Pro tento výpočet je potřeba vhodně vybrat odpovídající pixely. Střed je tedy kořenový bod cévního segmentu a společně s ním je znám i úhel, kterým céva pokračuje. Díky směru je možné v pravých úhlech hledat body představující okraje cévy. Se známým středem a okrajovými body lze zjistit šířku cévy, která je následně použita pro zařazení cévy do správné skupiny. Body pozadí jsou nalezeny tak, že se bude pokračovat

od středového bodu 2 krát větší vzdálenost stejným směrem, kterým leží okrajový pixel. Příklad výběru vyhovujících pixelů je znázorněn na obrázku 4.9.

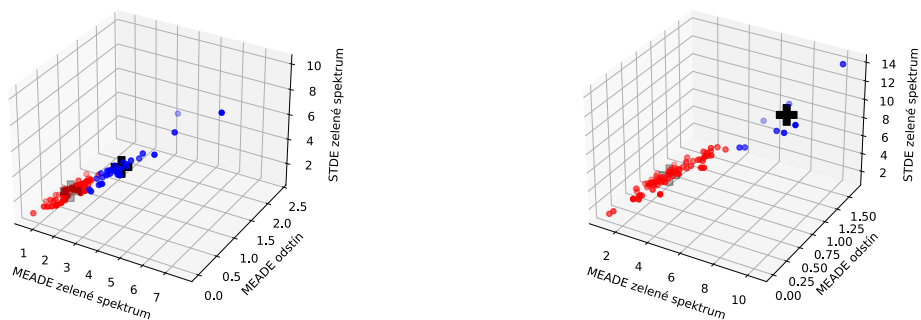


Obrázek 4.9: Výběr pixelů pro výpočet ODR. Černé pixely jsou pozadí a bílé céva. Střed cévy pak představuje modrý pixel a okraje jsou vyznačeny zeleně. Červené jsou vybrané pixely pro reprezentaci pozadí.

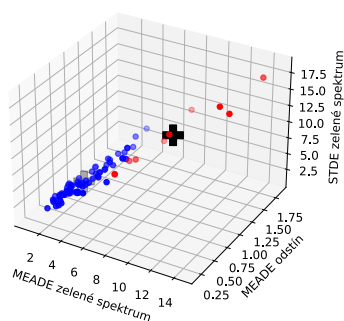
4.7.2 Druhý přístup

Tento přístup opět rozděluje všechny výchozí body cévních segmentů na tři podskupiny na základě jejich šířky. Body pro výpočet metrik byly nalezeny jako jednotkové šachovnicové okolí aktuálního pixelu. Z vybraných pixelů a jejich okolí se pak vypočítá **průměrná absolutní odchylka** ze zeleného spektra (dále jen **MEADE** z anglického *mean absolute deviation*), **průměrná absolutní odchylka** ze odstínu a **směrodatná odchylka** ze zeleného spektra (dále jen **STDE** z anglického *standard deviation*). Python disponuje modulem **statistics** pro výpočet metrik **STDE** a **MEADE**, takže vlastní implementace

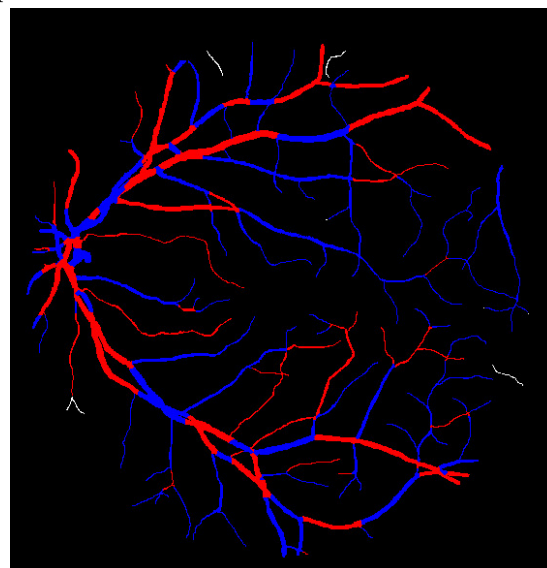
nebyla potřeba. Pro zařazení do shluků pak byl použit modul **cluster** z knihovny [27]. S pomocí středových bodů shluků se pak určí do kterého shluku aktuální bod patří. Výsledky všech podskupin jsou pak zkompileovány do jednoho výsledku. Příklad výstupu je uveden na obrázku 4.10.



(a) Cévní segmenty s modře vyznačenými počátečními body. (b) Cévní segmenty obarvené na náhodné barvy pro ilustraci.



(c) Středové pixely bez bifurkace a míst kde se cévy kříží.



(d) Nově vyznačené body znázorňující počátky cévních segmentů.

Obrázek 4.10: Ukázka rozdělení cévní struktury na menší segmenty.

4.8 Dokončení

Po vyplnění všech cévních segmentů, je potřeba reflektovat výsledky do segmentovaného výsledku. Toho je docíleno tím, že všechny pixely, které jsou označeny jako žíly a tepny, se stejně označí na výsledném obrázku z **Segmentace cévního řečiště** (viz 4.2). Následně se každý nevyplněný pixel změní na příslušný barevný pixel, který se nachází v jeho nejbližším okolí. Poslední krok se opakuje do té doby, dokud se změnil alespoň jeden pixel. Tímto se dosáhne podobného výsledku jako při záplavovém algoritmu.

Kapitola 5

Zhodnocení výsledků

5.1 Hodnocení metod

Jelikož jsou jednotlivé metody značně odlišné, bylo potřeba přijít se sadou kritérií, která by dokázala zhodnotit výsledky napříč rozdílným metodám. Další velkou roli hraje data-báze snímků, nad kterou se měření provádí, protože každá má rozdílné rozlišení, poměr přítomných nemocí a celkovou kvalitu obrázků.

5.1.1 Měřená kritéria

Vždy se porovnává výstupní snímek dané metody a snímek ručně vytvořený zkušeným oftalmologem. Snímek od lékaře pak považujeme za očekávaný výsledek. Porovnáním těchto dvou snímků můžeme získat hodnotu pro měřená kritéria. Tyto hodnoty pak jsou:

- **TP**(True positive) správně vyhodnocená céva - Hodnota pixelu je očekávána jako céva a je vyhodnocena jako céva.
- **TN**(True negative) správně vyhodnocené pozadí - Hodnota pixelu je očekávána jako pozadí a je vyhodnocena jako pozadí.
- **FP**(False positive) chybně vyhodnocená céva - Hodnota pixelu je očekávána jako pozadí a je vyhodnocena jako céva.
- **FN**(False negative) chybně vyhodnocené pozadí - Hodnota pixelu je očekávána jako céva a je vyhodnocena jako pozadí.

Specificita

Specificita, dále jen **SP**, je míra schopnosti algoritmu detekovat správně pozadí snímku.

$$SP = \frac{TN}{(TN + FP)} \quad (5.1)$$

Senzitivita

Senzitivita, dále jen **SN**, je míra schopnosti algoritmu detekovat správně cévy.

$$SN = \frac{TP}{(TP + FN)} \quad (5.2)$$

Přesnost

Přesnost, dále jen **ACC** (od anglického accuracy), je měřena jako poměr správně vyhodnocených pixelů ke všem testovaným pixelům. Jedná se tedy o pravděpodobnost, s jakou bude pixel správně vyhodnocen.

$$ACC = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (5.3)$$

5.2 Nejvýkonnější metody

Se znalostmi ze sekce **Hodnocení** je tak možné srovnávat metody založené na zcela odlišných principech. Pro následné srovnání si můžeme uvést nejvýkonnější metody s různými přístupy. Tyto hodnoty budou následně použity k vyhodnocení účinnosti navrhovaných postupů. Tento způsob hodnocení lze použít k měření segmentace cév i jejich rozlišení na žíly a tepny. Bohužel k vyhodnocování rozpoznávání žil a tepen se nepodařilo nalézt uspokojivý výslednou množinu a tak společné hodnocení tohoto rozpoznávání prakticky neexistuje. Nejčastěji se ale používá poměr žil a tepen ve výsledném obrázku.

5.3 Výsledky navrhované metody pro segmentaci

Pro zjištění nejlepší kombinace profilů a úrovní jejich práhování byly pomocí hrubé síly vyzkoušeny všechny kombinace profilů vytvořených v sekci **Segmentace cévního řečiště** (viz 4.2). Z těchto výsledků vyšly profily uvedené v tabulce 5.1.

Profil založen na	σ	μ	hranice práhování
Gaussově rozdělení	0.8	7	4
druhé derivaci Gaussova rozdělení	0.8	7	3

Tabulka 5.1: Parametry použité metody.

5.3.1 DRIVE

Kombinace filtrů byla sestavována s pomocí DRIVE množiny, tím pádem byly očekávány nejlepší výsledky zde. Nejvyšší výsledek byl hned u prvního snímku a to zhruba 95.2 %. Průměrné ACC však nepřekonalo práci Odstreilik et al. 2013 [20], ve které byly filtry navržené pro množinu HRF.

Číslo obrázku	SP	SN	ACC
1	98.4%	69.5%	95.2%
2	98.4%	61.3%	93.4%
3	92.1%	71.2%	90.0%
4	98.8%	61.9%	92.4%
5	99.3%	45.2%	91.6%
6	98.0%	63.0%	93.7%
7	98.9%	61.0%	93.9%
8	98.6%	63.8%	93.6%
9	98.6%	62.7%	94.1%
10	98.3%	62.9%	94.2%
\bar{X}	98.0%	62.3%	93.2%

Tabulka 5.2: Výsledky segmentace provedené nad množinou DRIVE.

Metoda	SP	SN	ACC
Wang et al. 2017 [33]	97.33%	81.73%	97.67%
Moghimirad et al. 2012 [17]	99.4%	78.5%	96.6%
Odstrcilik et al. 2013 [20]	96.93%	70.60%	93.40%
Navrhovaná metoda	98.0%	62.3%	93.2%

Tabulka 5.3: Nejvýkonnější techniky segmentace pro DRIVE množinu.

Ve srovnání s nejlepšími metodami, které je viditelné v tabulce 5.3, se přibližuje pouze již zmíněné Odstrcilik et al. 2013[20] a v porovnání s ostatními je neefektivní. Výsledky na konkrétních snímcích jsou pak uvedeny v tabulce 5.2.

5.3.2 HRF

Snímky z množiny HRF jsou velmi odlišné od množiny DRIVE, proto bylo očekáváno, že výsledky nebudou dostačující. Tyto snímky se liší v průměrné šířce cév, což je hlavní faktor, který je potřeba vzít v potaz při sestavování filtrů. Z výsledků obsažených v tabulce 5.4 lze vyvodit, že největší vliv na snížení přesnosti metody má diabetická retinopatie. Průměrné výsledky sice byly nad 90 %, to ale v tomto případě znamená nedosažení chtěného výsledku, to můžeme vidět i na srovnání v tabulce 5.5. To taky lze vyzpozorovat z výsledku SN, který je vždy okolo 40 %. Nízké hodnoty SN znamenají velké množství chybně identifikovaných cév.

Snímky s diabetickou retinopatií				Snímky s glaucomem			
Číslo obrázku	SP	SN	ACC	Číslo obrázku	SP	SN	ACC
1	91.61%	41.27%	88.57%	1	96.01%	40.54%	91.61%
2	92.50%	40.25%	88.55%	2	95.99%	40.76%	91.11%
3	94.91%	43.12%	91.20%	3	96.69%	28.34%	92.00%
4	96.72%	35.92%	92.64%	4	95.97%	36.60%	91.43%
5	97.24%	44.38%	93.34%	5	93.63%	37.65%	89.28%
6	96.47%	35.04%	90.96%	6	94.89%	42.42%	90.63%
7	95.68%	41.97%	90.72%	7	96.70%	52.46%	93.29%
8	94.73%	44.20%	90.22%	8	96.87%	56.48%	93.62%
9	94.76%	33.90%	89.86%	9	94.76%	33.90%	89.86%
10	96.70%	37.45%	90.55%	10	97.49%	51.12%	93.88%
11	97.25%	43.62%	91.86%	11	98.10%	43.45%	93.14%
12	94.51%	41.80%	90.25%	12	96.97%	49.22%	92.29%
13	93.46%	39.56%	89.08%	13	97.56%	44.64%	93.29%
14	93.59%	45.66%	89.25%	14	95.86%	36.93%	90.92%
15	92.70%	40.86%	88.98%	15	96.65%	40.59%	91.78%
\bar{X}	94.86%	41.39%	90.4%	\bar{X}	96.27%	42.34%	91.87%

Snímky zdravých pacientů			
Číslo obrázku	SP	SN	ACC
1	97.62%	43.43%	91.06%
2	97.94%	46.82%	91.94%
3	97.26%	37.16%	89.72%
4	98.05%	40.04%	91.55%
5	97.07%	38.36%	90.77%
6	95.80%	50.54%	90.34%
7	98.07%	35.90%	91.49%
8	97.63%	41.22%	90.90%
9	99.02%	33.18%	92.94%
10	96.70%	37.45%	90.55%
11	96.57%	45.10%	90.86%
12	96.59%	44.25%	90.07%
13	96.96%	43.50%	91.43%
14	97.23%	40.86%	91.38%
15	96.91%	43.04%	91.89%
\bar{X}	97.29%	41.39%	91.12%

Tabulka 5.4: Výsledky segmentace provedené nad množinou HRF.

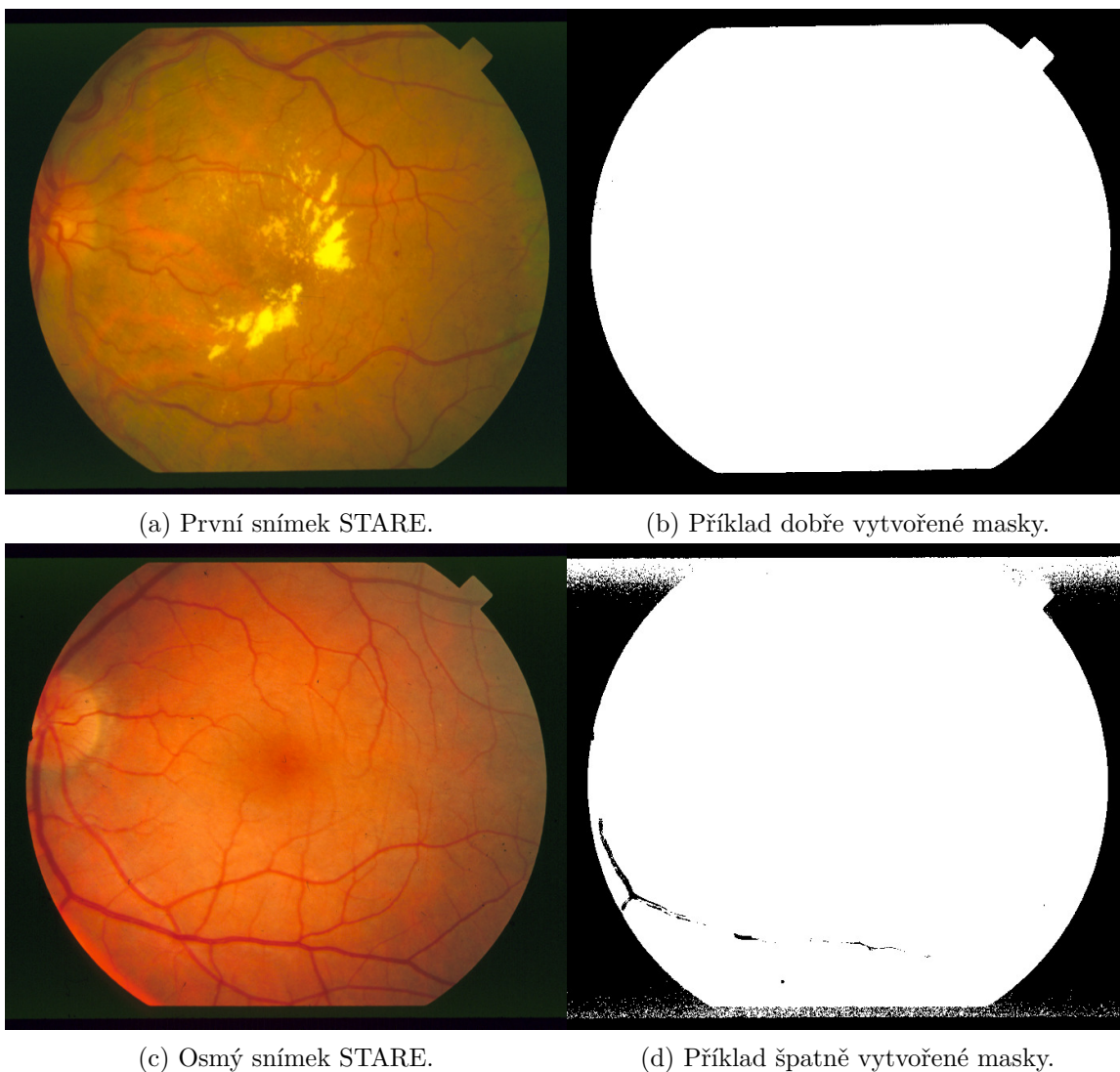
Metoda	SP	SN	ACC
Odstrčilik et al. 2013[20]	96.93%	70.60%	93.40%
Annunziata et al. 2015[3]	98.4%	71.3%	95.8%
Navrhovaná metoda	96.11%	42.12%	91.17%

Tabulka 5.5: Nejvýkonnější techniky segmentace pro HRF množinu.

5.3.3 STARE

Tato množina je svým rozlišením podobná DRIVE množině, proto byly očekávány podobné výsledky. Podle výsledků ale přesnost překročila výsledky dosažené na DRIVE. Existují dva pádné důvody, proč tomu tak je:

1. Manuální segmentace provedená expertem není tak kvalitní jako u DRIVE množiny.
2. Masky vytvořené ke každému snímku nejsou zcela přesné. Masky byly totiž vytvořeny pomocí globálního prahování a tím pádem několik z nich nebylo zcela přesných. Příklad lze vidět na obrázku 5.1.



Obrázek 5.1: Tvorba masek pro STARE.

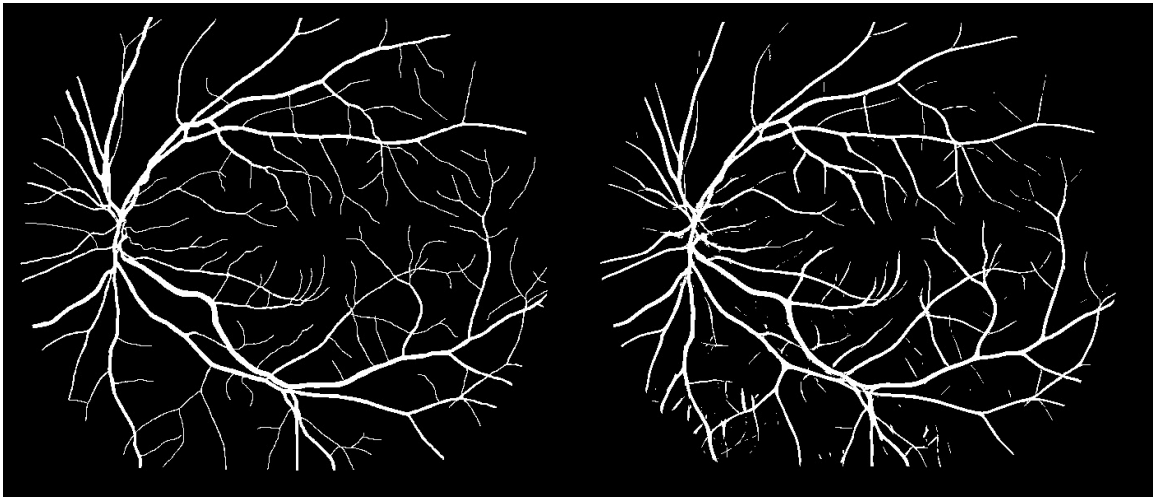
Číslo obrázku	SP	SN	ACC
1	94.44%	66.26%	91.42%
2	94.44%	56.28%	91.06%
3	92.11%	82.23%	91.30%
4	98.52%	57.79%	94.34%
5	94.98%	74.95%	92.44%
6	96.79%	78.24%	95.13%
7	96.17%	87.18%	95.26%
8	96.22%	84.74%	95.17%
9	96.43%	86.07%	95.32%
10	97.29%	73.97%	94.75%
11	97.00%	82.91%	95.59%
12	97.48%	86.05%	96.25%
13	97.04%	79.36%	94.87%
14	97.17%	78.48%	94.82%
15	96.84%	78.33%	94.66%
16	97.70%	69.43%	94.17%
17	97.91%	78.52%	95.76%
18	99.32%	67.98%	97.07%
19	98.14%	76.46%	96.90%
20	96.27%	62.34%	93.10%
\bar{X}	96.61%	75.38%	94.47%

Tabulka 5.6: Výsledky segmentace provedené nad množinou STARE.

Výsledky jsou obsaženy v tabulce 5.6. Podle těchto výsledků navrhaná metoda překonala jednu ze srovnávacích metod, to lze vidět na tabulce 5.7, tento výsledek je ale do značné míry zkreslený tvorbou nepřesných masek. Příklad viditelně dobrého výstupu pro snímek množiny STARE je vidět na obrázku 5.2.

Metoda	SP	SN	ACC
Wang et al. 2017[33]	97.91%	81.04%	98.13%
Moghimirad et al. 2012[17]	99.1%	81.3%	97.6%
Annunziata et al. 2015[3]	98.4%	71.3%	95.6%
Odstrcilik et al. 2013[20]	95.12%	78.47%	93.41%
Navrhovaná metoda	96.61%	75.38%	94.45%

Tabulka 5.7: Nejvýkonnější techniky segmentace pro STARE množinu.

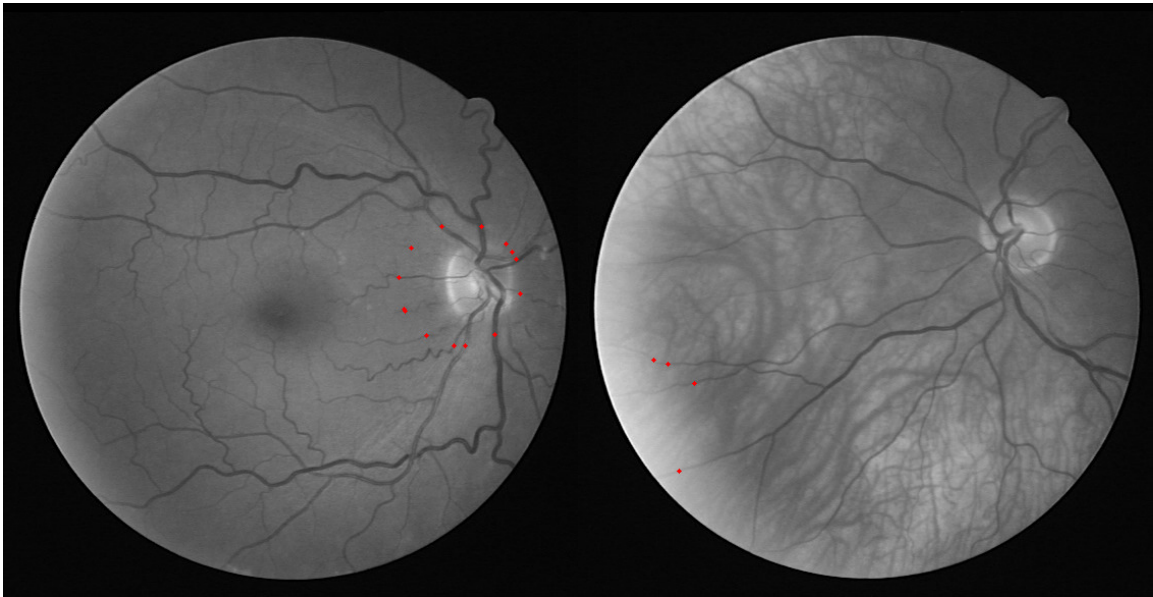


(a) Manuálně vytvořený výsledek pro snímek 11 z množiny STARE. (b) Výsledek procesu segmentace pro snímek 11 z množiny STARE.

Obrázek 5.2: Porovnání výsledku a manuálně získaného výsledku pro snímek číslo 11 z množiny STARE.

5.4 Výsledky lokalizace optického disku

Lokalizace optického disku se ukázala být dostatečně účinná. U devíti z deseti snímků obsažených v množině DRIVE byl optický disk úspěšně nalezen. U třetího snímku se optický disk nepodařilo nalézt, protože okraje snímku byly světlejší než střed optického disku. Na obrázku 5.3 je ukázáno porovnání vyhovujícího a nevhovujícího výsledku.



(a) Příklad správné lokalizace optického disku na snímku číslo 2. (b) Příklad chybné lokalizace optického disku na snímku číslo 3.

Obrázek 5.3: Příklad lokalizace optického disku a vyznačených startovacích bodů pro sledování.

5.5 Výsledky sledování

Sledování počátečních bodů a jejich rozdělení na menší cévní segmenty fungovalo poměrně uspokojivě. Některé segmenty se nepodařilo zmapovat do úplného konce, ale tyto nedokonalosti jsou pak odstraněny v posledním kroku. Rozdělení do malých segmentů je ostatně ukázáno na obrázku 4.8d. Výsledky obsahují všechny potenciaální začátky segmentů (dále jen jako **VPZS**), všechny začátky segmentů, které pokračují alespoň 3 pixely (dále jen **VZDS**) a jako poslední jsou začátku segmentů po provedení maskování míst s bifurkací a křížením (dále jen **KZS**). Počty všech sledovaných bodů jsou uvedeny v tabulkách 5.8 a 5.9.

Číslo obrázku	VPZS	VZDS	KZS
1	437	269	266
2	490	341	336
3	207	126	117
4	754	486	475
5	499	324	300
6	361	253	240
7	674	377	367
8	635	382	366
9	579	383	380
10	424	252	245

Tabulka 5.8: Počet nalezených kořenů cévních segmentů u množiny DRIVE.

Číslo obrázku	VPZS	VZDS	KZS
1	252	175	186
2	277	200	195
3	156	117	122
4	118	91	93
5	149	106	99
6	18	17	15
7	242	179	191
8	221	141	166
9	287	202	221
10	283	221	216
11	522	324	330
12	253	176	174
13	322	217	221
14	383	233	257
15	376	232	254
16	142	90	88
17	576	375	362
18	128	77	93
19	121	82	81
20	122	93	85

Tabulka 5.9: Počet nalezených kořenů cévních segmentů u množiny STARE.

Rychlost segmentace byla měřena na grafické kartě AMD RADEON 6700 XT. Naměřené hodnoty pak byly 0.86 sekundy na množině DRIVE, 1.14 sekundy na množině STARE a 24.69 na množině HRF. Měření zahrnuje jak proces předzpracování obrázku, tak proces segmentace.

5.6 Výsledky rozdělení na žíly a cévy

Pro vytvoření správných výsledků, které by zahrnovaly stejné metriky jako jsou uvedeny v sekci 5.1, by byly potřeba výsledné snímky. Bohužel v žádném článku či zdroji nebyla nalezena odpovídající data. Proto pro hodnocení budou uvedeny údaje vypovídající o procentuálním zastoupení pixelů reprezentujících žíly, tepny a nezmapované pixely. Měření nebude provedeno pro HRF, kvůli velké odlišnosti od DRIVE a STARE. Pro provedení měření nad množinou HRF by se musely upravit metriky pro nalezení optického disku a optimalizovat algoritmy.

Číslo obrázku	K-means metoda		ODR metoda		Nezmapované pixely
	Žíly	Tepny	Žíly	Tepny	
1	62.22%	36.07%	51.42%	47.91%	0.68%
2	58.10%	41.37%	26.50%	73.30%	0.20%
3	20.22%	69.26%	80.59%	19.41%	0.00%
4	41.61%	19.72%	55.53%	44.47%	0.00%
5	23.54%	25.08%	62.93%	37.07%	0.00%
6	7.89%	4.49%	53.18%	46.04%	0.78%
7	47.17%	52.62%	47.99%	51.80%	0.21%
8	35.03%	64.76%	43.35%	56.44%	0.21%
9	60.16%	39.03%	55.52%	43.93%	0.56%
10	77.49%	22.43%	56.49%	43.44%	0.07%

Tabulka 5.10: Procentuální zastoupení ve výsledcích rozřazování cévních segmentů u množiny DRIVE.

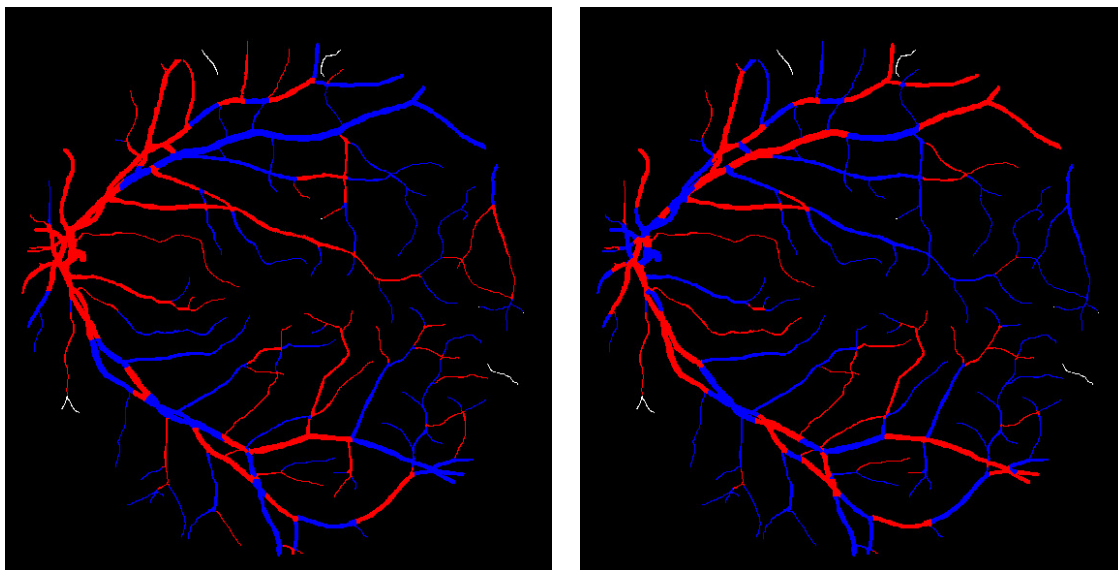
Číslo obrázku	K-means metoda		ODR metoda		Nezmapované pixely
	Žíly	Tepny	Žíly	Tepny	
1	47.63%	51.70%	63.21%	35.07%	1.71%
2	36.34%	63.46%	57.33%	42.15%	0.53%
3	48.20%	51.80%	43.22%	46.26%	10.53%
4	16.66%	83.34%	37.10%	24.24%	38.67%
5	37.38%	62.62%	21.58%	27.04%	51.38%
6	21.37%	77.85%	10.31%	2.07%	87.63%
7	71.00%	26.70%	36.67%	61.03%	2.30%
8	23.88%	54.74%	57.53%	21.09%	21.38%
9	60.16%	39.03%	76.78%	22.41%	0.81%
10	79.75%	19.17%	47.01%	51.91%	1.08%
11	39.78%	59.45%	57.42%	41.81%	0.77%
12	47.43%	52.57%	32.97%	67.03%	0.00%
13	31.79%	66.99%	53.77%	45.01%	1.21%
14	46.53%	53.47%	58.88%	41.12%	0.00%
15	27.81%	71.87%	76.63%	23.05%	0.32%
16	12.41%	26.49%	13.95%	24.95%	61.10%
17	11.74%	88.26%	64.70%	35.30%	0.00%
18	70.05%	27.08%	32.58%	64.54%	2.88%
19	63.01%	33.29%	67.56%	28.75%	3.69%
20	47.72%	39.09%	44.78%	42.03%	13.19%

Tabulka 5.11: Procentuální zastoupení ve výsledcích rozřazování cévních segmentů u množiny STARE.

V tabulkách 5.10 a 5.11 lze především vyzorovat fakt, že obě metody mají značně odlišné výsledky. Z posledního sloupce lze vyvodit, jak dobře fungovala lokalizace optického disku a na ní závislé mapování cévních segmentů. U výsledků z DRIVE množiny počet nezmapovaných pixelů nikdy nepřesáhl jedno procento. Oproti tomu pak u výsledků vygenerovaných nad množinou STARE zastoupení nezmapovaných pixelů v některých případech přesáhlo 50 %, což zbylé výsledky znehodnocuje.

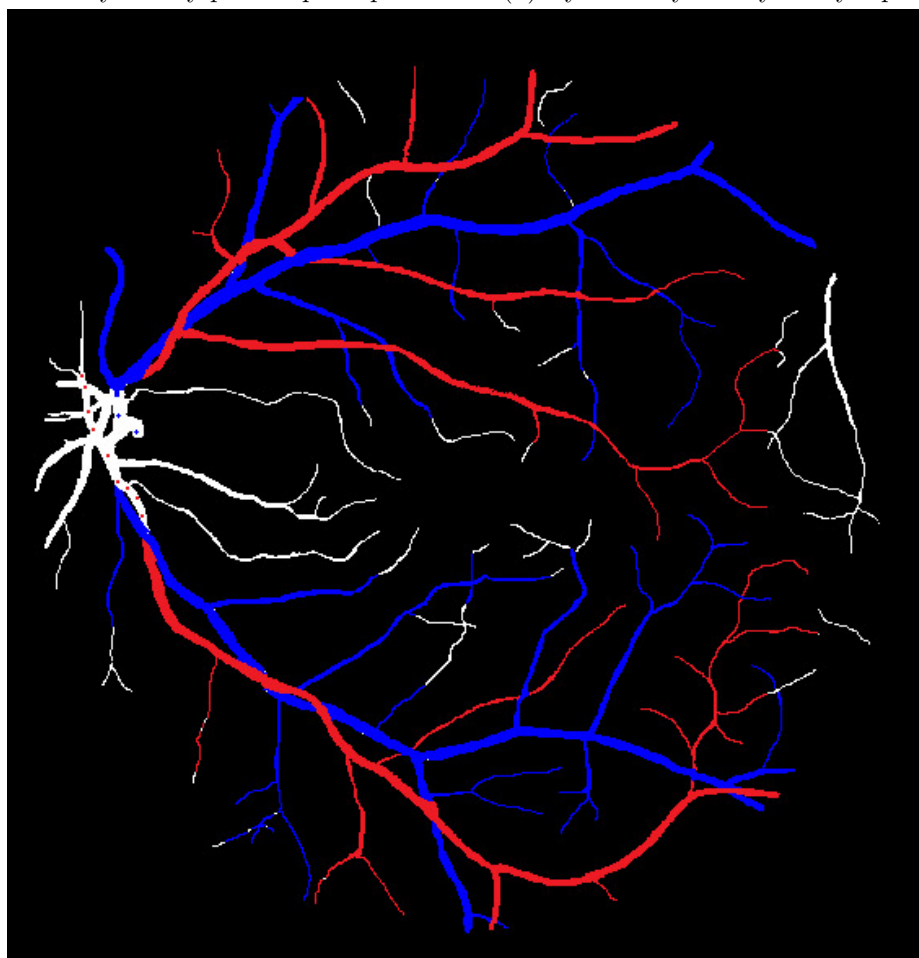
Tyto výsledky bohužel ukazují fakt, že ani jedna metoda nebyla zcela úspěšná při rozdělování cév na žíly a tepny. Pro potvrzení bohužel chybí data s rozdělenými cévami. Na ukázkou byl ručně vytvořen očekávaný výsledek u prvního snímku z množiny DRIVE, který je možno vidět na obrázku 5.4. Tento očekávaný výsledek není vytvořen profesionálem, byl vytvořen za použití znalostí získaných při studiu této problematiky a vyznačuje pouze cévy u kterých bylo jasné zařazení. První metoda ale vypadá jako ta výkonnější.

Drtivá většina výpočtů pro rozlišení žil a tepen byla provedena na procesoru AMD RYZEN 5 5600x. Používání procesoru a nevyužití hardwarové akcelerace zapříčinila velké časové zpoždění při posledním kroku. Pro množinu DRIVE pak výpočet všech segmentů přístupem **ODR** trval průměrně 3.01 sekund a přístupem **K-means** 2.81 sekund. Množina STARE pak pro přístup **ODR** dosahovala času 3.14 sekund a přístup **K-means** 3.13 sekund. Nejvíce času pak zabere proces dokončení **Dokončení** (viz. 4.8), který průměrně zabere 27.25 sekund na množině DRIVE a 30.8 sekund na množině STARE.



(a) Výsledek vytvořený prvním přístupem.

(b) Výsledek vytvořený druhým přístupem.



(c) Manuálně vytvořený výsledek.

Obrázek 5.4: Srovnání výsledků obou metod a očekávaného výsledku.

Kapitola 6

Závěr

Cílem této práce bylo nastudovat si problematiku segmentace cév v sítnicových snímcích a jejich rozdělení na žíly a tepny, navrhnout postup řešení, implementace řešení a následné vyhodnocení výsledků postupu. Úvod do problematiky je popsán v první kapitole. Druhá kapitola shrnuje známé postupy a řešení. Následuje popis navrhované metody pro segmentaci cév ze sítnicových snímků a jejich rozdělení na žíly a tepny. Zbytek práce pak popisuje implementaci navrhovaného postupu a shrnutí výsledků.

Tato práce je unikátní v použité kombinaci speciálně navržených filtrů, které jsou založeny na křivce normálního rozdělení a její druhé derivaci. Nejmarkantnější jsou výsledky segmentace cévního řečiště na databázi DRIVE, kde se podařilo dosáhnout průměrného výsledku 93.2 %. Za zmínku také stojí přesnost 94.47 %, která byla naměřena na databázi STARE, zde ale výsledek není zcela průkazný.

Rozdělení cév pak nepřineslo očekávané výsledky. Byly použity dva přístupy, ze kterých přístup založený na výpočtu takzvaného ODR, přinesl částečně lepší výsledky. Práce by mohla být dále rozšířena o automatické sestavování filtrů na základě zadaného rozptylu cév.

Další vylepšení by mohlo přinést sestavení orientovaného grafu jednotlivých segmentů a následné přepočítání segmentu na základě jeho umístění v grafu.

Literatura

- [1] M. Al-Rawi, M. Qutaishat, and M. Arrar. An improved matched filter for blood vessel detection of digital retinal images. *Computers in Biology and Medicine*, 37(2):262–267, 2007.
- [2] M. Alam, T. Son, D. Toslak, J. I. Lim, and X. Yao. Combining odr and blood vessel tracking for artery–vein classification and analysis in color fundus images. *Translational Vision Science amp; Technology*, 7(2):23, 2018.
- [3] R. Annunziata, A. Garzelli, L. Ballerini, A. Mecocci, and E. Trucco. Leveraging multiscale hessian-based enhancement with a novel exudate inpainting technique for retinal vessel segmentation. *IEEE Journal of Biomedical and Health Informatics*, 20:1129–1138, 07 2016.
- [4] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on Medical Imaging*, 8(3):263–269, 1989.
- [5] O. Chutatape, L. Zheng, and S. Krishnan. Retinal blood vessel detection and tracking by matched gaussian and kalman filters. *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286)*, 1998.
- [6] Drive - retinal images database. <https://drive.grand-challenge.org/>.
- [7] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. *Medical Image Computing and Computer-Assisted Intervention — MICCAI’98*, page 130–137, 1998.
- [8] M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. Rudnicka, C. Owen, and S. Barman. Blood vessel segmentation methodologies in retinal images – a survey. *Computer Methods and Programs in Biomedicine*, 108(1):407–433, 2012.
- [9] F. Girard and F. Cheriet. Artery/vein classification in fundus images using cnn and likelihood score propagation. pages 720–724, 2017.
- [10] G. Hassan, N. El-Bendary, A. E. Hassanien, A. Fahmy, S. A. M., and V. Snasel. Retinal blood vessel segmentation approach based on mathematical morphology. *Procedia Computer Science*, 65:612–622, 2015.
- [11] R. Hemelings, B. Elen, I. Stalmans, K. Van Keer, P. De Boever, and M. B. Blaschko. Artery–vein segmentation in fundus images using a fully convolutional network. *Computerized Medical Imaging and Graphics*, 76:101636, 2019.

- [12] V. S. Joshi, J. M. Reinhardt, M. K. Garvin, and M. D. Abramoff. Automated method for identification and artery-venous classification of vessel trees in retinal vessel networks. *PLoS ONE*, 9(2), 2014.
- [13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [14] B. Ladislav. *Soustavná anatomie člověka*. Státní zdravotnické nakladatelství, 1955.
- [15] F.-A.-U. E.-N. Lehrstuhl für Mustererkennung. High-resolution fundus. <https://www5.cs.fau.de/research/data/fundus-images/>.
- [16] Lidské oko - wikipedie. https://cs.wikipedia.org/wiki/Lidské_oko#/media/Soubor:Schematic_diagram_of_the_human_eye_cs.svg.
- [17] E. Moghimirad, S. H. Rezatofghi, and H. Soltanian-Zadeh. Retinal vessel segmentation using a multi-scale medialness function. *Computers in Biology and Medicine*, 42(1):50–60, 2012.
- [18] R. Nekovei and Y. Sun. Back-propagation network and its configuration for blood vessel detection in angiograms. *IEEE Transactions on Neural Networks*, 6(1):64–72, 1995.
- [19] Numpy. <https://numpy.org/>.
- [20] J. Odstrcilik, R. Kolar, A. Budai, J. Hornegger, J. Jan, J. Gazarek, T. Kubena, P. Cernosek, O. Svoboda, E. Angelopoulou, and et al. Retinal vessel segmentation by improved matched filtering: evaluation on a new high-resolution fundus image database. *IET Image Processing*, 7(4):373–383, 2013.
- [21] Opencv. <https://opencv.org>, Feb 2022.
- [22] L. E. M. C. M. O. M. Penedo. A snake for retinal vessel segmentation. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, 2007.
- [23] PyImageSearch. Pyimagesearch/imutils: A series of convenience functions to make basic image processing operations such as translation, rotation, resizing, skeletonization, and displaying matplotlib images easier with opencv and python.
- [24] P. M. O. Riebel. *Vybrané kapitoly z oftamologie*. Státní pedagogické nakladatelství Praha, 1985.
- [25] S. A. Salem, N. M. Salem, and A. K. Nandi. Segmentation of retinal blood vessels using a novel clustering algorithm (racal) with a partial supervision strategy. *Medical and Biological Engineering and Computing*, 45(3):261–273, 2007.
- [26] C. Sinthanayothin, J. F. Boyce, H. L. Cook, and T. H. Williamson. Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images. *British Journal of Ophthalmology*, 83(8):902–910, 1999.
- [27] Skimage. <https://scikit-image.org/>.
- [28] Structured analysis of the retina. <https://cecas.clemson.edu/~ahoover/stare/>.

- [29] K. W. Sum and P. Y. S. Cheung. Vessel extraction under non-uniform illumination: A level set approach. *IEEE Transactions on Biomedical Engineering*, 55(1):358–360, 2008.
- [30] B. M. ter Haar nbsp;Romeny, E. J. Bekkers, J. Zhang, S. Abbasi-Sureshjani, F. Huang, R. Duits, B. Dashtbozorg, T. T. Berendschot, I. Smit-Ockeloen, K. A. Eppenhof, and et al. Brain-inspired algorithms for retinal image analysis. *Machine Vision and Applications*, 27(8):1117–1135, 2016.
- [31] Y. Toliaş and S. Panas. A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering. *IEEE Transactions on Medical Imaging*, 17(2):263–273, 1998.
- [32] H. Václav and S. Miloř. *Zpracování signálů a obrazů*. České vysoké učení technické, 2009.
- [33] S. Wang, Y. Yin, G. Cao, B. Wei, Y. Zheng, and G. Yang. Corrigendum to “hierarchical retinal blood vessel segmentation based on feature and ensemble learning” [neurocomputing 149 (2015) 708–717]. *Neurocomputing*, 226:270–272, 2017.