VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

SÉMANTICKÁ SEGMENTACE PATOLOGIÍ V OBRAZECH SÍTNICE

SEMANTIC SEGMENTATION OF PATHOLOGIES IN RETINAL IMAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. ROMAN ČABALA**
AUTHOR

VEDOUCÍ PRÁCE **MSc. ANDRII KAVETSKYI**
SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání diplomové práce

147240

Ústav:	Ústav inteligentních systémů (UITS)
Student:	**Čabala Roman, Bc.**
Program:	Informační technologie a umělá inteligence
Specializace:	Kybernetická bezpečnost
Název:	**Sémantická segmentace patologií v obrazech sítnice**
Kategorie:	Umělá inteligence
Akademický rok:	2022/23

Zadání:

1. Prostudujte literaturu týkající se sítnice lidského oka, patologií, které se na ní vyskytují a prozkoumejte stávající přístupy používané pro segmentaci patologií.
2. Navrhněte si vlastní algoritmus pro sémantickou segmentaci vybraných patologií v obrazech sítnice včetně drúz, tvrdých a měkkých exsudátů, kde každý pixel obrazu sítnice bude odpovídajícím způsobem klasifikován jako zdravý nebo obsahující vybranou patologii.
3. Implementujte a otestujte algoritmus z předchozího bodu na dostupných databázích.
4. Shrňte dosažené výsledky, tyto diskutujte a navrhněte možné rozšíření práce.

Literatura:

- R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng and A. Nandi. Medical Image Segmentation Using Deep Learning: A survey. IET Image Processing, 2022.
- S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, Ph. Torr and L. Zhang. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. Computer Vision and Pattern Recognition, 2021.
- R. Imtiaz, T. Khan, S. Naqvi, M. Arsalan, & J. N. Syed. Screening of Glaucoma Disease from Retinal Vessel Images Using Semantic Segmentation. Computers and Electrical Engineering, 2021.
- M. Badar, M. Shahzad and M. M. Fraz. Simultaneous Segmentation of Multiple Retinal Pathologies Using Fully Convolutional Deep Neural Network. Medical Image Understanding and Analysis, 2018.

Při obhajobě semestrální části projektu 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:	1.11.2022
Termín pro odevzdání:	17.5.2023
Datum schválení:	3.11.2022

Abstrakt

Cieľom diplomovej práce bolo segmentovať patológiu viditeľnú na snímkach sietnice, ako sú exsudáty, hemoragia a mikroaneuryzmy. Za týmto účelom boli vyskúšané dve dobre známe hlboké neurónové siete, konkrétne U-Net a SegFormer. Na testovanie výkonnosti modelov sa použil jeden verejne dostupný dataset IDRiD. Získané výsledky boli opísané po analýze rôznych faktorov, ktoré ovplyvnili výkon modelov U-Net a Segformer.

Abstract

The thesis aimed to segment pathology visible in the retina images, such as exudates, hemorrhages, and microaneurysms. For that, two well known deep neural networks, named U-Net and SegFormer, were trained. To test the performance of the models, one publicly available dataset was used, named IDRiD. Obtained results were reported after analyzing different factors which affected the performance of the models U-Net and Segformer.

Kľúčové slová

sémantická segmentácia, sietnica, drúzy, exudáty, hemoragia, mikroaneurizmy, segmentačné transformátory, IDRiD, U-Net, SETR, SegFormer

Keywords

semantic segmentation, retina, drusen, exudates, hemorrhages, microaneurysms, segmentation transformers, IDRiD, U-Net, SETR, SegFormer

Citácia

ČABALA, Roman. *Sémantická segmentace patologií v obrazech sítnice.* Brno, 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce MSc. Andrii Kavetskyi

Sémantická segmentace patologií v obrazech sítnice

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána MSc. Andrii Kavetskeho. Ďalšie informácie mi poskytla doc. Sangeeta Biswas. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Roman Čabala
16. mája 2023

Poďakovanie

Rád by som poďakoval svojmu vedúcemu práce MSc. Andrii Kavetskemu s spolu s ním výskumníčke doc. Sangeeta Biswas za ich odborné rady, neustálu pomoc a pozitívny prístup pri riešení práce. Rovnako by som rád poďakoval svojej rodine, priateľke a kamarátom, ktorý ma podporovali počas celého štúdia.

Obsah

Zoznam obrázkov

Kapitola 1

Úvod

Ľudské oko je najdôležitejší zmyslový orgán. Pomocou oka vnímame najväčšiu časť informácií zo svojho okolia. Oko sa skladá z mnoho častí, ale najdôležitejšou časťou je sietnica. Pridružené choroby alebo poškodenie oka či sietnice, môže viesť k zhoršeniu zraku v najhoršom prípade k strate zraku. Preto si treba dávať pozor a pacienti, ktorý trpia chorobou súvisiacou so zrakom by mali pravidelne chodiť na očné vyšetrenia. Táto práca sa venuje možnému uľahčeniu vyšetrení sietnice oka. Kedy by doktor mohol mať technológiu, ktorá automaticky detekuje patologické nálezy na sietnic. Tým by sa uľahčilo a zrýchlilo možné vyšetrenie. Ďalším využitím by mohol byť autonómny systém na detekciu patológií. Pacientov, ktorý by mali vysokú pravdepodobnosť niektorej z patológií by systém presmeroval k lekárovi. Autonómna predkontrola umožňuje riešiť vzrastajúcu potrebu pre oftalmológiu.

Medzi základné patologické nálezy na oku patria: mikroaneurizmy, drúzy, exudáty a hemoragia. Táto práca sa snaží sémanticky segmentovať patologické nálezy, za pomoci neurónových sietí. Na začiatku sa použije sieť U-Net. Rozpoznajú sa patológie jedna po jednej. Následne sa všetky patológie spoja a vyskúša sa multiclass sémantická segmentácia. Po natrénovaní neurónovej siete za pomoci architektúry U-Net vyskúšam aj natrénovať ďalšiu architektúru SegFormer, ktorá používa transformátory. Je to rozdielny pohľad na sémantickú segmentáciu, kde táto architektúra vychádza z architektúr, ktoré riešia prirodzený jazyk.

Práca je členená do šiestich kapitol. Kapitola 2 vysvetľuje základnú stavbu oka a sietnice. Následne sú vysvetlené patologické nálezy na sietnici a priebeh vyšetrenia očí. Ďalšia kapitola 3 sa venuje neurónovým sieťam pre sémantickú segmentáciu. V počiatku kapitole je popísaný úvod do sémantickej segmentácií a jej rozdiel. V tejto kapitole sa nachádza prehľad vybraných architektúr neurónových sietí, ktoré riešia sémantickú segmentáciu. Kapitola 4 ukazuje stručný návrh riešenia tejto práce. Spolu s návrhom je popísaný dataset s ktorým som pracoval. V kapitole 5 je popísaná implementácia a postup pri riešení práce. Kapitola poukazuje na problémy, riešenia a nezdary, ktoré nastali pri riešení tejto práce.

Kapitola 2

Ľudské oko

Ľudské oko je najdôležitejší zmyslový orgán človeka. Okom človek vníma najväčšiu časť informácií z okolitého prostredia. Oko je najrýchlejší sval tela, má približne guľovitý tvar s priemerom 24 *mm* a hmotnosťou 7 *g*. Samotné oko videnie nezaistí. Oko je prepojené so zrakovým centrom mozgu a vďaka tomu človek je schopný vidieť svet. Všetky časti oka, cez ktoré lúč prechádza sú priehľadné. Je to z toho dôvodu aby sa zabraňovalo rozptylu dopadajúceho svetla.

2.1 Stavba oka

Ľudské oko sa skladá z mnoho častí. Obrázok 2.1 zobrazuje stavbu ľudského oka [31].

- **Očné bielko** *(sclera)* je nepriehľadná, tuhá, biela vrstva, ktorá obaľuje očnú buľvu. Skladá sa z troch častí: episkléra, sklerálna stróma a lamina fusca. Ochraňuje vnútorné časti očnej buľvy. Farba bielka sa vekom mení. Deti môžu mať bielko ľahko priehľadné a preto cez neho môže presvitať pigment. V dospelosti ma skléra bielu farbu a v staršom veku môže byť bielko žltasté v dôsledku ukladania tuku. Vpredu je bielko tenšie asi 0,4 mm a vzadu hrubšie asi 1 mm. V prednej časti bielka prechádza skléra v rohovku.

- **Očná rohovka** *(cornea)* je transparentné číre tkanivo bez ciev. Rohovka je jediným vstupom pre svetlo. Na dioptrickom účinku oka sa rohovka podieľa vyše dvojtretinovým podielom. Tento účinok je na základe zakrivenia plôch rohovky a rozdielnym indexom lomu prostredia pred okom.

- **Očná dúhovka** *(iris)* je kruhovitý útvar v strede s otvorom (zrenice), ktorá plní úlohu optickej clony. Reguluje prístup svetla do oka. Dúhovku tvorí predovšetkým väzivová stróma a dva hladké svaly. Svaly sa starajú o veľkosť zrenice a reflexne regulujú množstvo svetla, ktoré prechádza okom. V svetlých podmienkach je zrenica menšia a prepúšťa menej svetla. V tmavších podmienkach nastáva opak zrenica je väčšia a prepúšťa viacej svetla. Vďaka dúhovky určujeme u ľudí aj farbu očí. Farba dúhovky a teda aj očí závisí od množstva a hĺbky uloženia pigmentu.

- **Očná šošovka** *(lens)* leží za dúhovkou v sklovci a je upevnená vláknami zaveseného aparátu, ktorý sa upína medzi výbežkami ciliárneho telieska. Šošovka má priehľadný diskovitý tvar. Je približne 4 mm hrubá. Hlavnou funkciou šošovky je ohýbať lúče

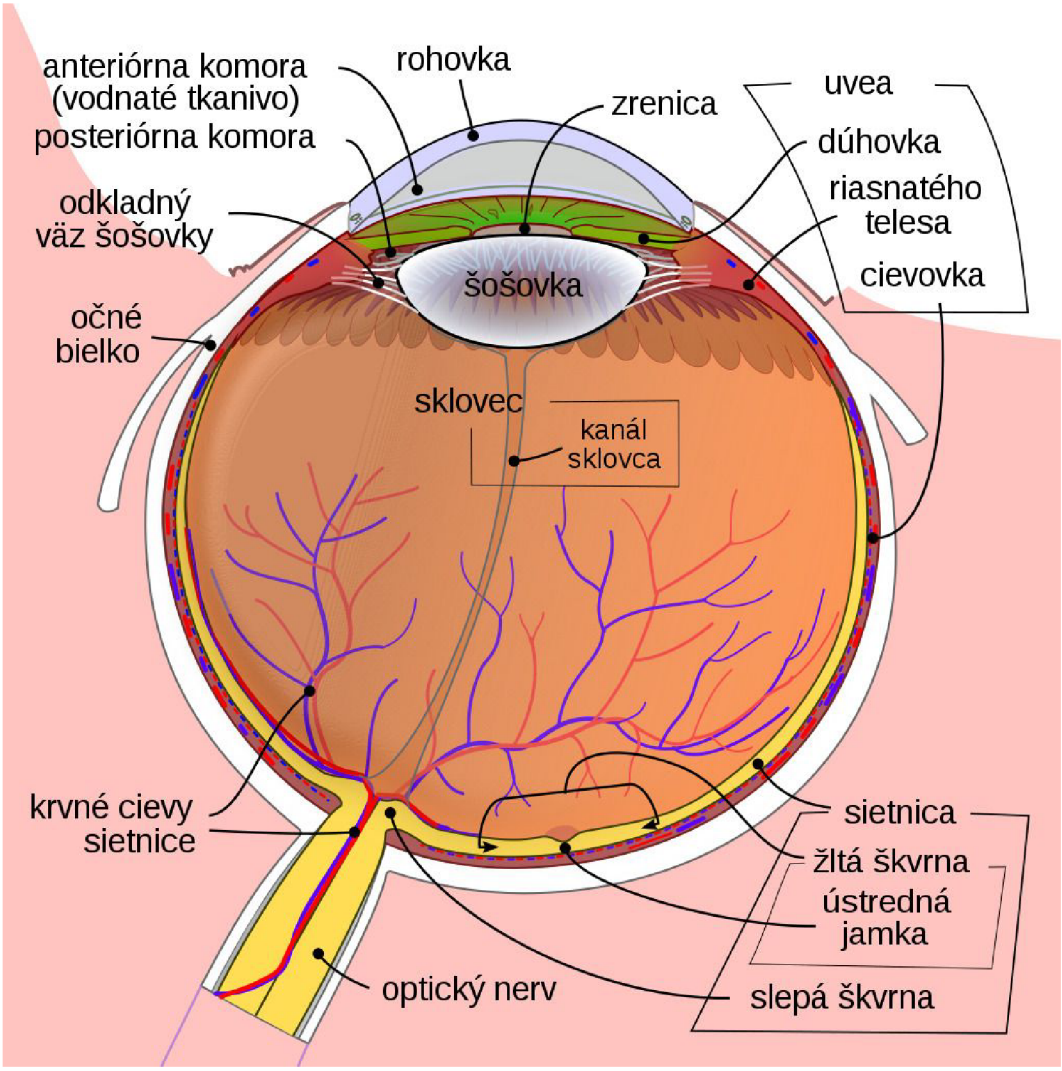
svetla, tak aby sa zbiehali na sietnici. Je schopná meniť svoju optickú mohutnosť. Priehľadnosť šošovky klesá s vekom a mení sa aj jej tvar.

- **Sklovec** *(corpus vitreum)* tvorí najväčšiu časť oka. Je to priehľadná rôsolovita hmota, ktorá sa skladá s prepletenými vláknami s tekutinou, ktorá je bohatá na bielkoviny. Až 98% stavby sklovca je voda, ktorá ma objem približne 4 ml. Nenachádzajú v ňom žiadne cievy ani nervy. Hlavnou úlohou sklovca je udržiavať tvar oka. Vďaka sklovcu je oko pevné a pružné. Poškodený sklovec vedie k poruchám sietnice a k problémom s videním.

- **Cievnatka** *(choroidea)* má sfarbenie do hneda, kvôli vysokému obsahu pigmentu. Vysoký počet pigmentu vytvára čiernu komoru, ktorá pohlcuje svetelné lúče. Nachádza sa medzi sietnicou a bielkom a vypĺňa skoro celý rozsah očnej buľvy. Tenká vrstva väziva oddeľuje cievnatku od bielka. Cievnatka obsahuje veľké množstvo ciev a podľa toho je aj odvodený názov.

- **Riasnaté teleso** *(corpus ciliare)* sa nachádza na vnútornej strane bielka. Je priamym pokračovaním cievnatky a je to usporiadaný sval z hladkej svaloviny. Po priamom reze má tvar trojuholníka. Vpredu je riesnaté teleso hrubšie a je spojené s dúhovkou. Zadný okraj je tenší ako predný a prechádza do cievnatky. Svalová časť riasnatého telesa pomáha šošovke zaostrovať objekty.

- **Sietnica** *(retina)* vystiela vnútornú stranu očnej buľvy. Sietnica je podrobnejšie popísaná v následujúcej podkapitole, lebo je to najpodstatnejšia časť pre túto prácu.



Obr. 2.1: Stavba ľudského oka [38]

2.2 Sietnica

Sietnici sa venuje táto práca. Je to najdôležitejšia a najcitlivejšia časť ľudského oka. Nachádza sa na vnútornej časti očnej buľvy a je to tenká priehľadná vrstva. Jej hrubosť nie je všade rovnaká. V zadnej časti je hrubá asi 0,5 mm a v prednej časti približne 0,1 mm. Vonkajšia plocha susedí so sklovcom a vnútorná s cievnatkou. Považuje sa za jedinú časť centrálneho neurónového systému, ktorú je možné pozorovať bez invazivného prístupu. V sietnici sa nachádzajú receptory, ktoré sú schopné reagovať na svetelné paprsky. Je to rovnako jediná časť ľudského oka, ktorá reaguje na svetelné podnety.

Na sietnici sa nachádzajú dve časti, ktoré sú veľmi výrazne a vieme ich rozlíšiť. Prvou je časťou je optický disk (slepá škvrna) a druhou časťou je makula (žltá škvrna). Rovnako na sietnici sa nachádzajú tyčinky a čapíky. Obrázok 2.2 zobrazuje sietnicu ľudského oka.

Tyčinky a čapíky

Tyčinky sa nachádzajú na celej sietnice, okrem žltej škvrne. Dokopy sa nachádza približne 130 miliónov tyčiniek na sietnici. Funkciou tyčiniek je prijímať svetelný prijímač, vďaka ktorému je oko schopné vytvoriť obraz. Sú veľmi citlivé na dopadajúce svetlo citlivejšie ako čapíky. Vďaka tejto citlivosti tyčinky slúžia na videnie za šera a v noci. Tyčinky však nedokážu rozpoznať farbu vidia iba čiernobielo [43].

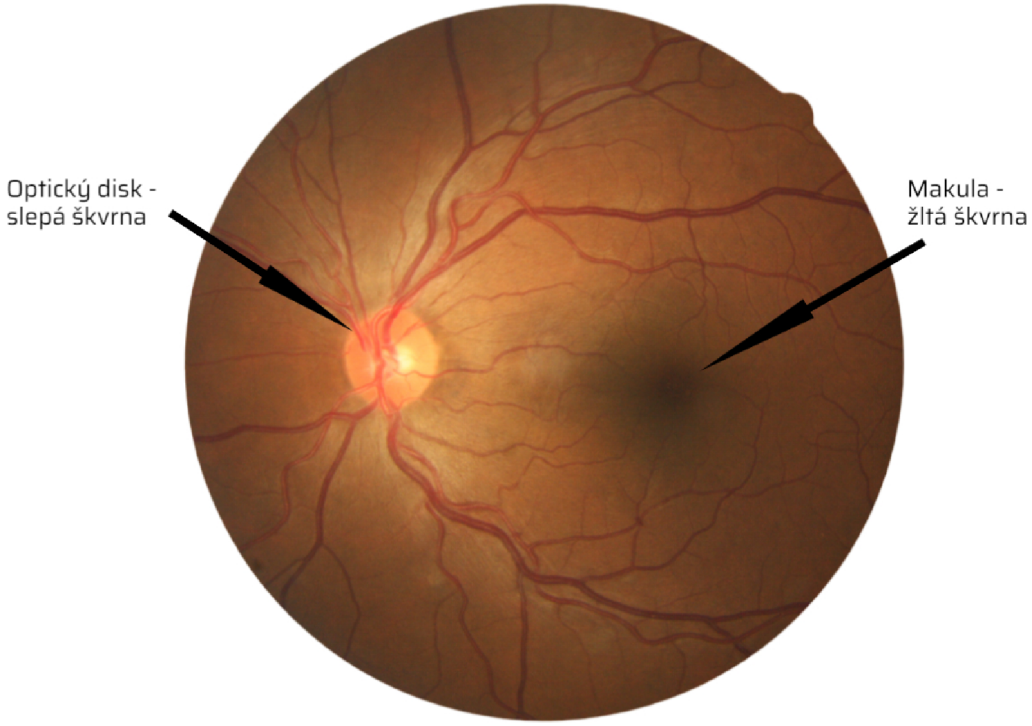
Čapíky majú podobnú stavbu ako tyčinky. Rozdiel medzi tyčinkami a čapíkmi je v tvare. Čapíky sú kratšie, silnejšie ako tyčinky a majú kužeľovitý tvar. Na sietnici oka sa nachádza asi 7 miliónov čapíkov a nachádzajú sa hlavne v žltej škvrne. Čapíky zaisťujú lepšiu ostrosť ako tyčinky. U človeka existujú tri druhy čapíkov S, M a L. Každý druh je inakšie citlivý a inak reagujú na vlnové dĺžky viditeľného spektra pre farby červená, zelená a modrá. Vzniká diferenciálna citlivosť k farbám. Po kombináciách farieb vzniká výsledný obraz [43].

Optický disk (slepá škvrna)

Optický disk je miestom kde sa zbiehajú nervové vlákna a vystupuje z neho zrakový nerv. Má okrúhly tvar s priemerom približne 1,5 mm. Na obrázku sietnice je optický disk viditeľný ako svetlý oblúk a je to svetlejšia časť sietnice. V optickom disku sa nenachádzajú žiadne fotoreceptory. To znamená, že ak svetelný paprsok dopadne do optického disku, človek predmet nevníma a nevidí ho. Človek si to neuvedomuje, lebo mozog dopĺňa obraz a z toho dôvodu nevníma medzeru v zornom poli, ktorá vzniká paprskom dopadajúcim na optický disk. Z toho dôvodu sa nazýva optický disk aj slepou škvrnou. Optický disk je umiestnený 3-4 mm od makuli [43].

Makula (žltá škvrna)

Makula je miestom najostrejšieho videnia o priemere do 5 mm. Skladá za z miliónov čapíkov, okrem čapíkov sa tu nenachádzajú žiadne iné bunky. V strede makuli sa nachádza priehlbinka, ktorej názov je fovea (*fovea centralis*). Vďaka vlastnostiam fovei, že je to mierna priehlbinka umožňuje obsahovať viac čapíkov. Makula spolu s foveo na sietnici vyzerajú ako tmavá časť sietnice. Jej názov žltá škvrna vznikol preto, lebo makula sa zafarbí na žlto po smrti človeka [43].



Obr. 2.2: Sietnica ľudského oka

2.3 Patológia na sietnici

Medzi základné patologické nálezy na sietnici oka patria mikroaenurizmy, retinálne hemogarie, drúzy a exudáty.

Mikroaneurizmy

Mikroaneurizmy je patologický nález na sietnici, ktorý väčšinou ako prvý sa nájde u pacientov. Podobu majú malých čiernych bodiek a majú ohraničené okraje .Nachádzajú v strednej vrstve sietnici a vznikajú dôsledkom odumierania buniek (pericytov), tým sa oslabuje kapilárna stena. Veľkosť je v rozmedzí 12 - 100 µm. Malý počet mikroanerizmov samy od seba nespôsobujú vážne postihnutie zraku. Z ich pribúdaním vznikajú mikrovaskulárne zmeny. Sú veľmi podobné hemoragií. Rozlíšiť sa dajú pomocou fluoresenčnej angiografií. Pri tomto vyšetrení mikroaneuryzmy jasne svietia oproti hemoragií [15].

Retinálne hemoragie

Retinálne hemoragie alebo krvácanie majú vzhľad tmavo červených bodiek. Rozdeľujeme ich na bodkované, plamienkové a škvrnité. Bodkované sú malé tmavo červené bodky a nachádzajú sa v stredných vrstvách sietnice. Práve bodkované hemoragie sa podobajú mikroaenurizmom a nachádzajú sa v okolí makuly. Plamienkové hemoragie sú väčšie ako bodkované a nachádzajú vo vrstve nervových vlákien. Škvrnité hemoragie sú charakteristické svojím tvarom, často vytvárajú zhluky. Nachádzajú sa v strednej vrstve sietnice [15].

Drúzy

Drúzové telieska sú extracelulárne usadeniny lipidov, proteínov a bunkových zvyškov. Sú charakteristické, ale nie jedinečne spojené so starnutím a podmienenou degeneráciou makuly. Drúzy sú výsledkom zníženej schopnosti sietnice čistiť odpadové produkty fotoreceptorov a môžu sa objaviť po celej sietnice. Drúzy delíme na tvrdé a mäkké.

- Tvrdé drúzy sú to ohraničené ložiská uložené na hranici pigmentového epitelu a Bruchovej membrány. Sú dožlta sfarbené a sú menšie oproti mäkkým drúz. Ich veľkosť je pod 63 µm. Mäkké drúzy je lepšie rozoznať oproti mäkkých drúz vďaka tomu, že majú výrazné okraje. Výskyt tvrdých drúz u ľudí, ktorý trpia vekom podmienenou degeneráciou makuly je veľmi bežná. Môžu sa však vyskytovať aj u mladých ľudí, ktorý nemajú ešte degeneráciu makuly. Početným výskytom tvrdých drúz môžu vzniknúť mäkké drúzy [13].

- Mäkké drúzy sú väčšie, blízko pri sebe a splývajú do seba spolu s pozadím sietnice Splývajú na základe toho, že nemajú jasne viditeľný okraj. Sú svetložltej alebo sivobielej farby a majú kupolovitý tvar. Veľkosť mäkkých drúz je väčšia ako 63 µm [13].

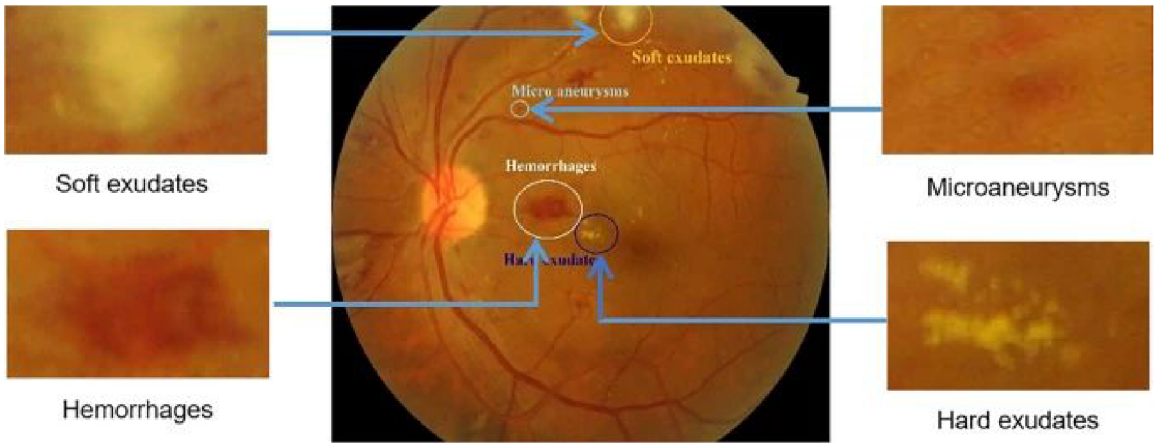
Exudáty

Exudáty sú spôsobené presakovaním tukových usadenín z krvných ciev a objavujú sa v kompaktných skupinách. Exudáty sa delia na mäkké a tvrdé.

- Tvrdé exudáty sú ložiskami bielkovín a lipidov. Ich vznik je podmienený presakovaním mikroaneuryziem a krvnej plazmy z kapilár. Bodové tvrdé exudáty rozpoznáme ako žlté bodky, ktoré sa zhlukujú a môžu sa premiestnovať v rámci sietnice. Existujú aj chumáčovité exudáty, ktoré majú hrudkovitú formu a uložené sú na povrchu. Tvrdé exudáty môžu vzniknúť v oblasti fovei. Takýto výskyt exudátu vo fovei môže poškodiť sietnicu a vedie k strate zraku. Počas ochorenia exudáty môžu meniť svoju veľkosť ale tendencia je, že sa prevažne zväčšujú ako zmenšujú [15].

- Mäkké exudáty alebo aj vatové ložiská, sú chumáčiky vaty sfarbené do bielo-žltej farby. Občas sú ohraničené mikroaneurizmami, ale väčšinou sú bez ohraničenia. Mäkké exudáty môžu vzniknúť infarktami u človeka. Nachádzajú vo vrstve nervových vlákien. V dôsledku infarktov vzniká porušenie nervových vlákien a vylieva sa axoplazma. Existenciu mäkkých exudátov spôsobuje vážnejšie ochorenie, ako napríklad diabetologická retinopatia. Nespôsobujú stratu zraku ani žiadne vážnejšie postihnutie zraku. Pacient môže vidieť škvrny v zornom poli [15].

Drúzy aj exudáty sú si veľmi podobné. Na obrázkoch sietnice oka vyzerajú ako totožné. Oftalmológ, ale vie rozlíšiť či o ktorý patologický nález ide a vie tomu prispôsobiť liečbu pacienta [2]. Obrázok 2.3 zobrazuje sietnicu oka s viacerými patológiami.



Obr. 2.3: Sietnica ľudského oka s patológiami [29]

2.4 Vyšetrenie očného pozadia

Medzi základné oftalmologické vyšetrenia patrí vyšetrenie očného pozadia. Lekár pri tomto vyšetrení vie skontrolovať zadnú časť oka pri ktorom skontroluje stav sietnice, cievy, zrakový nerv a podobne. Na pravidelne vyšetrenie by mali chodiť ľudia, ktorý trpia napríklad cukrovkou alebo vysokým krvným tlakom. Lekár vie pravidelnými vyšetrením kontrolovať stav oka, či sa patologický nález zhoršuje alebo zhoršuje. Na vyšetrenie očného pozadia existuje mnoho prístrojov. Najčastejšie sa používa oftalmoskop, štrbinová lampa alebo fundus kamera [11] [14].

2.4.1 Oftalmoskopia

Pri oftalmoskopie ide o postup, pri ktorom sa pomocou oftalmoskopu, teda zariadenia na osvetľovanie a pozorovanie oka, vyšetruje stav oka vrátane jeho pozadia, teda očného dna. Vyšetrením oftalmoskopom sa dá skontrolovať napríklad stav očnej sietnice, cievy a nervy na oku, ale aj hĺbka oka. Toto vyšetrenie by malo byť vykonávané pravidelne v rámci preventívnych prehliadok, najmä u ľudí s vysokým rizikom vzniku očných ochorení. Vyšetrenie oftalmoskopom je bezbolestné a trvá zvyčajne len niekoľko minút [11] [14] [20]. Oftalmoskopiu vieme rozdeliť na priamu a nepriamu:

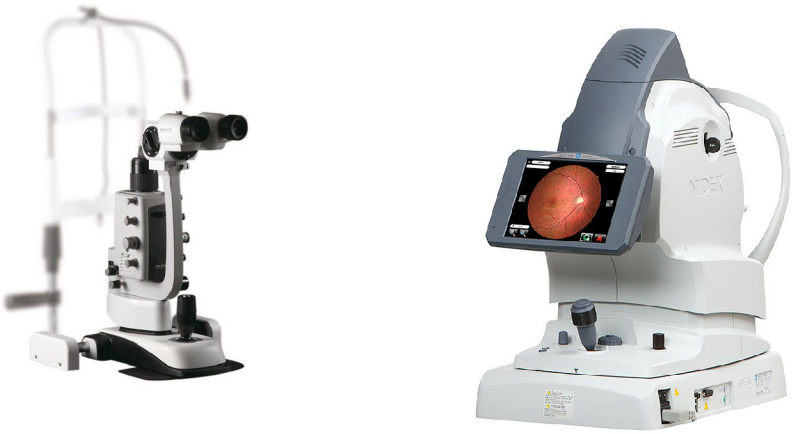
- **Priama oftalmoskopia** je metóda, ktorá sa vykonáva pomocou oftalmoskopu umiestneného tesne pred okom pacienta. Svetlo z oftalmoskopu prechádza cez šošovku na povrchu oka a umožňuje lekárovi vidieť vnútorné štruktúry oka. Táto metóda sa používa najmä pri vyšetrení prednej časti oka, ako sú rohovka, šošovka a cievky na očnej dne. Výsledný obraz po vyšetrení je priamy, a približne 16-krát zväčšený. Práve takéto zväčšenie je výhodou a umožňuje detailnejšie vyšetrenie sietnice oka. K tejto výhode patrí aj nevýhoda v tom, že pri takomto zväčšený je vidieť iba malú plochú sietnice [14].
- **Nepriama oftalmoskopia** je metóda, ktorá sa vykonáva pomocou špeciálneho oftalmoskopu, ktorý sa umiestňuje na väčšiu vzdialenosť od oka pacienta. Tento oftalmoskop produkuje veľké množstvo svetla, ktoré vstupuje do oka a umožňuje lekárovi vidieť očné dno a vnútorné štruktúry oka v širšom poli. Táto metóda sa používa

najmä pri vyšetrení zadnej časti oka, ako sú sietnica, nervové dráhy a krvné cievy. Zdroj svetla je umiestnený na čelenke. Výsledný obraz je reálny a prevrátený. Výsledný obraz nie je tak zväčšený ako pri priamej oftalmoskopií. Výhodou však je zobrazená väčšia časť sietnice a pre lekára je vyšetrenie sietnice priehľadnejšie [11] [14].

Obr. 2.4: Vľavo priamy oftalmoskop, vpravo nepriamy oftalmoskop

- **Štrbinová lampa** je zdroj svetla používaný pri vyšetrení pozadia oka. Štrbinová lampa je umiestnená v blízkosti pozorovacej časti oftalmoskopu a vysiela svetelný paprsok do oka pacienta. Svetlo prechádza očným tkanivom následne je odrazené späť pozorovateľovi. Štrbinová lampa slúži predovšetkým vyšetreniu prednej strany oka. Tak isto sa štrbinová lampa používa pri niektorých očných operačných zákrokoch, ako je napríklad katarakta [11].

- **Fundus kamera** tiež nazývaná retina kamera alebo oko kamera je zdravotnícky prístroj, ktorý foto-dokumentuje sietnicu oka. Je založená na podobnom princípe ako nepriama oftalmoskopia. Hlavná výhoda fundus kamery je, že dokáže ukladať digitálne obrázky sietnice oka. Vďaka tomu sa vie docieliť efektívnejšia diagnostika na základne porovnania obrázkov v čase. Týmto spôsobom lekár vie sledovať vývoj onemocnenia. Fundus kamera má rôzne režimy používania. Tieto režimy slúžia pre zvýraznenie určitej časti očného pozadia. Skladá sa z mikroskopu na ktorý je pripevnená kamera s CCD čipom. Výsledný obraz je priamy a niekoľkonásobne zväčšený [11] [14].



Obr. 2.5: Vľavo štrbinová kamera, vpravo fundus kamera

2.5 Zhrnutie

V úvode kapitoly bolo opísané ľudské oko. Podkapitola 2.1 ukázala hlavnú stavbu ľudského oka. Táto práca sa zameriava na sietnicu ľudského oka. Sietnica a jej základná stavba bola popísaná v podkapitole 2.2. Nasledujúca podkapitola 2.3 sa zameriavala na patologické nálezy na sietnici oka. Na sietnici oka vieme spozorovať: mikroaneurizmy, hemoragie, drúzy alebo exudáty. Jednotlivé patologické nálezy boli podrobnejšie popísané.

Posledná podkapitola 2.4 sa venovala vyšetreniu očného pozadia. Technológiám a prístrojom, ktoré sa používajú pri vyšetrení očného pozadia.

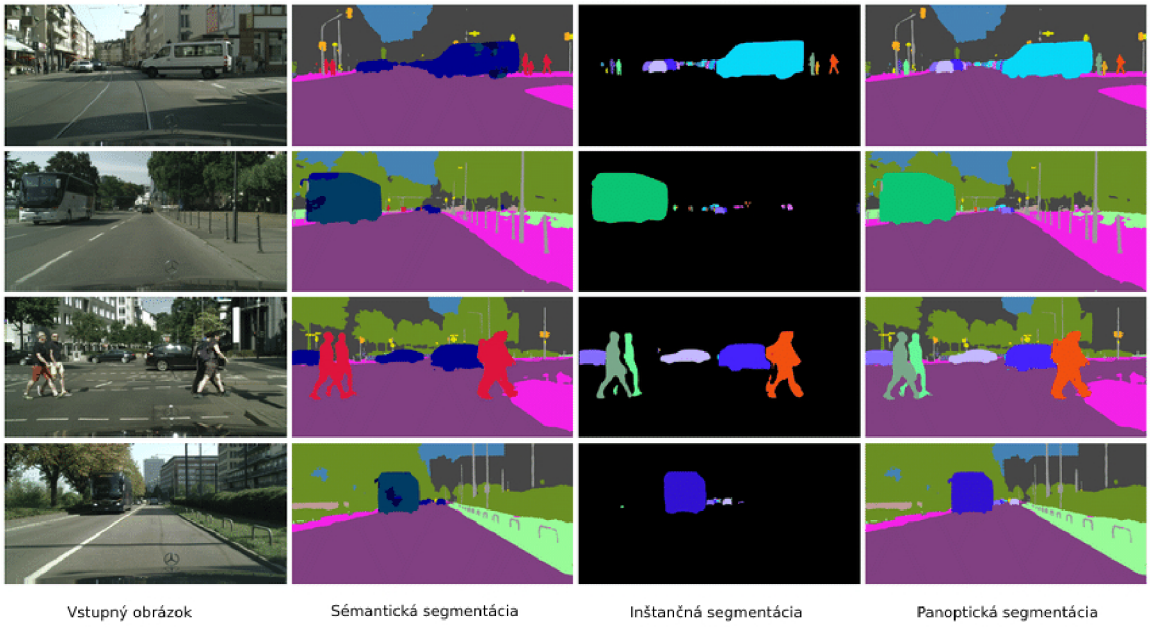
Kapitola 3

Neuronové siete pre sémantickú segmentáciu

Sémantická segmentácia sa v súčastnosti používa pri mnoho rozličných úlohách ako napríklad pri autonómnych autách, robotoch, v medicínskej sfére a pri mnoho ďalších využití. Sémantická segmentácia v porovnaní s rozličnými úlohami počítačového videnia, je komplikovanejšia kvôli potrebe množstva malých informácii z obrázka. S príchodom neurónových sietí a predovšetkým konvolučných neurónových sietí vzniklo mnoho modelov, ktoré s vysokou úspešnosťou vykonávajú sémantickú segmentáciu.

3.1 Úvod do sémantickej segmentácii

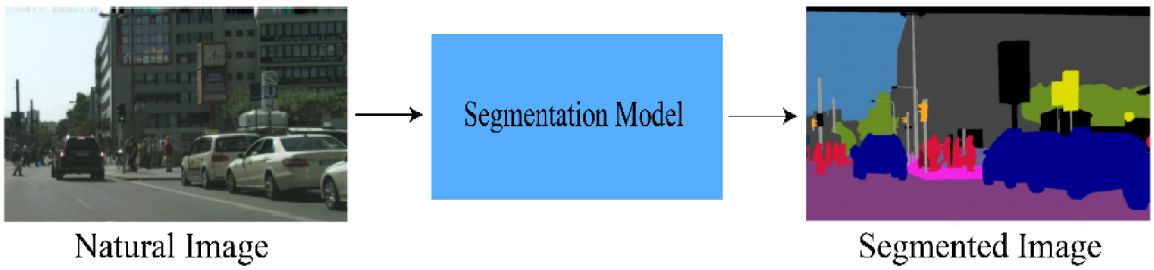
V počítačovom videní sa pod pojmom segmentácia z obrázka rozumie určité rozdelenie digitálneho obrázka do viacero možných kategórií podľa vlastnosti pixelov obrázka. Rozdiel oproti klasifikácií alebo detekcií je v tom, že ide o nízku úroveň videnia teda na úrovni pixelov. Je to z dôvodu toho, že priestorové informácie obrazu sú veľmi dôležité pre sémantickú segmentáciu. Segmentácia z obrázka sa rozdeľuje na sémantickú segmentáciu a inštančnú segmentáciu. Kombináciu obidvoch rozdelení vzniká panoptická segmentácia. Rozdiel medzi typmi segmentácie je v tom, ako sa pozerajú a spracovávajú veci v obraze. Sémantická segmentácia analyzuje každý pixel obrazu a priradí každému pixelu určitú triedu. Z toho dôvodu každý pixel bude mať priradenú triedu a predovšetkým farbu. Inštančná segmentácia sa zvyčajne zaoberá úlohami súvisiacimi so spočítateľnými vecami. Dokáže odhaliť každý objekt alebo inštanciu triedy prítomný na obrázku a priradiť mu masku alebo ohraničený rámček s jedinečným identifikátorom [35]. Panoptická segmentácia spojuje dva predchádzajúce typy a teda každému pixelu v obrázku je priradený sémantický štítok a jedinečný identifikátor inštancie [30] [18]. Obrázok 3.1 zobrazuje rozdiel medzi jednotlivými typmi segmentácie.



Obr. 3.1: Rozdiel medzi sémantickou, inštančnou a panoptickou segmentáciou [6]

3.2 Sémantická segmentácia

Sémantická segmentácia analyzuje každý pixel obrazu a priradí každému pixelu určitú triedu. Obrázok 3.2 ukazuje black-box pohľad na sémantickú segmentáciu, kde vstupom segmentačného modelu je klasický obrázok a výstupom je segmentovaný obrázok. Prvou úspešnou konvolučnou neurónovou sieťou sa označuje model AlexNet, ktorý vznikol v roku 2012. Od tohoto modelu vzniklo mnoho ďalších úspešných modelov, ktoré riešia sémantickú segmentáciu. Táto podkapitola sa venuje evolúcií modelov, ktoré vznikali postupne od spomínaného modelu AlexNet [30].

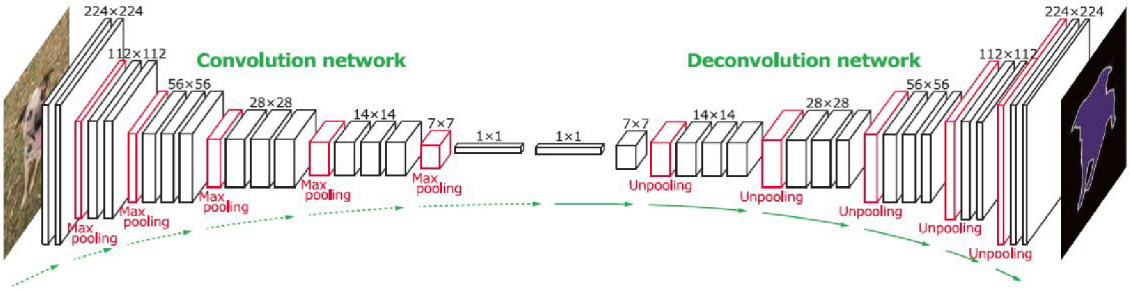


Obr. 3.2: Pohľad na proces sémantickej segmentácií [30]

3.2.1 Deconvnet

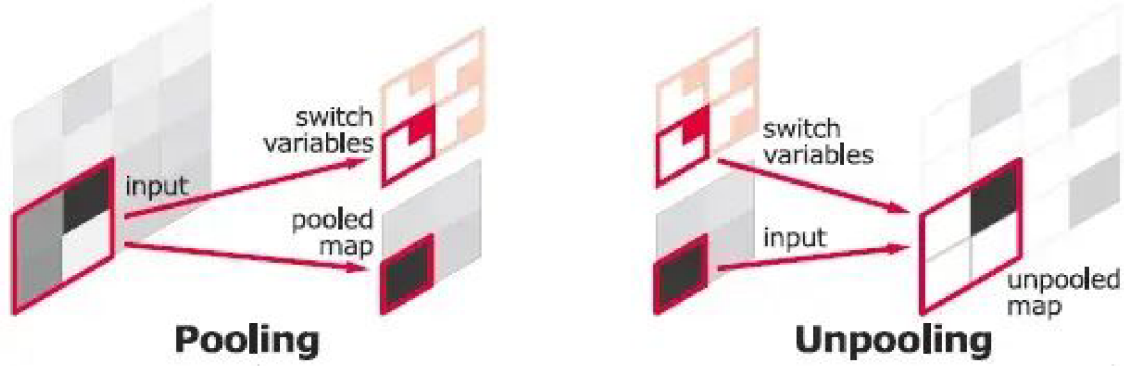
Názov modelu Deconvnet vychádza z dekonvolúcie. Tento model sa skladá v prvej časti z konvolučnej sietí a v druhej časti dekonvolučnej sietí. Prvá časť je typický konvolučná plne prepojená sieť ktorá sa skladá z konvolučných vrstiev a pooling vrstiev. Autori však

pridávajú novú časť a to dekonvolučnú časť sieti. Obrázok 3.3 zobrazuje model architektúry deconvnet.



Obr. 3.3: Deconvnet architektúra [23]

Pre vykonanie unpoolingu je potreba si zapamätať pozíciu každej maximálne aktivačnej hodnoty pri vykonaní max poolingu. Potom sa zapamätaná hodnota použije na unpooling. Tento postup zobrazuje obrázok 3.4.



Obr. 3.4: Zobrazuje zapamätanie si pozícií pri pooling a znova použitie pri unpoolingu[33]

Dekonvolúcia sa dá rozumieť ako konverzia vstupného obrázku späť na väčšiu veľkosť. Dekonvolučná sieť je identická s konvolučnou sieťou, ale je hierarchicky opačná. Všetky vrstvy konvolučnej a dekonvolučnej siete extrahujú mapy funkcií okrem poslednej vrstvy dekonvolučnej siete. Posledná vrstva generuje pixelové mapy pravdepodobnosti triedy, rovnakej veľkosti ako vstupný obrázok. Na rozdiel od plne prepojenej konvolučnej siete, autori aplikovali svoju sieť na návrhy objektov extrahované zo vstupného obrázka a vytvorili predikciu po pixeloch. Následne sú agregované výstupy všetkých návrhov do pôvodného obrazového priestoru pre segmentáciu celého obrazu. Tento prístup inteligentnej segmentácie spracováva viacrozmerné objekty s jemnými detailami a tiež znižuje zložitosť trénovanie, ako aj spotrebu pamäte pre trénovanie. Na zvládnutie vnútorného posunu kovariátu v sieti, autori použili batch normalizáciu na vrch konvolučnej a dekonvolučnej vrstvy [23] [30].

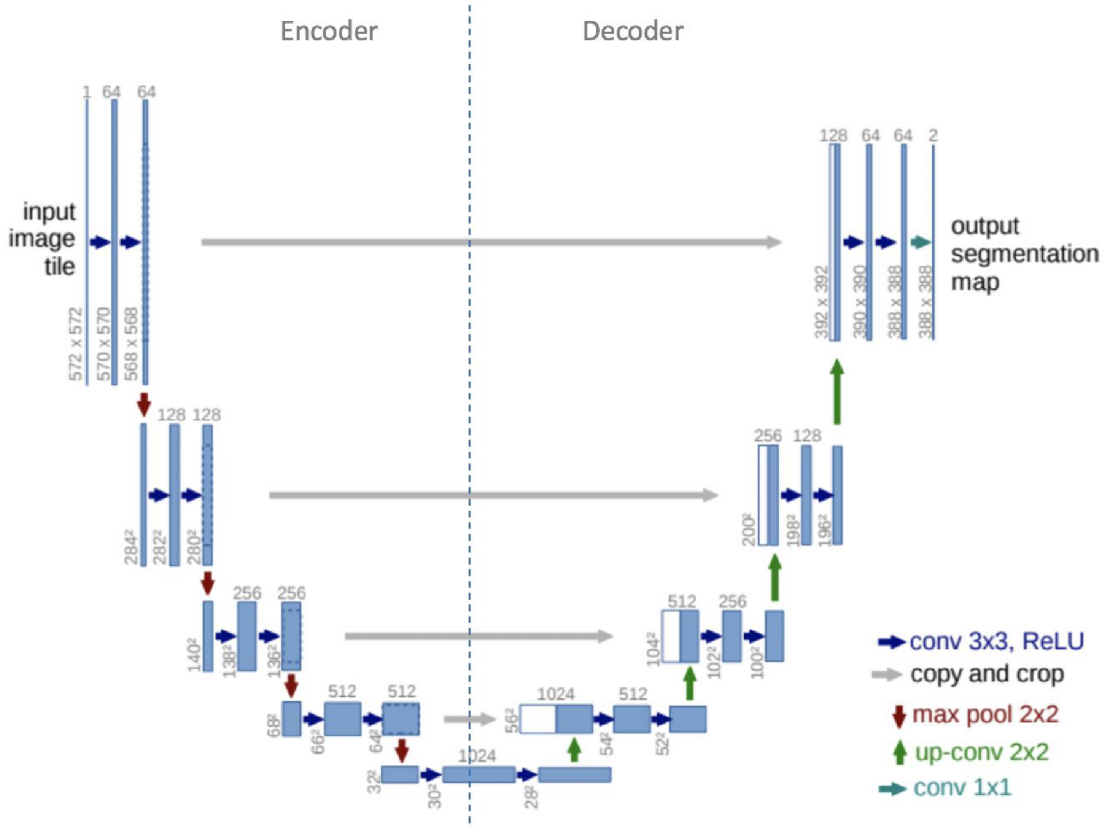
3.2.2 U-Net

Model U-Net je plne prepojenou konvolučnou neurónovou sieťou. Tento model bol vytvorený pre biomedicínsku segmentáciu obrazu a názov si nesie podľa písmena U, kvôli podobnosti tvaru modelu s písmenom U. U-Net architektúra je upravená tak, aby sa učila z malého

počtu obrázkov a pritom poskytovala presnejšiu segmentáciu objektov v obraze. Segmentácia obrázka o veľkosti 512x512 pixelov zaberie menej ako jednu sekundu na súčasných moderných grafických kartách.

Vo všeobecnosti môžeme model chápať ako sieť kodéra, za ktorou nasleduje sieť dekodéra. Ale nejedná sa o klasickú architektúru kodér - dekodér. Obrázok 3.5 zobrazuje U-Net architektúru. Na obrázku je vidieť spomínané rozdelenie na časť kodéra a dekodéra.

- **Kodér** je prvou polovicou architektúry. Zvyčajne ide o vopred trénovanú klasifikačnú sieť ako VGG/ResNet kde sa aplikujú konvolučné bloky. Po týchto blokoch nasleduje maxpool downsampling na zakódovanie vstupného obrazu do reprezentácií funkcií na viacerých úrovniach [32] [28].
- **Dekodér** je druhá polovica architektúry. Cieľom dekodéru je semanticky premietnuť diskriminačné vlastnosti (nižšie rozlíšenie), ktoré sa naučil kodér do priestoru pixelov (vyššie rozlíšenie). Dekodér sa skladá z upsampling-u a zreťazenia, po ktorých nasledujú pravidelné konvolučné operácie [28] [1].

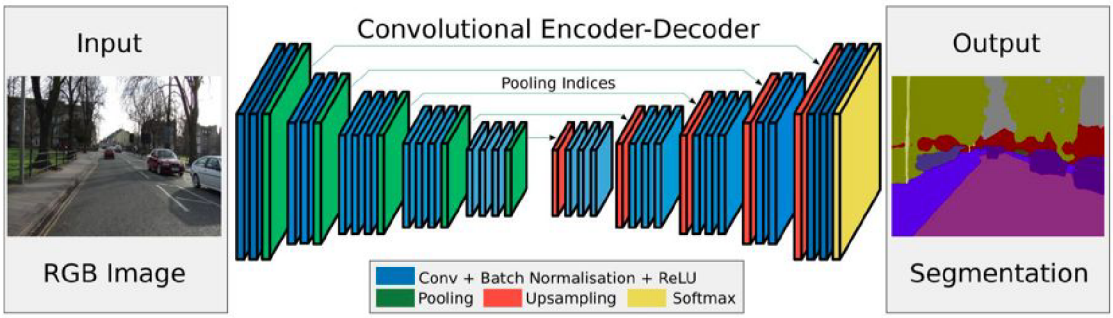


Obr. 3.5: U-Net architektúra, modré boxy reprezentujú viackanálové mapy objektov, biele boxy reprezentujú skopírované mapy objektov. Farebné šípky reprezentujú rôzne operácie. [1]

3.2.3 SegNet

Model SegNet je architektúrou kodér-dekodér po ktorej nasleduje konečná klasifikačná vrstva po pixloch. Kodér obsahuje 13 konvolučných vrstiev ako VGG16 a zodpovedajúca

časť dekodéra ma tiež 13 dekonvolučných vrstiev. Autori nepoužili plne prepojené vrstvy ako to je u VGG16 a tak isto zmenili počet parametrov na 14,7 milióna z 134 miliónov parametrov. V každej vrstve kodéra konvolučné operácie sú vykonávane pomocou banky filtrov na vytvorenie máp prvkov. Na redukciu kovariantného posunu autori použili batch normalizáciu nasledovanú nelineárnou operáciou ReLU. Na Výsledné výstupné mapy funkcií je použitý max-pooling pomocou 2x2 neprekrývajúcim oknom. Po max-poolingu nasleduje sub-sampling operácia. Kombináciou max-pooling a sub-samplingu sa dosahuje lepšia presnosť klasifikácie, ale znižuje sa veľkosť máp prvkov, čo vedie k stratovej reprezentácií obrázka s rozmazanými hranicami. V rámci úlohy sémantickej segmentácií je dôležité mať potrebné informácie o hraniciach. Pre zachovanie informácií o hraniciach v mapách prkov kódera SegNet ukladá iba indexy max-poolingu pre každú mapu kodéra. Dôležitou úlohou semantickej segmentácií je zachovať veľkosť vstupného obrázka a výstupného obrázka. SegNet vykonáva up-sampling vo svojom dekodéri pomocou uložených indexov max-poolingu z príslušnej mapy funkcií kodéra. To vedie k riedkej mape funkcií vo vysokom rozlíšení. Aby boli mapy prvkov husté, operácia konvolúcie sa vykonáva pomocou trénovateľnej banky filtrov dekodéra. Následne sa použije batch normalizácia. Výstupná mapa funkcií s vysokým rozlíšením z finálneho dekodéra je privedený do trénovateľnej multi-class softmax klasifikátora pre označovanie po pixloch [3] [30]. Architektúra SegNet je znázornená na obrázku 3.6.



Obr. 3.6: SegNet architektúra [3]

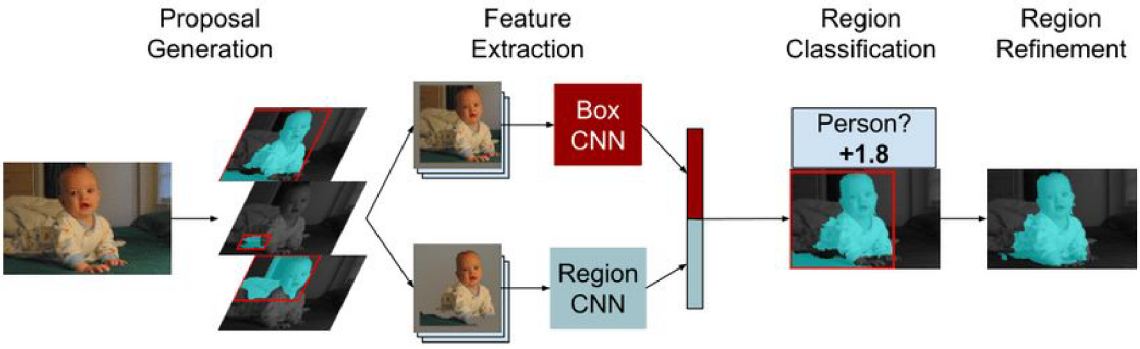
3.3 Inštančná segmentácia

Inštančná segmentácia nezávisle maskuje každú inštanciu objektu v obraze. Úlohou detekcie objektov a inštančnej segmentácií spolu súvisia. Pri detekcií objektov sa používajú ohraničujúci rámček na detekciu každej inštancie objektu s označením na klasifikáciu. Inštančná segmentácia berie rovnaký princíp a pridáva pre každú inštanciu objektu ešte masku segmentácie. Rovnako výskumníci začali používať konvolučné neuronové siete pre zlepšenie presnosti segmentácie [30].

3.3.1 SDS

Skrátka SDS je z originálneho názvu *Simultaneous Detection and Segmentation*. Jedná sa o architektúru ktorá je založená na generovaní návrhu ohraničujúceho okna. SDS sa skladá zo štyroch krokov: proposal generation, feature extraction, region classification a region refinement. Tieto kroky si môžeme všimnúť na obrázku 3.7. Na vstupnom obrázku autori

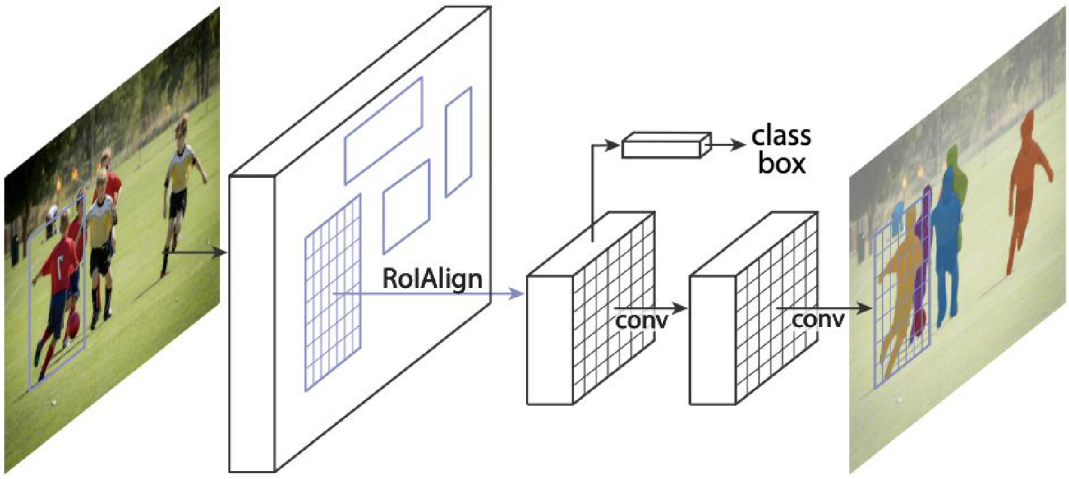
použili algoritmus Multi-scale Combinatorial Grouping. Tento algoritmus generuje návrhy regiónov na obrázku. Následne každý región je daný do dvoch paralelne idúcich konvolučných neurónových sietí. Vrchná sieť znázornená červenou farbou je určená na generovanie vektorových znakov pre ohraničujúce rámčeky navrhovaných regiónov. Spodná sieť znázornená červenou farbou je určená na generovanie vektorových znakov pre segmentačnú masku. Výsledkom týchto dvoch sietí sú dva vektory, ktoré sú následne spojené. Nad spojeným vektorom sa skóre tried pre každého kandidáta na objekt za pomoc SVM predikčného modelu. Potom sa na skórovaných kandidátoch aplikuje nemaximálne potlačenie, aby sa znížil počet kandidátov na objekt v rovnakej kategórií. Nakoniec na spresnenie zachovaných kandidátov sú použité mapy funkcií z konvolučnej neurónovej sieti [9] [30].



Obr. 3.7: SDS architektúra [9]

3.3.2 Mask R-CNN

Architektúra Mask R-CNN sa skladá z troch vetiev na predpovedanie tried, ohraničenia a segmentačnej masky pre prípady v rámci oblasti záujmu. Rovnako ako architektúra SDS je založená na generovaní návrchu ohraničujúceho okna. Je rozšíreným modelom Faster-R-CNN. Rovnako ako Faster-R-CNN obsahuje dva stupne. V prvom stupni, používa RPN na generovanie oblasti záujmu a zachováva sa priestorové umiestnenie. Autori použili RoIAlign, ktorá je operáciou na extrakciu malej mapy prvkov z každej oblasti záujmu. V druhom stupni sa súčasne predpovedá označenie triedy, posun ohraničujúceho okna a binárnu masku pre každú oblasť záujmu. Predpoveď binárnej masky pre každú triedu je nezávislé a nie je to predikcia viacerých tried [10] [30]. Obrázok 3.8 zobrazuje architektúru Mask R-CNN.

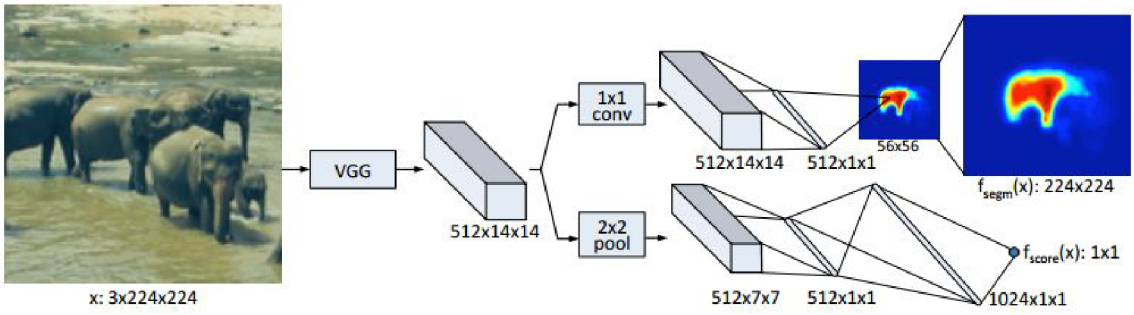


Obr. 3.8: Mask-R-CNN architektúra [10]

Generovanie ohraničujúceho okna je výpočtovo efektívne a užitočné pri detekcií objektu. Vedie to však k veľmi výpočtovo nákladným postupom pri postupe zarovnávania. Tieto modedy ako Mask R-CNN alebo SDS, ktoré sú založené na generovaní ohraničujúceho okna tiež musia generovať masky pre každú inštanciu samostatne. Z tohoto dôvodu vedci sa pokúsili vytvoriť architektúry založené na generovania návrhu masky segmentácie.

3.3.3 DeepMask

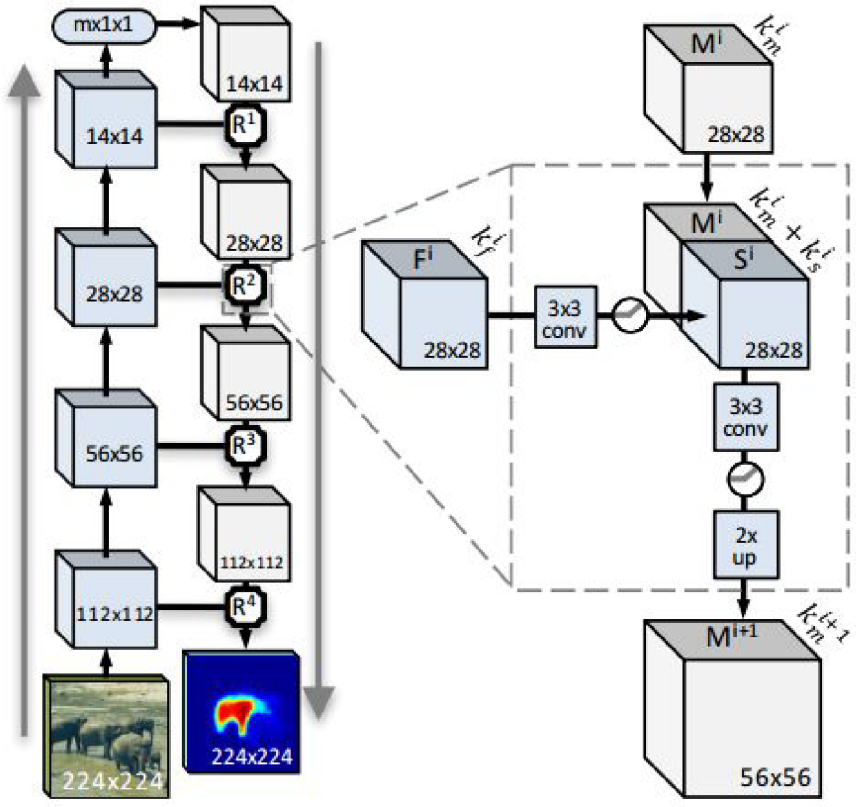
Jednou architektúrou, ktorá je založená na generovaniu návrhu masky segmentácie je model DeepMask. DeepMask použil konvolučné neurónové siete na generovanie segmentačných návrhov namiesto generovania ohraničujúceho okna. Z obrázku 3.9 je vidieť, že mapy prvkov sa privedú do dvoch paralelne vedúcich vetiev. Horná vetva je založená na konvolučnej neurónovej sieti a predpovedá triedu segmentačnej masky. Spodná vetva priraďuje skóre na odhad pravdepodobnosti záplaty, ktorá sa sústredí na celý objekt. Parametre siete sú navzájom zdieľané medzi obidvoma vetvami [25] [30].



Obr. 3.9: DeepMask architektúra [25]

3.3.4 SharpMask

O vylepšenie DeepMask modelu sa snaží architektúra SharpMask. DeepMask generuje presné masky na úrovni objektu, ale stupeň zarovnania masky so skutočnými hranicami objektu nebol dobrý. SharpMask obsahuje doprednú neurónovú sieť zdola nahor na vytváranie hrubej sémantickej segmentačnej masky a sieť zhora nadol na spresnenie týchto masiek pomocou spresňovacieho modulu. Autori vzali doprednú sieť návrhov segmentácie DeepMask a pridali so svojim modulom na spresňovanie. Na obrázku 3.10 je vidieť, že konvolučná neurónová sieť zdola nahor vytvára kódovanie hrubej masky. Nasleduje kódovanie výstupnej masky privedené do siete zhora nadol, kde ho spresňovací model rozdelí pomocou zodpovedajúcich funkcií z modelu zdola nahor. Tento proces sa opakuje, až do rekonštrukcie obrazu v plnom rozlíšení s maskou finálneho objektu [26] [30].

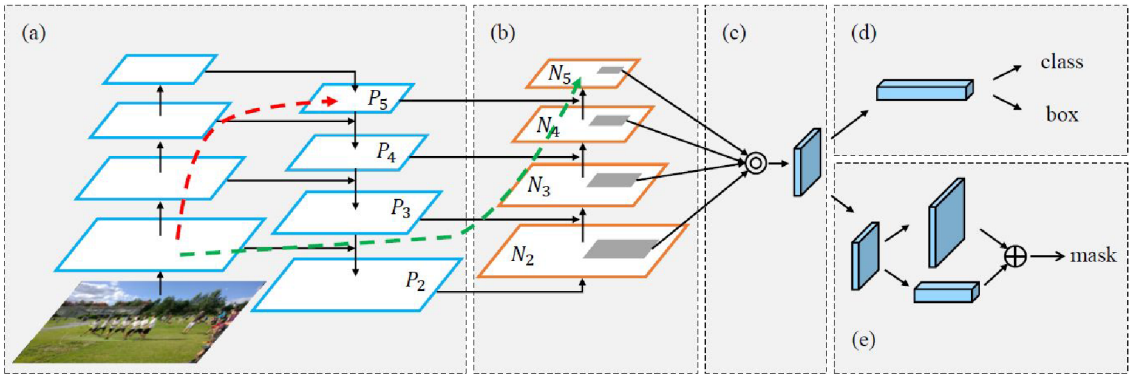


Obr. 3.10: SharpMask architektúra [26]

3.3.5 PANet

Architektúra PANet je založená na šírení funkcií. Je to iný pohľad ako generovanie ohraničujúceho okna alebo generovanie návrhu masky. Pri konvolučných neurónových sietiach je veľmi dôležitý tok informácií, pretože mapy prvkov na nízkej úrovni majú bohaté a dôležité informácie z hľadiska lokalizácie. Mapy prvkov na vysokej úrovni pre zmenu sú bohaté na sémantické informácie. Z toho dôvodu vznikol model PANet, ktorý je kombináciou Mask R-CNN a Feature Pyramid Network (FPN). Architektúra PANet použila FPN ako svoju základnú sieť na extrahovanie funkcií z rôznych vrstiev. Na šírenie funkcie nízkej vrstvy cez sieť sa používa rozšírená cesta zdola nahor. Výstup každej vrstvy sa generuje pomocou

mapy prvkov s vysokým rozlíšením predchádzajúcich vrstiev a hrubej mapy z FPN pomocou bočného prepojenia. Následne sa použije adaptívna vrstva na agregáciu funkcií zo všetkých úrovní. V tejto vrstve sa vrstva RoiPooling používa na spojenie prvkov v každej úrovne pyramídy a na spojenie prvkov sa používa operácia maxima alebo súčtu prvkov. Z architektúry Mask R-CNN je použité to, že výstup vrstvy združovania funkcií ide do troch vrstiev. Na predikciu ohraničujúceho okna, predikciu triedy objektu a predikciu binárnej masky pixelov. Pomocou siete na šírenie funkcií a združovacej pyramídy tento model zahŕňa funkcie nízkej aj vysokej úrovne. Tieto funkcie zaobstarajú bohaté informácie pre inštančnú segmentáciu [17] [30]. Obrázok 3.11 zobrazuje PANet architektúru.



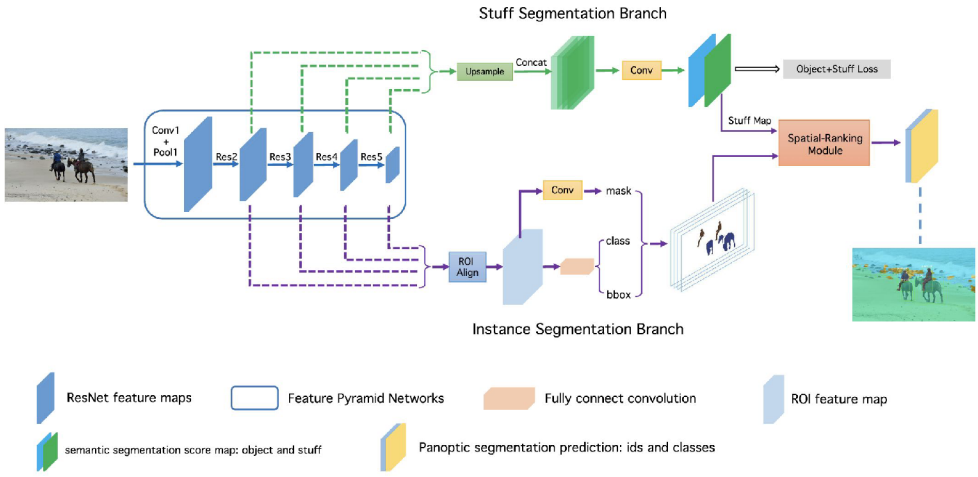
Obr. 3.11: PANet architektúra [17]

3.4 Panoptická segmentácia

Panoptická segmentácia je kombináciou sémantickej a inštančnej segmentácií. V súčastnosti sa jedná o novú oblasť výskumu. V panoptickej segmentáciá sa musia priradiť všetky pixely obrázku k sémantickému označeniu pre klasifikáciu a tiež identifikovať inštancie konkrétnej triedy. Výstupom bude model ktorý obsahuje dva kanály. Jeden kanál pre označenie pixelu (sémantická segmentácia) a druhý kanál pre predpovedanie každej inštancie pixelu (inštančná segmentácia). Následujúce sekcie poukazujú na súčasné modely panoptickej segmentácie [30].

3.4.1 OANet

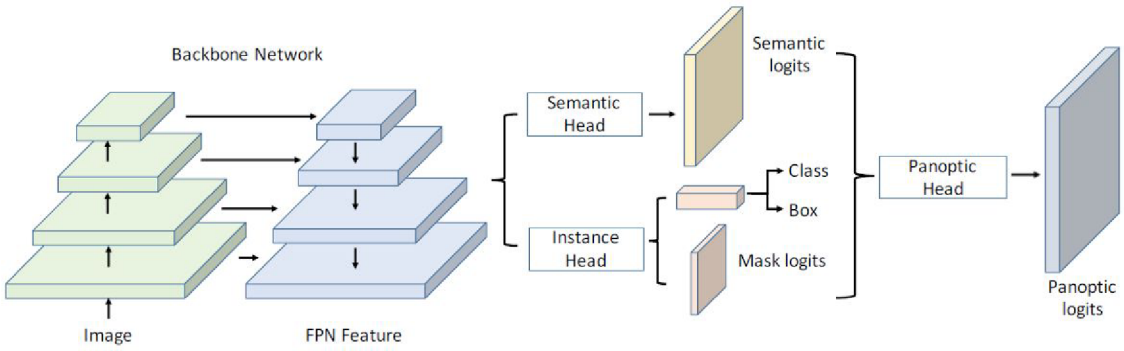
V tejto architektúre autori použili sieť Feature Pyramid Network na extrahovanie máp objektov zo vstupného obrázka. Na extrahovanie prvkov použili dve rôzne vetvy: jednu pre sémantickú segmentáciu a druhú pre inštančnú segmentáciu. Maska R-CNN sa používa pri inštančnej segmentácií. Výstup obidvoch vetiev je na vstupe modulu spatial ranking, ktorý vytvorí konečný výsledok panoptickej segmentácie [16] [30]. Obrázok 3.12 zobrazuje OANet architektúru.



Obr. 3.12: OANet architektúra [16]

3.4.2 UPSNet

Ďalšou architektúrou panoptickej segmentácie je model UPSNet. Táto architektúra vznikla za pomoci spoločnosti Uber, Torontskej univerzity a Čínskej univerzite v Hong Kongu. Autori použili ResNet a Mask R-CNN založenú na FPN ako backbone sieť na extrahovanie konvolučnej mapy funkcií. Tieto konvolučné mapy prvkov sú dodávané do troch podsietí: pre sémantickú segmentáciu, inštančnú segmentáciu a panoptickú segmentáciu. Sémantická segmentačná podsieť pozostáva z deformovateľnej konvolučnej siete na segmentovanie tried. Podsieť inštančnej segmentácií pozostáva z troch vetiev pre regresiu hraničného okna, klasifikácie a segmentačnej masky. Výstupy z obidvoch podsietí ďalej smerujú do podsiete panoptickej segmentácií na konečnú panoptickú segmentáciu obrázka [40] [30]. Obrázok 3.13 zobrazuje UPSNet architektúru.



Obr. 3.13: UPSNet architektúra [40]

3.5 SETR

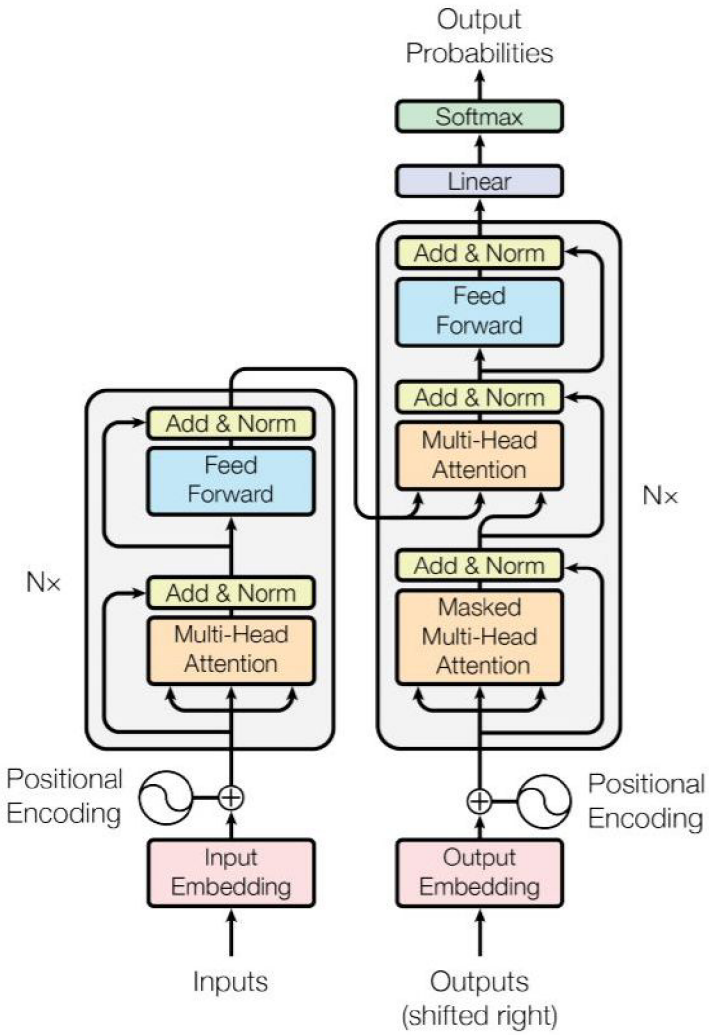
Všetky predchádzajúce architektúry boli klasickými modelmi, kde sa hlavne využíval kodér a dekodér, aj keď nie u všetkých sietí bol kodér a dekodér ale princíp bol rovnaký. V roku 2021 prišli vedci z Fudánskej univerzity, Oxfordskej univerzity, univerzity v Surrey za pomoci

Tencet Youtu labu a Facebook AI spoločnosti na prehodnotenie sémantickej segmentácií a využitia sekvencie na sekvenciu s transformátormi. Táto nová architektúra má skratku SETR od jej názvu segmentačný transformátor.

3.5.1 Transformátory

Transformátory sa využívajú v neurónových sieťach predovšetkým pri spracovaní prirodzeného jazyka. Využívajú sa na riešenie úlohu medzi sekvenciami a zároveň na ľahšie zvládnutie závislostí na dlhé vzdialenosti. Prvýkrát boli transformátory predstavené v tomto odbornom článku [34]. Na základe článku sa stali veľmi populárne a začali sa využívať v rozličných modeloch [22].

Transformátorové modely používajú vyvíjajúci sa súbor matematických techník, nazývaných pozornosť (z angl. attention) alebo sebapozornosť (z angl. self-attention), na detekciu.



Obr. 3.14: Transformátor - architektúra [34]

Kodér

Je ľavá časť modelu 3.14. Kodér pozostáva zo zásobníka N = 6 rovnakých vrstiev, pričom každá vrstva sa skladá z dvoch podvrstiev:

1. Prvá podvrstva implementuje multi-head self-attention mechanizmus. Multi-head mechanizmus implementuje hlavy, ktoré dostávajú rozličnú lineárne vedenú verziu dopytov, kľúčov a hodnôt, pričom každá z nich vytvára paralelne výstupy, ktoré sa potom používajú na generovanie konečného výsledku.

2. Druhá podvrstva je plne prepojenou doprednou neurónovou sieťou, pozostávajúcou z dvoch lineárnych transformácií s aktivačnou funkciou ReLU

$$FFN(x) = ReLU(W_1 x + b_1)W_2 + b_2 \tag{3.1}$$

Všetkých šesť vrstiev kodéra aplikuje rovnaké lineárne transformácie na všetky slová vo vstupnej sekvencií, ale každá vrstva na to používa iné parametre váhy (W_1, W_2) a odchýlky (b_1, b_2). Okrem toho má každá z týchto dvoch podvrstiev okolo seba zvyškové spojenie. Po každej podvrstve nasleduje normalizačná vrstva, $layernorm(.)$ ktorá normalizuje súčet vypočítaný medzi vstupom podvrstvy x a výstupom generovaným samotnou podvrstvou, $sublayer(x)$:

$$layernorm(x + sublayer(x)) \tag{3.2}$$

Architektúra transformátora nemôže zo svojej podstaty zachytiť žiadne informácie o relatívnych pozíciach slov v sekvencií, pretože sa využíva opakovanie. Informácia sa teda musí vložiť zavedením pozičného kódovania do vstupných vložení. Polohové kódovacie vektory majú rovnaký rozmer ako vstupné vloženia a sú generované pomocou sínusových a kosínusových funkcií rôznych frekvencií. Potom sa jednoducho spočítajú s input embedding, aby sa vložili polohové informácie [41].

Dekodér

Dekodér je pravá časť modelu 3.14. Každý dekodér pozostáva z troch komponentov: self-attention mechanizmu, attention mechanizmu nad kódovaním a doprednú neurónovú sieť. Dekodér funguje podobne ako kodér. Rovnako pozostáva zo zásobníka N = 6 rovnakých vrstiev, ktoré každá ma tri podvrstvy. Vložený je dodatočný mechanizmus attention. Tento mechanizmus čerpá relevantné informácie z kódovania generovaného kodérmi. Môžeme tento mechanizmus nazvať aj kodér-dekodér attention (z angl. encoder-decoder attention) [41].

1. Prvá podvrstva prijíma predchádzajúci vstup zo zásobníka dekodérov. Rozširuje ho o polohové informácie a implementuje nad ním multi-head self-attention. Zatiaľ čo kodér je navrhnutý tak, aby sa staral o všetky slová vo vstupnej sekvencii bez ohľadu na ich pozíciu v sekvencii. Dekodér je upravený tak, aby sa staral iba o predchádzajúce slová. Predikcia pre slovo na pozícií môže závisieť len od známych výstupov pre slová, ktoré sú pred ním v sekvencií. V multi-head attention mechanizme (ktorý implementuje paralelne viaceré funkcie jednej pozornosti) sa to dosiahne zavedením masky nad hodnotami vytvorenými škálovaným násobením matíc Q a K. Maskovanie je implementované potlačením maticových hodnôt, ktoré by inak zodpovedali nelegálnym spojeniam:

$$mask(QK^T) = mask \begin{bmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{bmatrix} = \begin{bmatrix} e_{11} & -\infty & \dots & -\infty \\ e_{21} & e_{22} & \dots & -\infty \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{bmatrix} \tag{3.3}$$

2. Druhá podvrstva implementuje multi-head self-attention mechanizmus podobný tomu, ktorý je implementovaný v prvej podvrstve kodéra. Na strane dekodéra tento multi-head mechanizmus prijíma hodnoty z predchádzajúcej podvrstvy dekodéra, kľúče a hodnoty z výstupu kodéra. Tento mechanizmus umožňuje dekodéru venovať pozornosť všetkým slovám vo vstupnej sekvencii.

3. Tretia podvrstva implementuje plne prepojenú doprednú sieť podobnú tej, ktorá je implementovaná v druhej podvrstve kodéra.

Okrem toho tri podvrstvy na strane dekodéra majú okolo seba tiež reziduálne spojenia a za nimi nasleduje normalizačná vrstva.

SETR sa najviac podobá na architektúru [36]. Existujú však niektoré kľúčové rozdiely. V obidvoch modeloch je konvolúcia úplne odstránená, lenže architektúra [36] nasleduje konvolučný dizajn plne prepojenej konvolučnej siete, tým že sa postupne zmenšuje priestorové rozlíšenie prvku. V architektúre SETR si predikčný model medzi sekvenciami uchováva rovnaké priestorové rozlíšenie a predstavuje tak skokovú zmenu v dizajne modelu. Druhý rozdiel je v maximalizovaní škálovateľnosti na moderných hardvérových akcelerátoroch a zjednodušenie ich používanie. Architektúra [36] namiesto toho používa špeciálne navrhnuté axiálne zameranie, ktoré je menej škálovateľné na štandardnom výpočtovom zariadení. Tak isto autori SETR ukazujú, že model má lepšiu presnosť segmentácií. Z obrázka 3.15 si vieme všimnúť, že SETR sa dá rozdeliť na tri časti. Kde prvá časť a) značí kodér a zvyšné časti sú možné typy dekóderov.

Prvá časť 3.15 má za úlohu preniesť obrázok na vektory. Priamočiaro sa dá obrázok sekventizovať tak, že sa vstupný obrázok sploští a hodnoty pixlov sa dajú do 1D vektora o veľkosti 3 * výška * šírka. Pre typický obrázok o veľkosti 480 x 480 je výsledná dĺžka vektora 691 200. Transformátor má kvadratickú zložitosť a teda taký vysoko dimenzovaný vektor by nešlo spracovať v reálnom čase. Klasický kodér pre sémantickú segmentáciu zníži 2D obrázok $x \in R^{H \times W \times 3}$ na mapu funkcií $x_f \in R^{\frac{H}{16} \times \frac{w}{16} \times C}$. Autori SETR sa rozhodli nastaviť dĺžku vstupnej sekvencie transformátora L na veľkosť $\frac{H}{16} \times \frac{W}{16} = \frac{HW}{256}$. Pre získanie takej veľkosti sa obrázok $x \in R^{H \times W \times 3}$ rozdelí na mriežku o veľkosti $\frac{H}{16} \times \frac{W}{16}$ na rovnomerne patche a následne sa sploští táto mriežka na sekvencie. Ďalej sa mapuje každý vektor patchu p na latentnú C-dimenzionálny pomocou lineárnej projekčnej funkcie $f : p \longrightarrow e \in R^C$. Týmto sa dosiahla 1D sekvencia patchu pre obrázok x. Na zakodóvanie priestorových informácií patchu, sa naučí špecifický embedding p_i pre každé miesto i, ktoré sa pridá do e_i na vytvorenie finálnej sekvencie vstupu $E = \{e_1 + p_1, e_2 + p_2, \cdots, e_L + p_L\}$. Týmto spôsobom sa priestorové informácie uchovávajú napriek self-attention povahe transformátorov. Týmto sa vkladá 1D sekvencia E do čistého transformátora, ktorý sa naučí reprezentáciu funkcií. To znamená, že každý vrstva transformátora má globálne receptívne pole, čím sa raz a navždy vyrieši problém obmedzeného receptívneho poľa existujúceho kodéra FCN.

Transformátorový kodér pozostáva L_e vrstiev multi-head self attention (MSA) a multilayer perceptron (MLP) blokov. Na každej vrstve l, je vstupom do self-attention mechanizmu trojica (*query key value*), ktorá je vypočítaná zo vstupu $Z^{l-1} \in R^{L \times C}$ ako:

$$query = Z^{l-1}\mathbf{w}_Q, key = Z^{l-1}\mathbf{w}_K, value = Z^{l-1}\mathbf{w}_v \tag{3.4}$$

kde $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_v \in \mathbb{R}^{C\times d}$ sú parametre troch vrstiev lineárnej projekcie (*query*, *key*, *value*), ktoré sa môžu učiť. Self-attention (SA) je formulovaný ako

$$SA(Z^{l-1}) = Z^{l-1} + softmax(\frac{Z^{l-1}\mathbf{W}_Q(Z\mathbf{W}_K)^T}{\sqrt{d}})(Z^{l-1}\mathbf{W}_v) \tag{3.5}$$

MSA je rozšírením s m nezávislými SA operáciami a následne zreťazia sa výstupy: $MSA(Z^{l-1} = [SA_1(Z^{l-1}); SA_2(Z^{l-1}); \cdots ; SA_m(Z^{l-1})]\mathbf{W}_O$ kde $\mathbf{W}_0 \in R^{md\times C}$. Parameter d je typický nastavený na C/m. Výstup MSA je transformovaný MLP blokom s reziduálnym preskočením a výstup vrstvy je:

$$Z^l = MSA(Z^{l-1} + MLP(MSA(Z^{l-1})) \in R^{L\times C} \tag{3.6}$$

Normalizácia je aplikovaná ešte pred MSA a MLP blokmi kvôli jednoduchosti je to vynechané. Vlastnosti vrstiev transformátora označujeme $\{Z^1, Z^2, \cdots, Z^{L_e}\}$.

Na vyhodnotenie účinnosti kodéra reprezentácie Z autori navrhlo tri rôzne návrhy dekodéra, ktoré vykonávajú segmentáciu na úrovni pixelov. Úlohou dekodéra je generovať segmentáciu v rovnakom rozlíšení 2D ako bol vstup obrázka [41].

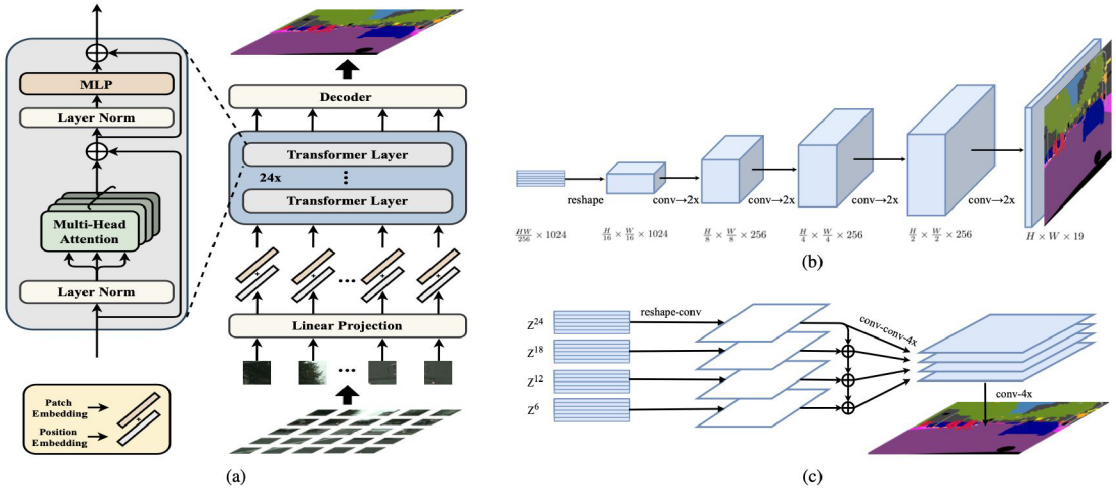
- **Naive upsampling (Naive)** Tento naivný prístup na začiatku premieta transformačné funkcie Z^{L^e} na dimenziu čísla kategórie. Na tento účel sa používa jednoduchá 2 vrstvová sieť s architektúrou 1 x 1 konvolúcia + synchronizačná batch normalizácia (w /ReLU) + 1 x 1 konvolúcia. Následne sa bilineárne prevzorkuje výstup na plné rozlíšenie obrazu, po ktorom nasleduje klasifikačná vrstva s pixel-wise cross entropy loss funkciou. Ak sa použije tento dekodér označuje sa celá architektúra ako SETR-*Naive* [41].

- **Progressive UPsampling (PUP)** Tento typ dekodéru namiesto jednostupňového pre-vzorkovania uvažuje o progresívnom stratégií pre-vzorkovania, ktorá strieda konverzie vrstiev a operácie pre-vzorkovania. Jednostupňové pre-vzorkovanie môže viesť ku zlým predpovediam, ktoré nebudú veľmi presné pre segmentačnú úlohu. Pre maximálne zmiernenie nepriaznivého účinku sa obmedzí pre-vzorkovanie na 2x. To zaručí to, že treba 4 operácie na dosiahnutie maximálnej veľkosti obrázka z Z^{L^e} o veľkosti $\frac{H}{16} \times \frac{W}{16}$. Tento prístup dekodéra je zobrazený na obrázku 3.15 (b). Pri použití tohoto dekodéra sa architektúra nazýva SETR-*PUP* [41].

- **Multi-Level feature Aggregation (MLA)** Tretí typ dekodéra sa vyznačuje viacúrovňovou agregáciou funkcií. Podobá sa na pyramídovú sieť no má veľa rozličného. Reprezentácia funkcií Z^L každej SETR vrstvy zdieľa rovnaké rozlíšenie bez pyramídového tvaru.

  Špeciálne sa berie vstup reprezentácií funkcií $\{Z^m\}(m \in \{\frac{L_e}{M}, 2\frac{L_e}{M}, \cdots, M\frac{L_e}{M}\})$ z M rovnomerne rozložených vrstiev. Potom sa nasadí M stream, pričom každý sa zameria na jednu konkrétnu vybranú vrstvu. V každom streame sa najskôr pretvorí funkcia kodéra Z^L z 2D tvaru $\frac{HW}{256} \times C$ do 3D máp funkcií $\frac{H}{16} \times \frac{W}{16} \times C$. Trojvrstvová

sieť (kernelová veľkosť 1x1, 3x3 a 3x3) sa aplikuje s kanálmi funkcií na polovicu prvej a tretej vrstve. Priestorové rozlíšenie sa zvýši 4x bilineárnou operáciou po tretej vrstve. Na zlepšenie interakcií naprieč rôznymi streamami sa používa dizajn agregácie zhora nadol prostredníctvom pridávania prvkov po prvej vrstve. Pridáva sa ďalšia 3x3 konvolučná vrstva po funkcií pridávanej po prvkoch. Po tretej vrstve sa získa fúzová funkcia zo všetkých streamov prostredníctvom konktatenácie kanálov, ktoré sa potom bilineárne prevzorkujú 4x na plné rozlíšenie. Pri použití tohoto dekodéra sa jedná o architektúru SETR-*MLA*. Obrázok 3.15 (c) zobrazuje dekodér MLA [41].

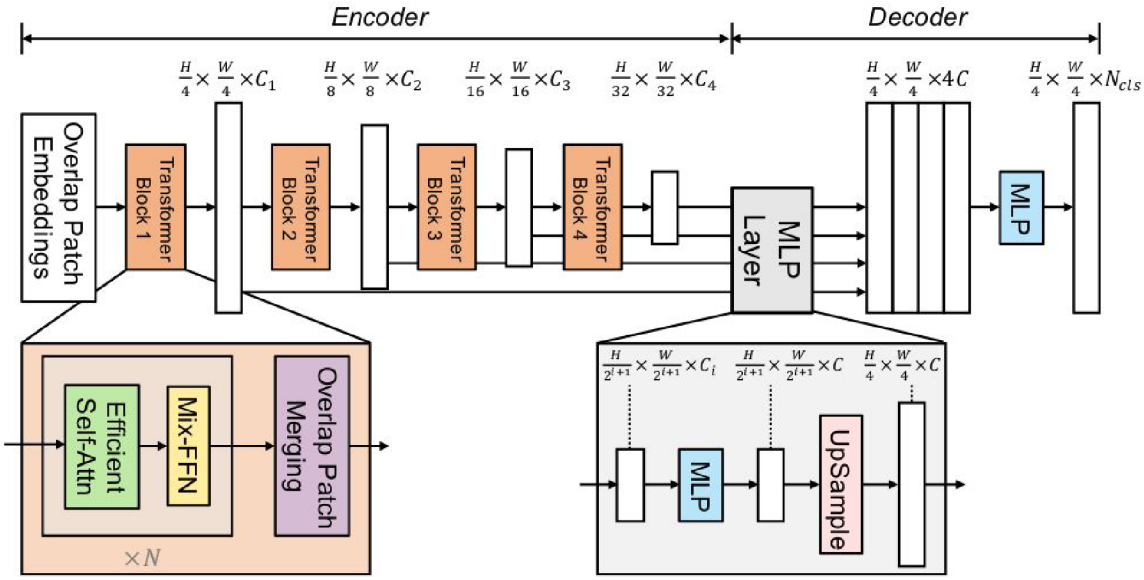


Obr. 3.15: Model architektúry segmentácie sekvencie na sekvenciu [41]

3.6 SegFormer

V tom istom roku, po vzniknutí publikácií architektúry SETR. Vznikla publikácia s názvom SegFormer. SegFormer je jednoduchý, efektívny framework pre sémantickú segmentáciu, ktorá zjednocuje transformátory s ľahkými dekodérmi viacvrstvového perceptrónu z anglického *multilayer perceptron* (MLP). SegFormer sa skladá z dvoch hlavných modulov: (1) hierarchický kodér transformátorov na generovanie hrubých funkcií s vysokým rozlíšením a jemných funkcií s nízkym rozlíšením a (2) ľahký All-MLP dekodér na zlúčenie týchto viacúrovňových funkcií a vytvorenie finálnej masky sémantickej segmentácie. Tieto hlavné moduly sú zobrazené na obrázku 3.16.

SegFormer vstupný obrázok o veľkosti $H \times W \times 3$ sa rozdelí na patche o veľkosti 4×4. Rozdiel oproti ViT [8] je v tom, že ViT používa patches o veľkosti 16×16. Použitím menších patchov uprednostňuje úlohu hustej predikcie. Tieto patche sa potom použijú ako vstup do hierarchického kodéra transformátorov na získanie viacúrovňových funkcií 1/4, 1/8, 1/16, 1/32 pôvodného rozlíšenia obrázka. Následne sa tieto viacúrovňové funkcie odovzdajú All-MLP dekodéru na predpovedanie segmentačnej masky pri $\frac{H}{4} \times \frac{W}{4} \times N_{cls}$ rozlíšenia, kde N_{cls} je počet kategórií pre segmentáciu [39].



Obr. 3.16: Model architektúry SegFormer [39]

3.6.1 Hierarchický kodér transformátorov

Vedci SegFormeru vytvorili viacero kodérov s rozličnými veľkosťami. Označili ich skratkou MiT (Mix Transformer encoders). Kodéri sú rozdelené od MiT-B0 do MiT-B5. Pričom MiT-B0 značí najľahší model pre rýchle odvodenie a MiT-B5 značí ich najväčší model pre najlepší výkon. Model MiT je čiastočné inšpirovaný ViT-om, ale je prispôsobený a optimalizovaný pre sémenatickú segmentáciu [39].

Hierarchické znázornenie funkcií

Oproti ViT, SegFormer sa snaží generovať pre vstupný obrázok viacúrovňové funkcie podobné konvolučným neuronovým sieťam. ViT dokáže generovať iba funkcie s jedným rozlíšením máp. Tieto funkcie poskytujú hrubé funkcie s vysokým rozlíšením a jemné funkcie s nízkym rozlíšením, ktoré zvyčajne zlepšujú výkon sémantickej segmentácie. Na vstupný obrázok s rozlíšením $H \times W \times 3$ sa vykonáva zlučovanie skupín oblastí (angl. patch merging) pre získane hierarchickej funkčnej mapy F_i, s rozlíšením $\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i$ kde $i \in 1, 2, 3, 4$ a C_{i+1} je väčšie ako C_i [39].

Zlúčenie prekrývajúcich sa opráv

ViT pre vstupné obrázkové patche zjednocuje $N \times N \times 3$ patch do $1 \times 1 \times C$ vektora. Tento proces spájania patchov, môže byť jednoducho rozšírený z $2 \times 2 \times C_i$ patchov do $1 \times 1 \times C_{i+1}$ vektora na získanie hierarchických máp prvkov. Pomocou toho sa môžu zmenšiť hierarchické vlastnosti z $F_1(\frac{H}{4} \times \frac{W}{4} \times C_1)$ do $F_2(\frac{H}{8} \times \frac{W}{8} \times C_2)$ a takto následne iterovať pre akúkoľvek inú mapu prvkov v hierarchií. Tento proces bol pôvodne navrhnutý tak, aby kombinoval neprekrývajúce sa opravy obrázkov alebo funkcií. Preto nedokáže zachovať lokálnu kontinuitu okolo týchto záplat. Namiesto toho sa používa proces spájania prekrývajúcich sa záplat. Z týmto účelom autori definovali K, S a P, kde K je veľkosť patchu. S je krok medzi dvoma susednými políčkami a P je veľkosť výplne [39].

Efektívna Self-Attention

Hlavnou výpočtovou prekážkou kodéra je vrstva self-attention. V pôvodnom multi-head self-attention procese má každá z hláv Q, K, V rovnaké rozmery $N \times C$, kde $N = H \times W$ je dĺžka sekvencie, self-attention sa odhaduje ako:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_{head}}})V. \tag{3.7}$$

Výpočtová zložitosť toho procesu je $O(N^2)$, čo je prekážkou pre veľké rozlíšenia obrazov. Namiesto toho autori používajú proces redukcie sekvencie zavedený v [37]. Tento proces používa redukčný pomer R na skrátenie dĺžky sekvencie následovne:

$$\hat{K} = Reshape(\frac{N}{R}, C \cdot R)(K) \tag{3.8}$$

$$K \ = Linear(C \cdot R, C)(\hat{K}), \tag{3.9}$$

kde K je sekvencia, ktorá má byť redukovaná, $Reshape(\frac{N}{R}, C \cdot R)(K)$ odkazuje na preformátovanie K na taký tvar, ktorý ma tvar $\frac{N}{R} \times (C \cdot R)$ a $Linear(C_{in}, C_{out})$ odkazuje na lineárnu vrstvu, ktorá berie C_{in} rozmerný tenzor ako vstup a generuje C_{out} rozmerný tenzor ako výstup. Z toho dôvodu má nové K rozmer $\frac{N}{R} \times C$. V dôsledku toho je zložitosť self-attention zredukovaná z $O(N^2)$ na $O(\frac{N^2}{R})$. Autori vo svojich experimentoch nastavili R na [64, 16, 4, 1] pre štádia 1 až 4 [39].

Mix-FFN

ViT používa kódovanie pozície na zavedenie informácií o polohe. Rozlíšenie koódovanie pozície je však fixné. Preto ak, je rozlíšenie testovania odlišné od trénovacieho, kód pozície musí byť interpolovaný a to často k zníženej presnosti. Autori vytvorili Mix-FFN, ktorý zohľadňuje účinok nulového paddingu na unikanie informácií o polohe pomocou 3×3 konvolúcie v doprednej sieti. Mix-FFN môže byť formulovaný ako:

$$\mathbf{X}_{out} = MLP(GELU(Conv_{3\times3}(MLP(\mathbf{x}_{in})))) + \mathbf{x}_{in}, \tag{3.10}$$

kde $\mathbf{x}_{in}$ je funkcia z modulu self-attention. Mix-FFN mieša konvolúciu 3×3 a MLP do každej FFN [39].

3.6.2 Ľahký All-MLP dekodér

SegFormer používa ľahký dekodér pozostávajúci len z MLP vrstiev a tak sa vyhýba klasickým a výpočtovo náročným komponentom používaným v iných metódach. Kľúčom k umožneniu takejto jednoduchosti dekodéra je to, že náš hierarchický kodér transformátorov má väčšie efektívne pole pôsobnosti ako tradičné CNN kodéry. Autormi navrhnutý All-MLP dekodér sa skladá zo štyroch hlavných krokov. Najprv viacúrovňové funkcie F_i kodéra z MiT predchádzajú MLP vrstvou na zjednotenie kanálovej dimenzie. Potom sa v druhom kroku funkcie zväčšia na štvrtinovú veľkosť a konkatenujú sa spolu. Tretím krokom je použitie MLP vrstvy na zlúčenie konkatenovaných funkcií F. Nakoniec sa ďalšou MLP vrstvou

vezme zlúčená funkcia na predikciu segmentačnej masky M s $\frac{H}{4} \times \frac{W}{4} \times N_{cls}$ rozlíšenia, kde N_{cls} je počet kategórií. Dekodér je následovne formulovaný takto:

$$\hat{F}_i = Linear(C_i, C)(F_i), \forall i \tag{3.11}$$

$$\hat{F}_i = Upsample(\frac{W}{4} \times \frac{W}{4})(\hat{F}_i), \forall i \tag{3.12}$$

$$F = Linear(4C, C)(Concat(\hat{F}_i)), \forall i \tag{3.13}$$

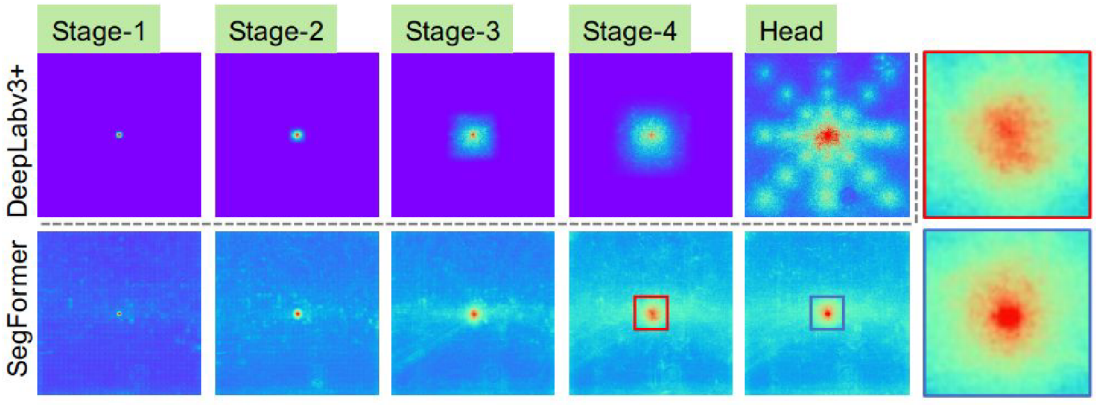
$$M = Linear(C, N_{cls})(F), \tag{3.14}$$

kde M označuje predpovedanú masku a $Linear(C_{in}, C_{out})(\cdot)$ označuje lineárnu vrstvu so vstupom C_{in} a výstupom C_{out}, ktoré majú vektorové rozmery [39].

Efektívna analýza vnímavého poľa

V úlohách sémantickej segmentácie efektívna analýza vnímavého poľa bola ústredným problémom pre udržanie veľkého receptívneho poľa. Autori architektúru SegFormer použili efektívne receptívne pole (ERF). Na obrázku 3.17 autori ukázali vizualizáciu ERF zo štyroch kodérov štádia a dekodéra hlavy pre oba DeepLabv3+ a SegFormer. Z pozorovania sa môžu vyvodiť tieto poznatky:

- ERF DeepLabv3+ je aj v najhlbšej fáze (Stage-4) pomerne malý.
- Segmentačný model SegFormer má kodér, ktorý prirodzene produkuje lokálne pozornosti, ktoré sa podobajú konvolúciám na nižších úrovniach, ale dokáže tiež vypúšťať vysoko nesúvisiace pozornosti, ktoré účinne zachytávajú kontexty na najvyššej úrovni.
- Ako je znázornené na približených patches na obrázku 3.17 ERF hlavy MPL (modrý rámček) sa líši od štádia 4 (červený rámček) s výrazne silnejšou lokálnou attention okrem ne-lokálnej attention [39].



Obr. 3.17: Efektívne receptívne pole na Cityscapes datasete [39]

Autori túto architektúru trénovali na dostupných datasetov: Cityscapes [5], ADE20K [42] a COCO-Stuff [4]. Dataset Cityscapes obsahuje 5000 detailných anotovaných obrázkov s vysokým rozlíšením s 19 kategóriami. Je to dataset z pohľadu jazdiaceho auto. ADE20K

Veľkosť kodéra	Parametre		ADE20K		Cityscapes		COCO-stuff	
	Kodér	Dekodér	Flops	mIoU(SS/MS)	Flops	mIoU(SS/MS)	Flops	mIoU(SS)
MiT-B0	3.4	0.4	8.4	37.4/38.0	125.5	76.2/78.1	8.4	35.6
MiT-B1	13.1	0.6	15.9	42.2/43.1	243.7	78.5/80.0	15.9	40.2
MiT-B2	24.2	3.3	62.4	46.5/47.5	717.1	81.0/82.2	62.4	44.6
MiT-B3	44.0	3.3	79.0	49.4/50.0	962.9	81.7/83.3	79.0	45.5
MiT-B4	60.8	3.3	95.7	50.3/51.1	1240.6	82.3/83.9	95.7	46.5
MiT-B5	81.4	3.3	183.3	51.0/51.8	1460.4	82.4/84.0	111.6	46.7

Tabuľka 3.1: Tabuľka zobrazujúca veľkosti, dizajnu kodéra a dekodéra

je sada pre analýzu scény, ktorá zahrňuje 150 jemne šturktúrovaných sémantických konceptov a obsahuje 20 210 obrázkov. COCO-Stuff je robustný dataset, ktorý obsahuje 172 klasifikačných tried. Tento dataset obsahuje dokopy 164 tisíc obrázkov z toho 118 tisíc pre trénovanie a 5 tisíc pre validáciu. Autori vo svojej práci experimentovali s veľkosťou kodéra MiT. Čím bol väčší kodér MiT, tým bola metrika mIoU lepšia na datasetoch. Tabuľka 3.1 zobrazuje ako sa zmenia metriky pri zmene veľkosti parametrov. Porovnanie je na 3 datasetch spomínaných vyššie. Metriky ktoré sa zobrazujú je flops a mIoU (SS/MS). Flops je odvodená z anglického *floating point operations per seconds* to znamená, že táto metrika opisuje koľko operácií je potrebných na spustenie jednej inštancie modelu. Metrika mIoU je priemer IoU segmentovaných objektov na všetkých obrázkov pri trénovaní alebo testovaní. Ide o základnú metriku, ktorá sa používa pri sémantickej segmentácií. Jej skratka je z anglického *Intersection over Union*. IoU spočíva v podiele plochy prekrývajúcej sa medzi predikovanou maskou a skutočnou maskou (tj. oblasť, ktorá je zdieľaná oboma maskami) a plochy zjednotenia oboch masiek. Vypočíta sa ako:

$$IoU = \frac{Area\,prekrytia}{Area\,zjednotenia} \tag{3.15}$$

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.16}$$

3.6.3 Vzťah s architektúrou SETR

SegFormer obsahuje viacero efektívnejších a výkonnejších dizajnov v porovnaní so SETR:

- SegFormer používa ia ImageNet-1k pre predtrénovanie. ViT v architektúre SETR je pretrénovaný na robustnejšej dátovej sade ImageNet-22k.
- Kodér SegFormeru má hierarchickú architektúru, ktorá je menšia ako ViT. Z toho dôvodu dokáže zachytiť hrubé funkcie s vysokým rozlíšením aj jemné funkcie s nízkym rozlíšením. Oproti tomu ViT kéder v SETR dokáže generovať iba jednu mapu prvkov s nízkym rozlíšením.
- SegFormer odstraňuje positional embeddings v kodéri. SETR používa positional embedding s pevným tvarom, čo vedie k zníženiu presnosti, keď sa rozlíšenie počas odhadu líši od trénovacieho rozlíšenia.

- SegFormer MLP dekóder je viac kompaktný a menej výpočtovo náročný ako dekodér v SETR. To vedie k zanedbateľnej výpočtovej réžií. Naproti tomu SETR vyžaduje ťažké dekodéry s viacnásobnými 3×3 konvolúciami [39].

3.7 Zhrnutie

Táto kapitola sa venovala sémantickej segmentácií. V úvode kapitoly 3.1 bolo vysvetlené čo je to sémantická segmentácia a že poznáme tri druhy a to menovite: sémantická segmentácia, inštančná segmentácia a panoptická segmentácia. Následne pre každú z kategórií 3.2, 3.3, 3.4 boli vybrané a popísané existujúce architektúry, ktoré spadajú do danej kategórií.

V nasledujúcej podkapitole 3.5 je podrobnejšie popísaná architektúra SETR. Táto architektúra sa snaží o iný pohľad na sémantickú segmentáciu za použitia transformátorov. Nadväzujúca podkapitola 3.6 opisuje architektúru SegFormer. Táto architektúra vychádza práve z architektúry SETR, ktorú vylepšuje. Výsledkom je použiteľný framework, ktorý sa dá jednoducho použiť na natrénovanie sémantickej segmentácie za pomoci transformátorov.

Kapitola 4

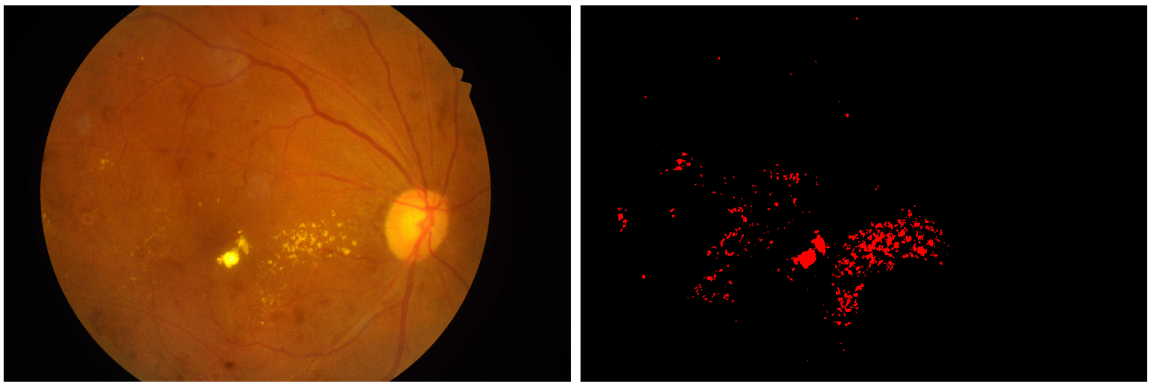
Návrh

Táto práca sa venuje natrénovaniu sémantickej segmentácie pre patologické nálezy na sietnici oka. Mojou snahou je natrénovať architektúry, ktoré sa venujú sémantickej segmentácie. Pre úspešné natrénovanie sietí je potrebné mať dátovú sadu s ktorou sa bude pracovať. Jednou z úloh je nájsť alebo vytvoriť vhodný dataset. Následne je dôležité vybrať správne technológie. V rámci tejto práce sa snažím natrénovať dve architektúry konkrétne U-Net a SegFormer. V počiatku sa sústredím na natrénovanie architektúry U-Net, lebo táto architektúra existuje dlhšie a je určená práve pre sémantickú segmentáciu lekárskych nálezov. U-Net vie fungovať v binárnej segmentácií, ale aj v multiclass segmentácií. Na začiatku sa sústredím na natrénovanie binárnej segmentácie, kde budem rozpoznávať iba jeden patologický nález. Model SegFormer je však multiclass segmentáciou, to znamená, že je určený na rozpoznávanie viac tried v rámci obrázka. Z toho dôvodu po natrénovaní binárnej U-Net siete sa pokúsim natrénovať multiclass U-Net. Následne sa natrénuje architektúra SegFormer na rovnakom datasete ako U-Net. Porovnajú sa výsledky a predikcie jednotlivých sietí. V ďalšej časti je popísaný dataset, ktorý bol použitý pre túto prácu a postup práce.

4.1 Dátová sada

Na úspešné natrénovanie neurónových sietí je všeobecne známe, že treba mať kvalitnú dátovú sadu. Rovnako je to aj v tomto prípade. V rámci sietnice oka, existuje mnoho dostupných datasetov. Predovšetkým čistých obrázkov, ktoré obsahujú iba sietnicu ľudského oka. Táto diplomová práca sa venuje sémantickej segmentácií. Z tohoto dôvodu je potrebne mať anotovanú dátovú sadu. Pod pojmom anotovaná dátová sada sa rozumie, že ku príslušnému obrázka sietnice oka patrí aj maska na ktorej je zobrazený patologický nález, ktorý nás zaujíma. Z toho dôvodu sa jedná o komplikovanejšiu úlohu, lebo už nie sú voľno dostupné dátové sady. Anotovanie dát svojpomocne sa v tomto prípade nemohlo udiať, lebo ide o medicínske údaje. Nevedel by som presne určiť, či sa jedná o určitý patologický nález na obrázku, napríklad či sa na sietnici oka nachádza exudát alebo drúza. V tomto prípade to musí určiť odborník - oftalmológ. Vo svojom meste som nenašiel očného lekára, ktorý by mal čas anotovať dáta, alebo mi poskytnúť obrázky. Z voľno dostupných databáz som si vybral databázu IDRiD [27]. Táto dátová sada je veľmi kvalitná a obsahuje segmentačné masky na trénovanie. Konkrétne databáza [27] obsahuje dokopy 81 obrázkov sietnice oka, ktoré sú rozdelené na 54 obrázkov pre trénovanie a 27 testovanie. Táto databáza obsahuje segmentačné masky pre 4 patologické nálezy a to konkrétne pre: mikroaneurizmy, hemoragiu, tvrdé exudáty, mäkké exudáty. Piata segmentačná maska je pre optický disk. V tomto

prípade ja som sa zameral na tvrdé a mäkké exudáty. Obrázky boli veľmi kvalitné v rozlíšení 4288 x 2848. Obrázok 4.1 zobrazuje jeden obrázok z dátovej sady IDRiD, kde vľavo je vstupný obrázok sietnice oka a vpravo príslušná maska zobrazujúca tvrdé exudáty.



Obr. 4.1: Ukážka datasetu IDRiD, vľavo originál obrázok, vpravo maska pre tvrdé exudáty

Nepodarilo sa mi nájsť dataset ktorý obsahuje drúzy. Po dohode s vedúcim práce sme sa dohodli, že budem pracovať na datasete IDRiD. Je to aj z dôvodu toho, že drúzy a exudáty sú si veľmi podobné. Na obrázkoch sietnice oka sa obidva patologické nálezy zobrazujú v žltkastej farbe. Rovnako sa drúzy aj exudáty rozdeľujú na mäkké a tvrdé. Podrobnejšie rozdiely boli spomínané v kapitole 2.3.

4.2 Postup práce

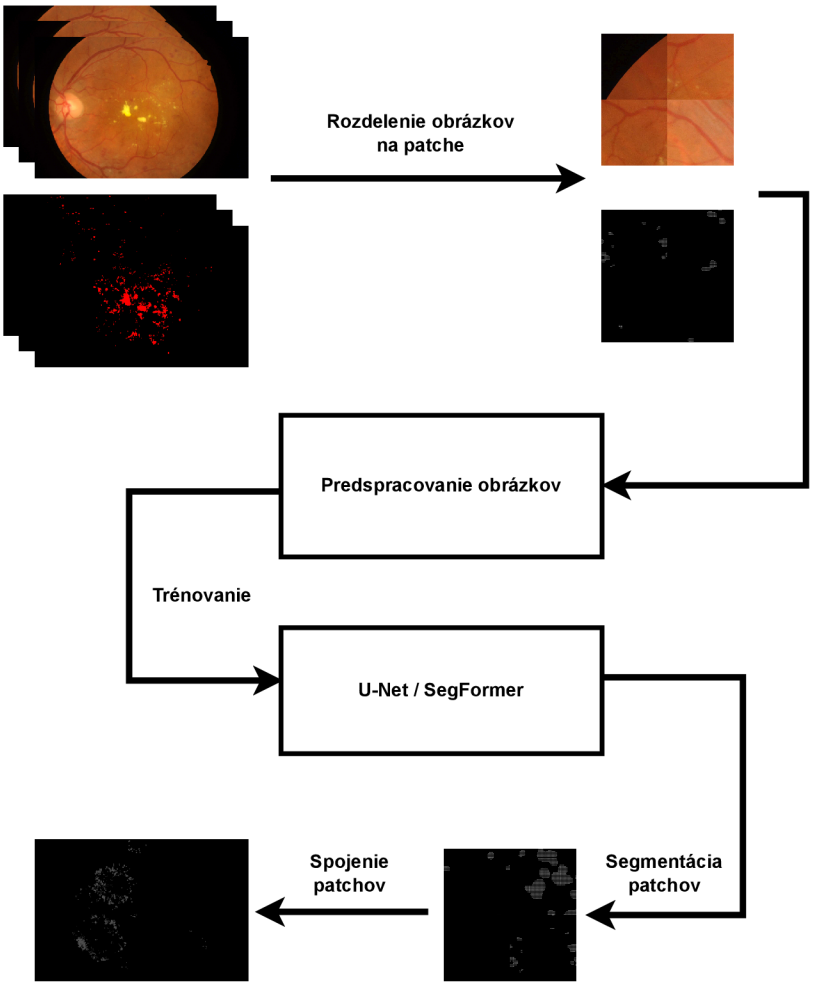
Po vybraní dátovej sady v rámci návrhu treba vybrať technológie. Ja som sa zameral na využitie programovacieho jazyka python a knižnice tensorflow. Programovací jazyk python ponúka mnoho užitočných knižníc na prácu s dátami, obrázkami alebo neurónovými sieťami. Ja využijem na prácu technológiu jupyter notebook. Tento notebook slúži k rýchlemu písania časti kódu do jednotlivých buniek. Každá bunka sa vie zvlášť spustiť a podať výsledok. Takto sa dá rýchlo implementovať a vzápätí kontrolovať výstup.

Na úspešné natrénovanie neurónových sietí, treba korektné spracovanie obrázkov a ich úprava. Dátová sada obsahuje veľmi kvalitné obrázky vo vysokom rozlíšení. Takýto veľký obrázok nie je vhodný na trénovanie. Z toho dôvodu na začiatku navrhujem jednotlivé vstupné obrázky rozdeliť na menšie obrázky, ktoré budú vhodné na trénovanie.

Následne sa vytvorí vhodný dáta generátor, ktorý zoberie vstupné obrázky z ich maskami a vytvorí batche, ktoré sa budú trénovať. Tento generátor v sebe bude aj obsahovať aj CLAHE techniku spracovania obrázkov. CLAHE je skratka z anglického výrazu *Contrast Limited Adaptive Histogram Equalization*. CLAHE zlepšuje ekvalizáciu histogramu prispôsobením prerozdelenia hodnôt pixelov miestnemu obsahu obrázka. To sa dosiahne rozdelením obrázka na malé, prekrývajúce sa bloky a následným použitím ekvalizácie histogramu na každý blok jednotlivo. Aby sa však zabránilo nadmernému vylepšeniu, CLAHE aplikuje limit kontrastu na každý blok, ktorý zabraňuje prílišnej redistribúcii hodnôt pixelov. Výsledok je obraz s vylepšeným kontrastom a datailami, pričom sa stále zachováva prirodzený vzhľad. Táto technika sa často používa pri lekárskych záznamoch, z toho dôvodu to aplikujem na môj dataset [21]. Následne vyskúšam či dáta generátor funguje správne a tvorí dobré batche.

Na vytvorenie architektúry využijem vhodnú knižnicu. V začiatku sa budem snažiť hlavne natrénovať architektúru U-Net. Prvé vyskúšam binárnu sémantickú segmentáciu a potom multiclass segmentáciu. V rámci metriky budem pracovať predovšetkým s metrikou IoU. V rámci trénovania treba experimentovať s loss funkciou, optimizerom a learning rate. Optimizer, ktorý som si vybral je Adam s learning rate 0.0001. V rámci loss funkcie treba rozlišovať či sa jedná o binárnu segmentáciu alebo multiclass segmentáciu. Pre binárnu segmentáciu použijem loss funkciu binary_crossentropy. Pre multiclass segmentáciu budem pracovať so sparse _categorical _crossentropy.

Dataset obsahuje masky zvlášť pre každú patológiu. Multiclass segmentácia potrebuje kombináciu týchto másk dokopy. Po natrénovaní binárnej segmentácie tieto masky spojím, aby každá patológia mala unikátne označenie (farbu) vrámci masky. Natrénujem multiclass segmentáciou na modely U-Net a neskôr vyskúšam na rovnakom datasete natrénovať model SegFormer. Obrázok 4.2 zobrazuje návrh práce.



Obr. 4.2: Schéma návrhu práce

Kapitola 5

Implementácia a experimenty

Táto kapitola popisuje implementáciu a postupy ako som riešil túto prácu. Na začiatku som musel správne spracovať dátovú sadu. Následne som si zobral architektúru U-Net a na nej natrénoval binárnu sémantickú segmentáciu. V počiatku sa nedarilo úspešné natrénovať sieť. Kapitola popisuje, aké problémy vznikli v priebehu riešenia práce a následná snaha ich odstrániť. Po binárnej segmentácií som sa snažil o multiclass klasifikáciu patologických nálezov. Rovnako som pracoval s U-Net architektúrou. Následne som prešiel na vyskúšanie modelu SegFormer.

5.1 Implementácia

Implementáciu som robil v jazyku python za pomoci tensorflow keras knižníc. V rámci architektúr neurónových sietí som použil knižnicu [12]. V tejto knižnici sa nachádzajú segmentačné modely neurónových sietí ako je napríklad U-net, FPN, Linknet a PSPNet. V tejto knižnici sa nachádzajú aj *Backbones*, ktoré sú natrénované na dátovej sade imagenet [7]. Medzi implementovanými Backbones sú napríklad architektúry: VGG, ResNet, Inception a pod. Mojim cieľom bolo na rovnakej databáze natrénovať architektúru U-Net a SegFormer. Následne porovnať výsledky a zhodnotiť či sa nový pohľad na sémantickú segmentáciu hodí pre rozpoznanie patologických nálezov na sietnici oka. Rovnako má táto knižnica implementované loss funkcie (Jaccard, Dice, Focal) a metriky (IoU, F-score). V trénovania a testovanie architektúr neurónových sietí som trénoval na svojom osobnom počítače, konkrétne na špecifikácií: GPU: Nvidia Rtx 3060 ti, CPU: AMD Ryzen 5 5600x, RAM: 16GB. Na začiatku som vyskúšal architektúru U-Net. U-Net je zaužívanou architektúrou, ktorá sa používa práve v medicíne pre segmentáciu patologických nálezov.

5.2 U-Net

Ako bolo spomínané vyššie pracoval som s databázou IDRiD [27]. Databáza obsahovala obrázky vo veľmi vysokom rozlíšení. U-Net bol navrhnutý pre veľkosť vstupného obrázka 512×512. Aj z toho dôvodu originálne obrázky z databáze IDRiD nemohli byť použité. Ďalším dôvodom by bolo že, grafická karta by nevedela alokovať toľko miesta na grafickej karte. Veľkosť originálnych obrázkov bola 4288×2848. Ja som sa rozhodol každý vstupný obrázok rozrezať na určitú veľkosť a vznikli patche daných obrázkov. Využil som na to knižnicu *patchify*, ktorej parametre sú veľkosť obrázka a krok o ktorý sa ma posunúť po orezaní. Ja som sa rozhodol vstupné obrázky rozdeliť v rozmeroch 512×512, 256×256,

160 × 160, 128 × 128 a 64 × 64.Obrázok 5.1 zobrazuje ukážku ako jedno miesto obrázka je rozrezané na určitú veľkosť.

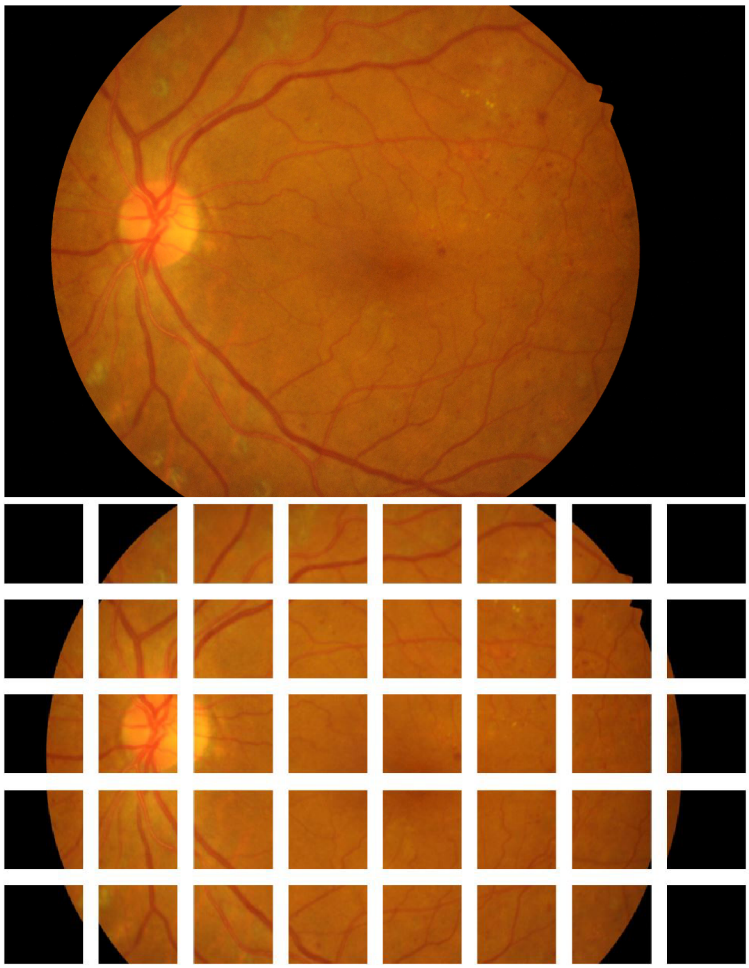
Obr. 5.1: Veľkosť obrázkov po rozdelení pôvodného obrázka: zľava 512x512, 256x256, 128x128

Rozdelil som to preto na toľko možných častí lebo v prvom kroku som chcel experimentovať a zistiť ktorá veľkosť funguje najlepšie na natrénovanie. Rovnako som rozdelil aj jednotlivé masky. Týmto rozdelením sa aj rozšírila dátová sada. Pre veľkosť obrázkov 512 × 512 bolo dokopy pre trénovanie 2160 obrázkov. Pôvodne ich bolo 54. Tabuľka 5.1 zobrazuje ako sa zväčšila dátová sada po rozdelení na menšie obrázky.

Rozmer obrázka	**Veľkosť dátovej sady**
512x512	2160
256x256	9504
160x160	23 868
128x128	39 204
64x64	159 192

Tabuľka 5.1: Zobrazuje veľkosť dátovej sade pri určitých rozmeroch obrázkov

Obrázok zobrazuje vizuálne ako sa rozdelí počiatočný obrázok na menšie obrázky o rovnakej veľkosti 5.2.

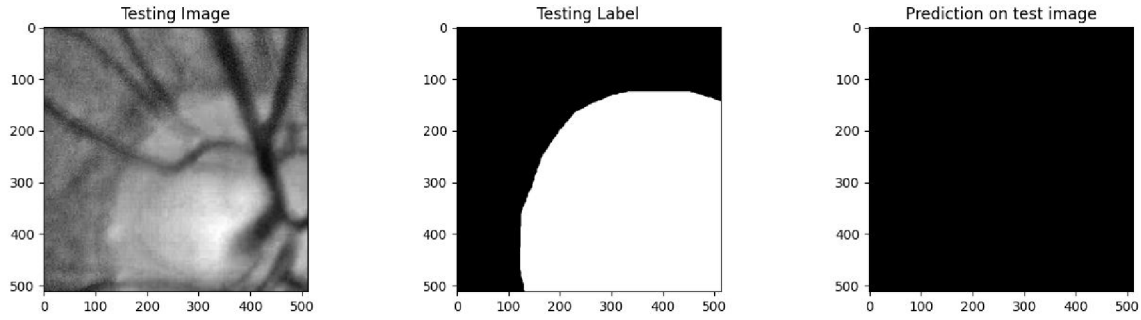


Obr. 5.2: Ukážka rozdelenia vstupného obrázka na obrázky o veľkosti 512 × 512

5.2.1 Binárna segmentácia

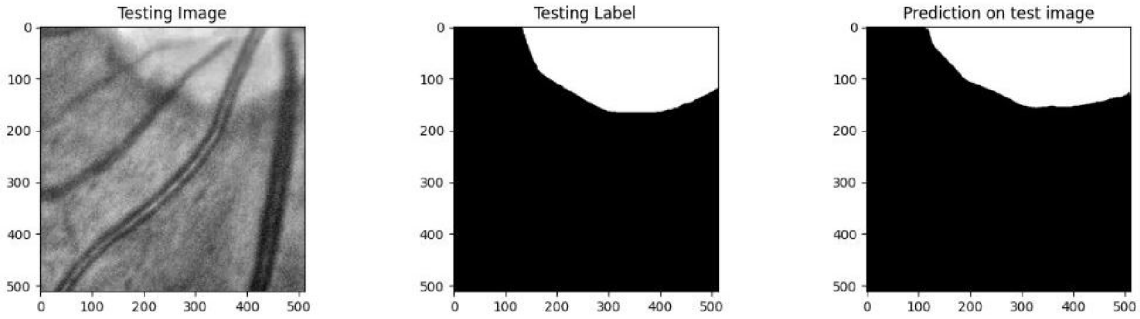
Po rozdelení obrázkov na menšie obrázky som sa prvotne pokúsil trénovať neurónovú sieť architektúry U-Net na exudátoch. Napísal som vlastný datagenerátor pre túto úlohu. V tomto trénovaní išlo o binárnu sémantickú segmentáciu, kde som sa snažil iba rozpoznať tvrdé exudáty. Ako backbone som používal resnet34. Pre každé trénovanie som rozdelil trénovací dataset 80% obrázkov pre trénovaciu sadu a 20% pre validačnú sadu. IDRiD obsahovala zvlášť 27 obrázkov pre testovanie sietí. V počiatku sa mi nepodarilo úspešne natrénovať U-Net sieť pre rozpoznanie exudátov. Skúšal som meniť loss funkcie, learning rate a metriky. Nič však nepomáhalo. Preto som v prvej fáze vyskúšal natrénovať rozpoznanie optického disku. IDRiD databáza v sebe obsahovala anotované dáta pre optický disk. Masky optického disku som rozdelil na veľkosti 512 × 512. Následne som sa snažil natrénovať U-Net na rozpoznávanie optického disku. Z počiatku bol tiež problém a sieť sa nedokázala učiť a vo výsledku nič predikovať. Experimentoval som s rôznymi loss funkciami a optimazermi. Podarilo sa mi nájsť príčinu zlého trénovania. Neoptimalizoval som výstupy masky. Maska bola binárna kde čierna farba znamenala pozadie a biela farba značila pozíciu objektu v tomto prípade to bol optický disk. Takto binárne by pre jeden pixel obrazu mala byť hodnota buď 0 čo značí čiernu farbu a žiadna patológia, alebo 1 čo značí bielu farbu a výskyt

patológie. V mojom prípade zo začiatku to tak nebolo a miesto jednotky som mal hodnotu 0.0039. Z toho dôvodu sa sieť netrénovala. Obrázok 5.3 zobrazuje predickou optického disku neurónovej siete po 50 epochách trénovania.



Obr. 5.3: Zlá predikcia optického disku

Túto chybu som lokalizoval a nastala pri tom, že som dvakrát delil obrázok hodnotou 255, aby som každý pixel normalizoval. Preto som to delil dvakrát, lebo pri vytvorení menších obrázkov z pôvodných veľkých obrázkov som už použil túto normalizáciu a delil som každý obrázok hodnotou 255. Hodnotou 255 sa to delilo preto, lebo každý pixel môže mať hodnotu od 0 do 256. Každá ta hodnota reprezentuje farebný kód. Takže predelením 255 dostaneme hodnotu od 0 do 1. Keďže ja som mal binárnu segmentáciu mal som práve hodnoty 0 pre pozadie a 1 pre patologický nález. Normalizuje sa to z dôvodu, že hlboké neurónové siete, ktoré pracujú z obrázkami sú numerické veľmi komplexné a trénovanie by bolo náročne a dlhšie. Po týchto zisteniach a úpravách sa mi podarilo natrénovať U-Net pre rozpoznávanie optického disku. Optimzer som použil Adam s learning rate 0.0001. Loss funkcia bola použitá binary_crossentropy. Ako metriku som použil IoU. Priemerná hodnota IoU tohoto modelu na sémantickú segmentáciu optického disku bola 0.8234. Obrázok 5.4 zobrazuje predikciu optického disku na natrénovanej U-Net sieti.



Obr. 5.4: Predikcia optického disku

Po úspešnom natrénovaní na optickom disku som sa mohol posunúť na sémantickú segmentáciu drúz. Tu som použil rovnakú architektúru a a spôsob trénovania ako pri optickom disku. Experimentoval som s rôznymi veľkosťami obrázkov. Všetky rozmery sa podarilo natrénovať. Trénoval som v rozpätí 100 až 200 epoch. Najdlhšie sa trénovali najmenšie obrázky. To bolo spôsobené tým, že ich bolo aj najviac. Na jednotlivé rozmery obrázkov som používal rôznu batch veľkosť pre trénovanie. Batch veľkosť som menil v závislosti toho

koľko zvládla moja grafická karta so svojou pamäťou. Používal som veľkosti 8, 16, 32, 64. Pre menší rozmer obrázkov som použil väčšiu batch veľkosť. Trénovacie data som stále rozdelil, ako 80% pre trénovaciu sadu a 20% pre validačnú sadu. Tabuľka 5.2 zobrazuje hodnoty po trénovaní pre jednotlivé rozmery obrázkov. Pre trénovanie binárnej sémantickej segmentácie som pridal metriku *precision* a *recall*.

Metrika precision je pomer medzi skutočne pozitívnymi a všetkými pozitívnymi predikciami. V tomto prípade sa berie jeden pixel, ktorý sa rozpozná či naozaj patrí do danej triedy alebo nie. V binárnej segmentácií to môže byť pozadia alebo patologický nález v tomto prípade to je exudát. Matematickým výrazom sa precision zapíše:

$$Precision = \frac{True\,positive}{True\,positive + False\,positive} \tag{5.1}$$

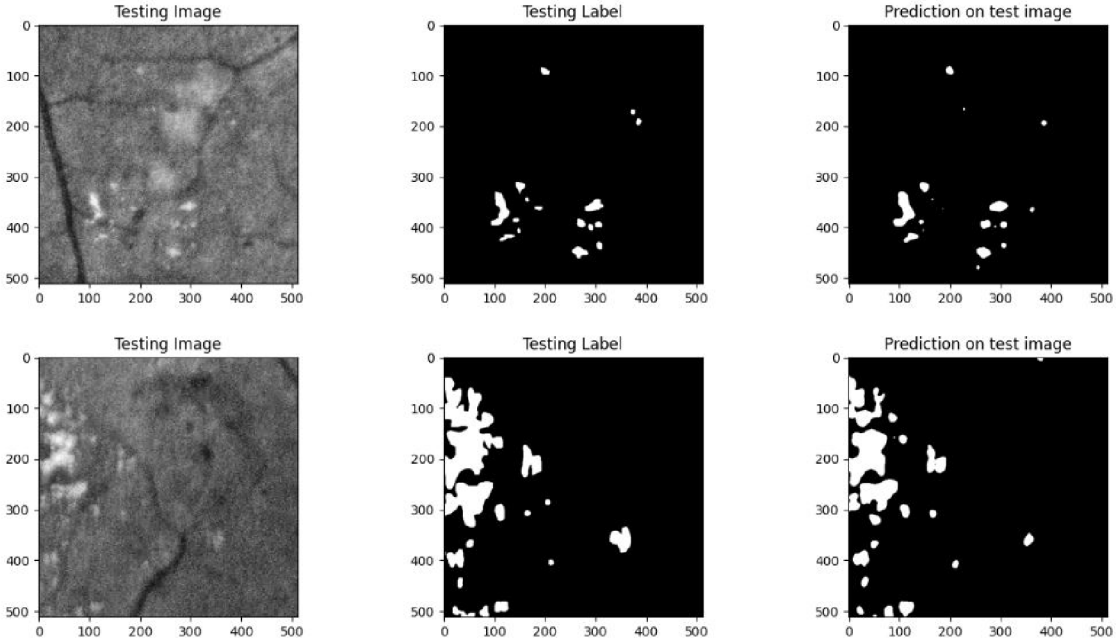
Metrika recall slúži odhalenie skutočných pozitívnych predikcií. V tomto prípade to hovorí o skutočnom dobrom odhade daního pixelu. Matematicky sa recall zapíše:

$$Recall = \frac{True\,positive}{True\,positive + False\,negative} \tag{5.2}$$

Input size (Epochs)	**IoU**	**Precision**	**Recall**	**Binary_IoU**	**Val_IoU**
512x512 (100)	0.8080	0.9357	0.9155	0.9335	0.4839
256x256 (150)	0.9567	0.9817	0.9787	0.9816	0.47350
128x128 (200)	0.9689	0.9882	0.9862	0.9880	0.47840
64x64 (200)	0.9764	0.9891	0.9878	0.9891	0.36940

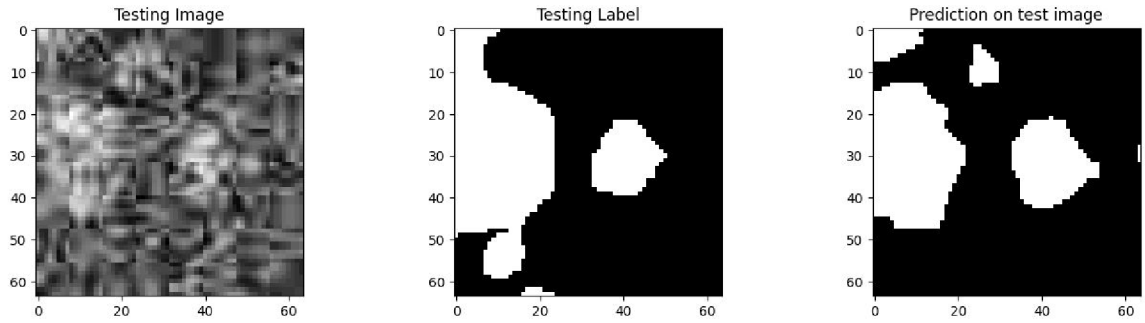
Tabuľka 5.2: Zobrazujúca metriky po natrénovaní pre špecificky rozmer obrázkov

Ako prvú veľkosť som trénoval 512×512. ukážka predikcie je na obrázku 5.5. Môžeme si všimnúť že predikcia siete bola na veľmi vysokej úrovni.



Obr. 5.5: Predikcia exudátov pre rozmer obrázka 512×512

Rovnakým spôsobom trénovania, som natrénoval zvyšné rozmery sietí. V prieme jedna epocha trénovania trvala do dvoch minút. Jedine pri trénovaní rozmeru 64×64 trénovanie bolo dlhšie a jedna epocha sa trénovala okolo 4 minút. Obrázok 5.6 zobrazuje predikciu o veľkosti vstupného obrázka 64×64. Tiež je vidieť že čím menšie rozlíšenie obrázka, tým je obrázok viac rozmazaný pre ľudské oko. To však nevadí v počítačovom videní. Na predikcií si môžeme všimnúť, ako presne sieť sa snažila predikovať. Je vidieť voľným okom, že viditeľné tvrdé exudáty označila veľmi dobré. Z môjho pohľadu v tejto predikcií, sieť označila exdutáty viac oddelene. Pri týchto dvoch obrázkoch originálnej masky a predikovanej masky by som povedal, že predikovaná maska je originálna. To je iba môj subjektívny názor.



Obr. 5.6: Predikcia exudátov pre rozmer obrázka 64×64

Po natrénovaní binárnej segmentácií tvrdých exudátov, som sa rozhodol v rámci rozšírenia natrénovať aj zvyšné patologické nálezy. IDRiD dataset v sebe obsahuje masky pre 4 patologické nálezy a masku pre rozpoznanie optického disku. Z predchádzajúceho trénovania tvrdých exudátov, kde som skúšal rôzne rozmery obrázkov, som teraz vybral iba

Patológia	IoU	Precision	Recall	Binary_IoU	Val_IoU
Tvrdé exudáty	0.9567	0.9817	0.9787	0.9816	0.4735
Mikroaneurizmy	0.9130	0.9590	0.9493	0.9583	0.1855
Hemoragia	0.9413	0.9803	0.9756	0.9793	0.1495
Mäkké exudáty	0.9146	0.9707	0.9646	0.9702	0.2926

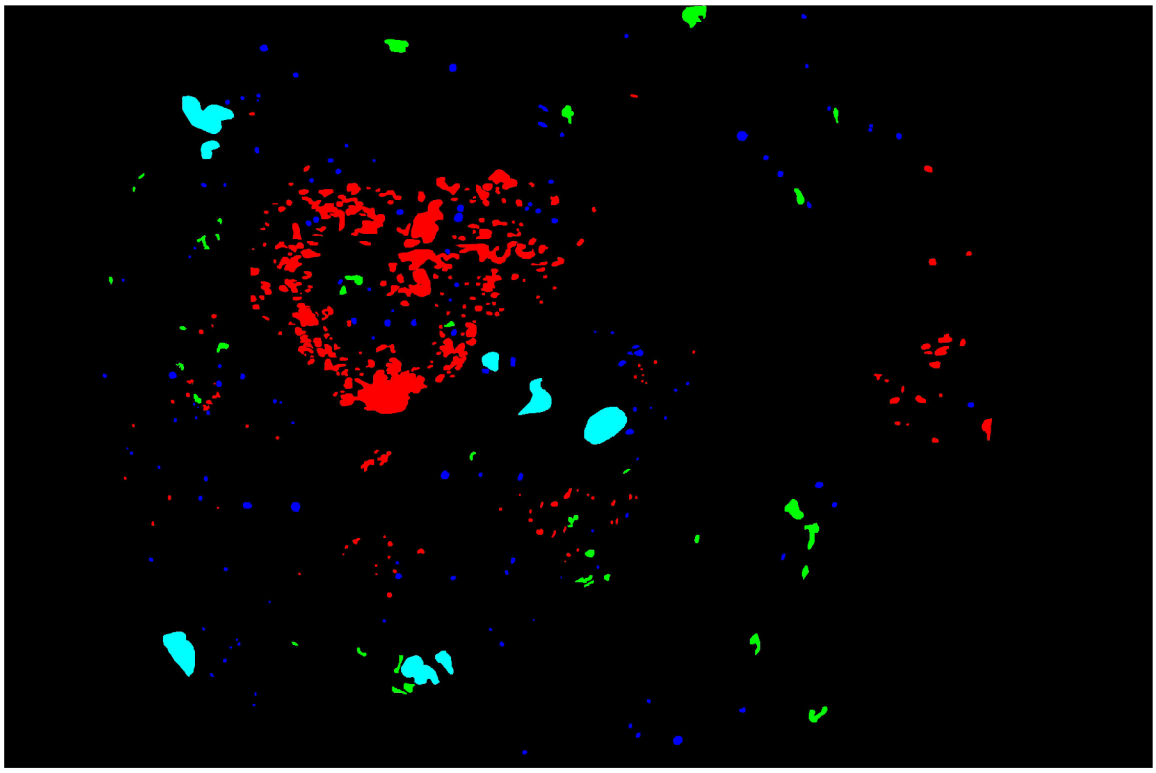
Tabuľka 5.3: Zobrazujúca metriky pre rôzne typy patológií

jednú veľkosť a natrénoval na jednej veľkosti segmentáciu. Vybral som si rozmery obrázkov 256×256. Tento rozmer som vybral z toho dôvodu, lebo bral som do úvahy rýchlosť trénovania a výsledné metriky po trénovaní. Pri tomto rozmere mi prišla dobrá cena presnosti za čas trénovania. Približne jedna epocha trénovania trvala 2 minúty.

IDRiD dataset v sebe pre trénovanie obsahoval 54 obrázkov. Lenže nie každý obrázok sietnice oka obsahoval všetky 4 typy patológií. Tvrdé exudáty a mikroaneurizmy sa nachádzali na všetkých 54 obrázkoch. Hemoragia bola zaznamenaná na 53 obrázkoch. Najmenším počtom výskytu boli mäkké exudáty, ktoré sa nachádzali iba na 26 obrázkoch z celkového počtu 54 obrázkoch. Po rozdelení obrázkov na rozmer 256×256 trénovacia sadat pre tvrdé exudáty a mikroaneurizmy obsahovala 9504 obrázkov. Hemoragia obsahovala 9328 obrázkov a mäkké exudáty obsahovala 4576 obrázkov. Tieto obrázky boli rozdelené ešte na trénovaciu a validačnú sadu. Trénovacia sada obsahovala 80% z celkového počtu obrázkov a validačná sada 20%. Výstup trénovania zobrazuje tabuľka 5.3. Tabuľka zobrazuje metriky pre každú patológiu. Tvrdé exudáty som mal natrénované z minulého experimentu. Ostatné patológie som trénoval na 100 epochách. Výsledky metrík sú veľmi podobné pre každú patológiu. Najväčší rozdiel je vidieť pre metriku IoU na validačnej sade.

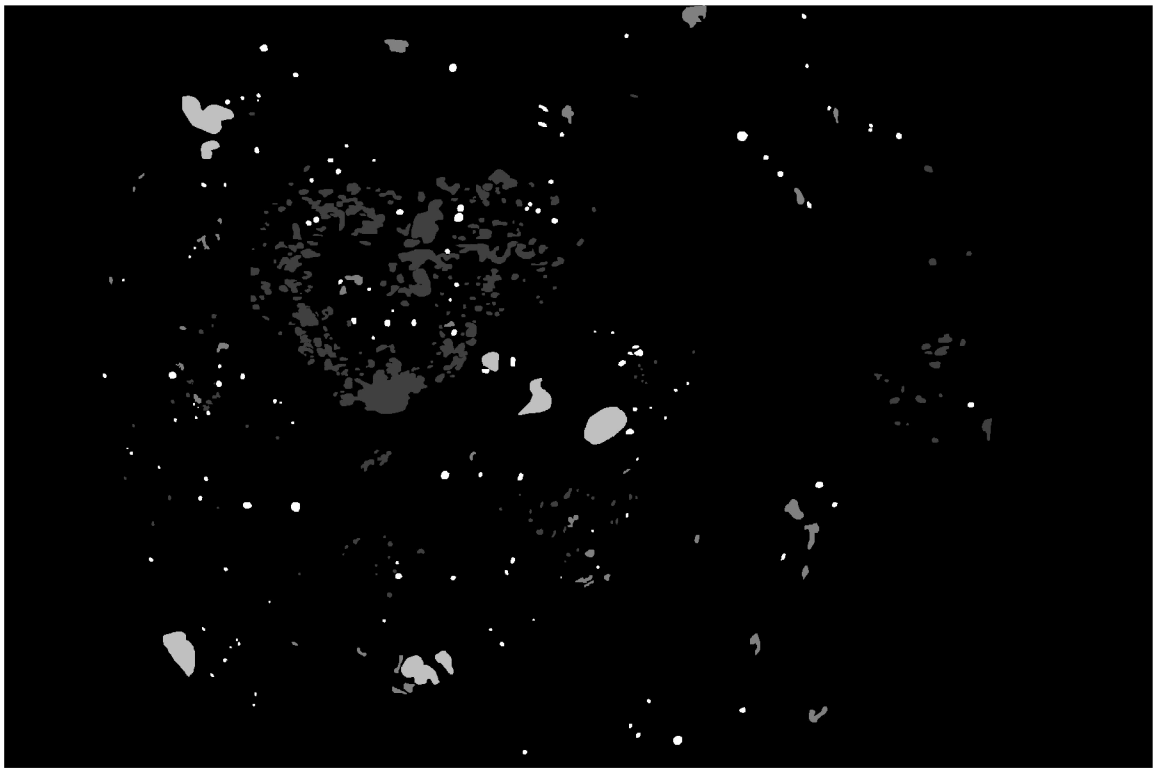
5.2.2 Multiclass segmentácia

Po natrénovaní binárnej sémantickej segmentácie exudátov som prešiel na viac triednu (z angl. *multiclass*) segmentáciu. Keďže IDRiD dataset neobsahoval v sebe drúzy. Pokúsil som sa nad rámec zadania segmentovať ostatné patologické nálezy, ktoré dataset IDRiD obsahuje. Chcel som natrénovať multiclass sémantickú segmentáciu na architektúre U-Net. IDRiD obsahuje dokopy 5 typov segmentačných masiek. Ja som nepoužil optický disk z toho dôvodu, že sa nejedná o patologický nález ale o časť sietnice. Preto som sa snažil rozpoznať: tvrdé a mäkké exudáty, mikroaneuryzmi a hemoragiu. Teraz išlo o komplikovanejšiu úlohu, ako pri binárnej segmentácií. Každý pixel mohol byť klasifikovaný do jednej z 5 tried. Pri binárnej segmentácií to bolo do jednej z 2 tried. Pred trénovaním takej siete bolo potrebné upraviť aj vstupné masky. Keďže dataset bol vytvorený tak, že každý patologický nález mal vlastnú binárnu masku, jednotlivé masky museli byť správne spojené. Spojil som masky dokopy. Každá patológia musela mať unikátnu farbu, aby sa medzi sebou rozlíšili. Tým sa aj docielilo to, že sieť vie rozlíšiť patológie a vie sa učiť na nich. Na začiatku som každej patológie dal veľmi jasnú farbu, kde som išiel podľa zloženia RGB. Tvrdé exudáty mali v RGB prevedení hodnotu $(255, 0, 0)$, hemoragia mala hodnotu $(0, 255, 0)$, mikroaneuryzmi mali hodnotu $(0, 0, 255)$ a mäkké exudáty mali hodnotu $(0, 255, 255)$. Pozadie, teda všetko ostatné čo nás nezaujímalo bolo čiernou farbou a preto to malo hodnotu $(0, 0, 0)$. Obrázok 5.7 zobrazuje, ako vyzerá výsledná maska spojením všetkých masiek dokopy.



Obr. 5.7: Výsledná maska spojením patologických masiek dokopy

Po analýze som zistil, že spojenie másk sa robí rozličným spôsobom. Datasety ich nerozlišujú v RGB spektre ale v intenzite, ako napríklad CIPH dataset [19]. Preto som znova všetky masky zobral a menil som intenzitu. Každá patológia mala inakšiu intenzitu. Finálna maska sa zobrazovala na monitoroch ako celá čierna. To je z dôvodu toho, že klasické programy na zobrazenie obrázkov pracujú v pôvodnom režime s maximálnou intenzitou čiže od 0 do 255. Súčasná kombinovaná maska ale mala intenzitu od 1 do 5. Preto obrázok 5.8 zobrazuje finálnu masku tak, hodnoty intenzity boli upravené aby zobrazovali od 0 do 5 a nie od 0 do 255. Rovnako ako v predchádzajúcej spojenej masky, každá patológia ma vlastnú hodnotu, pre správane určenie kategórie.



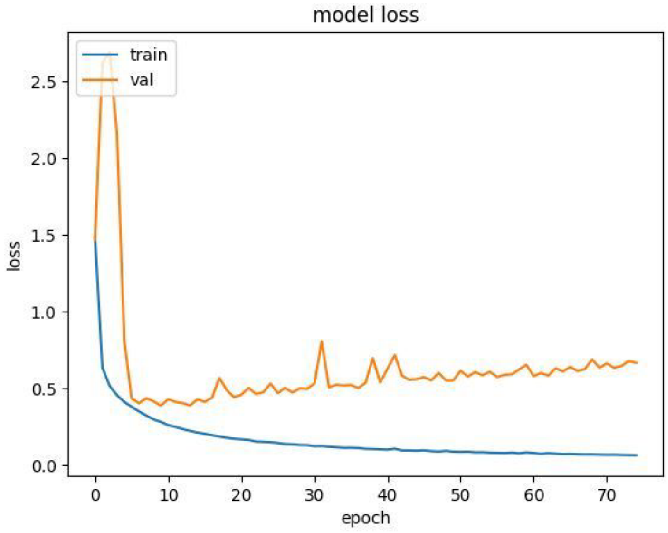
Obr. 5.8: Upravená kombinovaná maska s použitím intenzity

Následne s takýmito to maskami som rovnako rozdelil masky na menšie obrázky o špecifických veľkostiach napríklad o veľkosti 128×128. Použil som rovnaký datagenerátor aj architektúru, ktorú som použil pri binárnej sémantickej segmentácií. Narazil som však na problém, že datagenerátor správne nefungoval pre túto úlohu. Preto som sa inšpiroval oficiálnym návodom[1] na keras stránke. V tomto návode pracovali s datasetom *Oxford-IIIT Pet Dataset* [24]. Tento dataset obsahuje psy a mačky, ku ktorým je urobená segmentačná maska ich výskytu na obrázku. Masky tejto siete obsahuje 3 kategórie, kde jedna kategória je pozadia, druhá je obvod zvieraťa a tretia je telo zvieraťa. Sieť som upravil pre moju úlohu a to segmentáciu pre 5 kategórií. Pôvodné obrázky a masky som rozdelil na veľkosti 160×160. Bolo to z dôvodu toho, aby táto neurónová sieť pracovala so vstupom obrázkov o danej veľkosti. Jednalo sa rovnako o architektúru U-Net ale so štýlom siete Xception. Siete si boli podobné či sa jednalo o sieť pri binárnej segmentácií alebo pri tejto multiclass segmentácií. V počiatku sa mi nedarilo natrénovať sieť na mojich obrázkoch. Sieť nič nevedela predikovať. Po analýze so svojím vedúcim práce mi bolo doporučené, aby som prvé natrénoval túto sieť na inom datasete a následne do-trénoval na mojich obrázkoch sietnice oka. Bol mi doporučený dataset CIPH [19]. Tento dataset obsahuje 20 kategórií, ktoré rozpoznávajú inštancie ľudskej postavy, ale aj oblečenia. Dokopy dataset obsahuje 38,280 obrázkov. Obrázok 5.9 zobrazuje ukážku CIPH datasetu.

[1] $https://keras.io/examples/vision/oxford_pets_image_segmentation/$

Obr. 5.9: Ukážka obrázka spolu s jej maskou z CIPH datasetu

Ja som si z toho datasetu zobral 5000 obrázkov. Zobral som menší počet kvôli tomu, že jednotlivé obrázky som ešte orezal na požadovaný rozmer 160×160. Rozdelil som to preto, aby sa sieť naučila čo najlepšie na takom rozmere ako som mal pripravené obrázky sietnice oka. Týmto krokom sa mi zväčšila dátová sada na 24,880 obrázkov, ktoré som rozdelil rovnako 80% pre trénovanie a 20% na validovanie. Moja obrázky mali však 5 kategórií. Preto som si vybral zo CIPH datasetu 5 kategórií a natrénoval som sieť na redukovanom počte kategórií. Kategórie, ktoré som si vybral boli: čiapka, vlasy, tvár, šaty. Piata kategória bolo pozadie. Sieť som trénoval s optimizerom *RMSprop* a learning rate 0.001. Loss funkcia bola použitá sparse_categorical_crossentropy. Trénoval som 75 epoch túto sieť a jej priebeh je zobrazený na grafe 5.10.

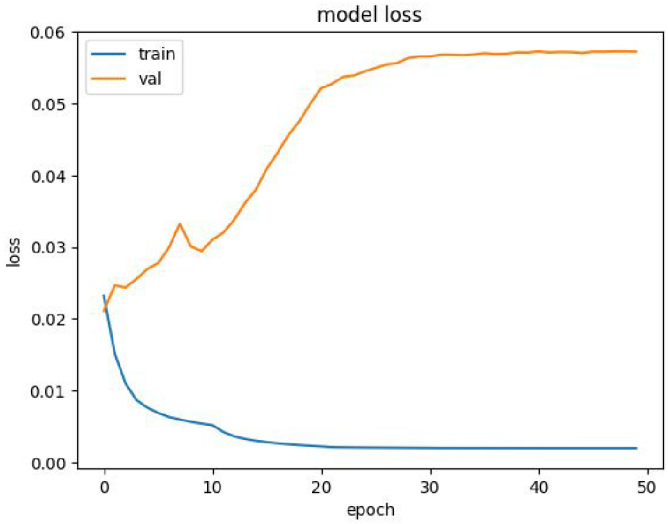


Obr. 5.10: Ukážka loss funkcie v priebehu trénovania

Následne som si odkontroloval predikcie siete s origiálnymi maskami a architektúra sa naučila veľmi dobre a predikuje 5 vybraných kategórií. Obrázok 5.11 zobrazuje vstupný obrázok a k nemu skutočnú masku a predikovanú masku.

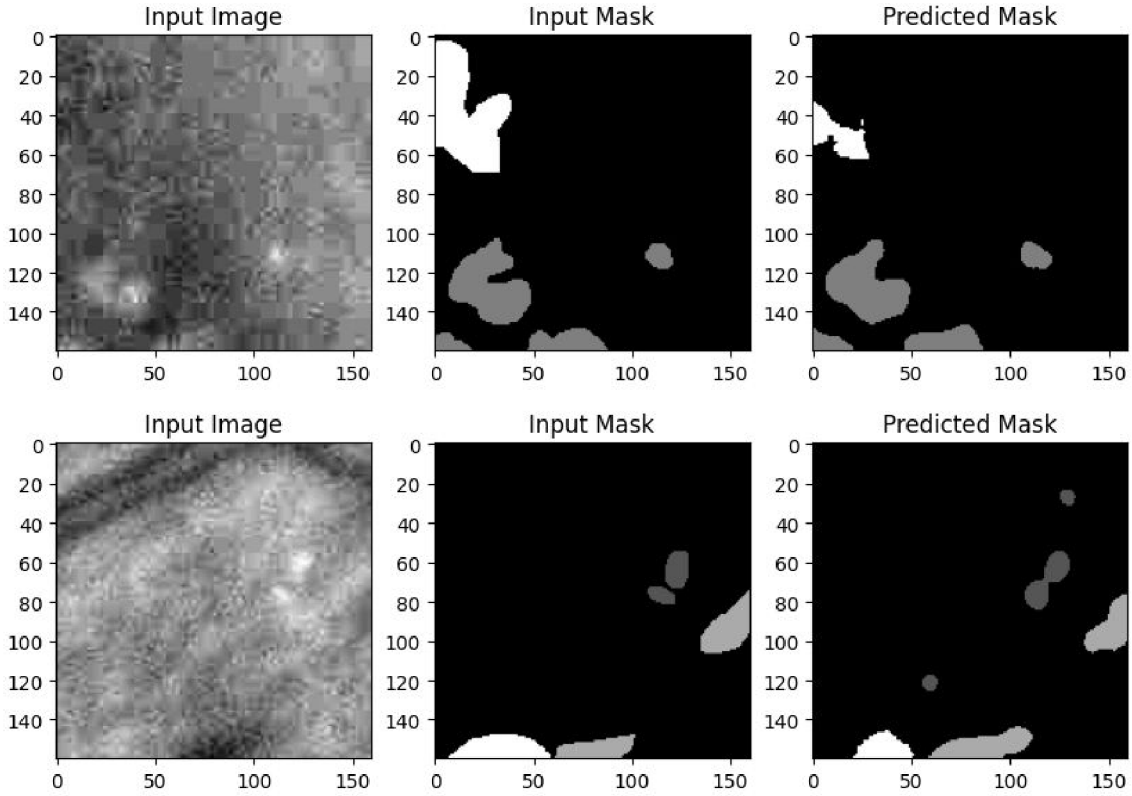
Obr. 5.11: Ukážka predikcie masky po natrénovaní na U-Net architektúre

Po natrénovaní na CIPH datasete som mal uložený model s naučenými váhami. Tento model som vzal a snažil som sa dotrénovať na svojom datasete multiclass IDRiD. Prvý pokus trénovania, spočíval v tom, že som nič na sieti nemenil, každú vrstvu som nechal tak ako bola. Stále som používal ako loss funkciu sparse_categorical_crossentropy spolu s optimizerom *RMSprop*, na ktorom bola nastavená learning rate 0.001. V binárnej sémantickej segmentácií som použil metriku binárnu IoU. V tomto prípade som sa snažil použiť metriku OneHotIoU. Táto metrika slúži na vypočítanie klasického IoU, ale na každú triedu, ktorú v úlohe rozpoznávame. Pri trénovaní, mi však stále dávala hodnotu 0 aj po 50 epochách. Konzultoval som to, ale neprišli sme na príčinu, prečo táto metrika nefunguje. Sieť sa aj tak podarilo natrénovať. Druhý pokus trénovania spočíval v zamrznutiu prvých vrstiev siete. Konkrétne sa U-Net skladá z dvoch hlavných častí a to downamapling a upsampling. Na konzultácií mi bolo poradené, aby som downsampling časť zamrzol. Nastavil som vrstvy tejto časti, aby sa neučili. To znamená, že sa zvýšil počet parametrov, ktoré nemenia svoju váhu. Priebeh trénovania zobrazuje obrázok 5.12.



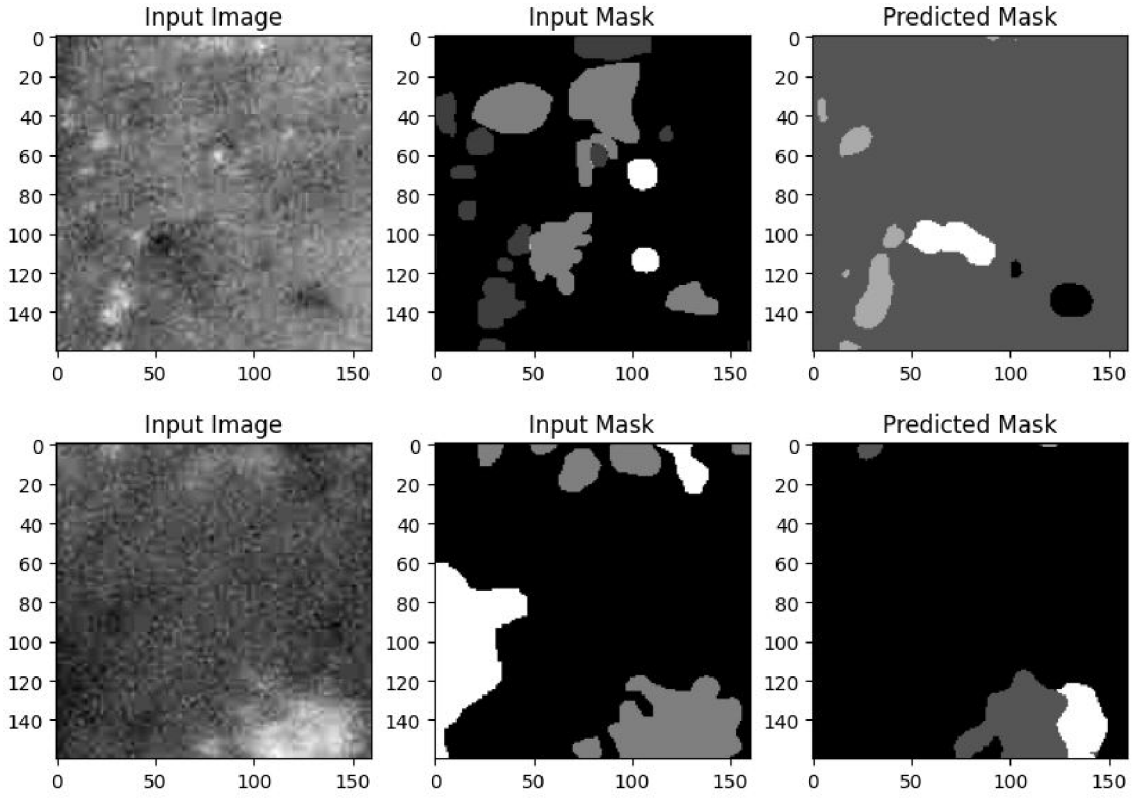
Obr. 5.12: Priebeh loss funkcie pri trénovaní

Trénoval som architektúru 50 epoch. Z grafu je vidieť, že loss funkcia na testovacej sade sa stále znižovala. Opačne to bolo u loss funkcie pri validačnej sade, kde hodnoty vyskočili vyššie a posledných 20 epoch sa držali pri rovnakej hodnote. Takýto priebeh podľa mňa vznikol kvôli náročnosti obrázkov. Obrázkov dokopy na trénovanie spolu s validačnou sadou bolo 23 868. Z toho počtu bolo veľa obrázkov, ktoré neobsahovali žiadnu patológiu, teda maska bola celá čierna. Tým sa dá povedať, že dátová sada bola menšie. V metrike OneHotIoU som v parametri nastavil, aby sa trieda ktorá nič neobsahovala nepočítala. Napriek tomu sa sieť dokázala naučiť a bola schopná predikovať v niektorých prípadoch veľmi presne. Obrázok 5.13 zobrazuje predikciu siete. Na obrázku je vidieť, že sieť dokázala celkom pekne sa priblížiť originálnej maske. Pri takejto úlohe je samozrejme, že sa sieť nenaučí na 100% predikovať všetko správne.



Obr. 5.13: Ukážka správnej predikcie

Ďalšou ukážkou predikcie je obrázok 5.14. Tento obrázok ukazuje opačnú stránku predikcie. Sieť so vstupným obrázkom sa nepriblížila svojou predikciou originálnej maske. Na obrázku si môžeme všimnúť, ak sa na obrázku nachádzalo viacej patologických nálezov predikcia nefungovala správne.



Obr. 5.14: Ukážka chybnej predikcie

5.3 SegFormer

Po natrénovaní U-Net binárnej podoby a multiclass podoby som sa pustil do architektúry SegFormer. Ako bolo spomínané vyššie SegFormer funguje na multiclass sémantickej segmentácie. V rámci riešenia som vychádzal z oficiálnej keras stránky[2]. Chcel som porovnať úspešnosť architektúry. Bohužiaľ cez veľa pokusov sa mi nepodarilo natrénovať túto sieť. Rovnakým spôsobom pristupoval k tomu, ako pri architektúre U-Net. Keďže táto architektúra funguje pre multiclass segmentáciu, nemohol som trénovať binárnu segmentáciu teda napr. rozpoznávania exudátov. Z toho dôvodu som rovnakým spôsobom začal ako o U-Net architektúry. Prv som chcel použiť dataset CIPH [19]. Nepodarilo sa mi však pustiť trénovanie na tejto spomínanej sietí. Vstupné obrázky sa nepodarilo načítať. Z toho dôvodu som vyskúšal rovno svoj IDRiD dataset [27]. Výsledok bol rovnaký. Samotné trénovanie sa nespustilo. Po viacerých neúspešných pokusov som sa pokúsil zmeniť knižnicu. Pracoval som doteraz s knižnicou *tensorflow*. Pre tento prípad som vyskúšal knižnicu *pytorch*. S touto knižnicou som predtým nepracoval a dostal som ku nej až v neskoršej fázy práce. Pokúšal som sa rovnako svoj dataset natrénovať a využiť túto sieť žiaľ sa to nepodarilo. Možnú chybu pri implementácií mohla byť, pri zlom spracovaní obrázkov. Kde bolo za potreby použiť identifikáciu ku každej hodnote masky. Môžno som s touto identifikáciou nesprávne zachádzal. Ďalšou možnou chybou mohla byť zlá implementácia, alebo zlé použitie architektúry. Z dôvodu nenatrénovania modelu SegFormer nemôžem na svojom datasete ukázať fungovanie tejto sieti.

[2] *https : //keras.io/examples/vision/segformer/*

Z teoretického hľadiska sa môžeme pozrieť ako by také trénovanie mohlo vyzerať. V kapitole modelu SegFormer 3.6, bolo vysvetlené ako model navrhli. V rámci vytvorenia modelu vnizkli 6 rôznych veľkostí kodéra. Tieto kodéry boli označené MiT-B0 až MiT-B5. Ich rozdiel predovšetkým spočíval vo veľkosti parametrov. Najmenší kodér MiT-B0 obsahoval 3.7 milióna parametrov a najväčší kodér MiT-B5 obsahoval 82 miliónov parametrov. V rámci úspešnosti a predovšetkým metriky IoU, väčší počet parameter značil lepšiu úspešnosť ako menší počet parametrov. Sieť bola trénovaná na datasetoch, ktoré obsahovali viac ako 100 kategórií. V rámci tejto práce nás maximálne zaujímalo 5 kategórií. Z toho dôvodu by som rovnako prv použil dataset CIPH, v ktorom som si vybral z 20 kategórií päť, aby to čo najviac reprezentovalo moju problematiku. Následne by som natrénoval sieť na viacero kodérov. Zo začiatku by som začal s najmenším MiT-B0. Následne by som prešiel na trénovanie najväčšieho kodéra MiT-B5. V prípade dostatku času, by som zvážil do-trénovanie na všetkých typov kodérov. Výsledkom by mohlo byť porovnanie podobné ako to mali autori vo svojom článku [39]. Po takomto natrénovaní by mi vznikli váhy. Potom na týchto váhach by som do-trénoval svoj dataset s patologickými nálezmi na sietnici oka. Výsledkom by bolo určitá úspešnosť správnej segmentácií. Postupne by sa vedelo porovnať úspešnosť medzi jednotlivými kodérmi alebo porovnať s architektúrou U-Net.

V rámci svojej predikcie a analýzy, by táto architektúra nemala veľmi vysokú úspešnosť. Je to z dôvodu toho, že takýto typ sietí na natrénovanie potrebuje robustnú dátovú sadu. S mojou dátovou sadou, ktorá bola ešte znásobená tým, že sa pôvodné veľké obrázky rozdelili na menšie obrázky napríklad na veľkosti 160×160 by nestačili na úspešné natrénovanie. Na úspešné trénovanie sémantické segmentácií treba sto tisícky obrázkov. Tento dataset mal pôvodne 54 obrázkov na natrénovanie, z ktorých som spravil 23 868 obrázkov o veľkosti 160×160. Sieť by sa určite nejak natrénovala, ako to bolo u architektúry U-Net. Lenže nemala by oslňujúce výsledky.

5.4 Zhrnutie

Táto kapitola sa venovala implementácií a môjmu spôsobu postupu práce. V rámci implementácie boli rovno vykonané aj experimenty na docielenie úspešnosti. Na začiatku som začal s architektúrou U-Net a konkrétne s binárnou segmentáciou. Snažil som sa detekovať drúzy. Zo začiatku som mal problémy s trenovaním, tie som vyriešil a vysvetlil. Po binárnej segmentácií som sa pokúsil o multiclass segmentáciu. Na obrázku som sa snažil rozpoznať 4 typy patologických nálezov menovite: tvrdé exudáty, mäkké exudáty, mikroaneurizmy a hemoragiu. Piata kategória bolo pozadie sietnice. Rovnako som prvé natrénoval na modelu U-Net. V počiatku sa sieť nedokázala učiť a nič nepredikovala. Z toho dôvodu som použil dataset CIPH [19], ktorý som natrénoval na piatich kategóriach, aby sa podobal mojej problematike. Následne som natrénované váhy použil na dotrénovanie na mojom datasete. Takto sa sieť dokázala naučiť a predikovala patologické nálezy. Po úspešnom natrénovaní modelu U-Net som sa snažil natrénovať SegFormer. Bohužiaľ túto architektúru sa mi nepodarilo natrénovať a nedokážem porovnať dve architektúry. V rámci riešenia som teoreticky popísal, ako by som to riešil a aké výsledky by mohli vzniknúť po natrénovaní.

Kapitola 6

Záver

Táto práca sa venovala sémantickou segmentáciou patologických nálezov na sietnici oka. V prvej kapitole bola vysvetlená stavba ľudského oka, sietnice a patologické nálezy. Základné patológie sú: mikroaneurizmy, drúzy, exudáty a hemoragia.

V ďalšej kapitole bola vysvetlená sémantická segmentácia, konkrétne spolupráci s neurónovými sieťami. Boli vysvetlené vybrané architektúry, ktoré danú problematiku riešia. Rovnako podrobnejšie bola popísaná novšia architektúra SegFormer, ktorá čerpá hlavné rysy z architektúr pre spracovanie prirodzeného jazyka. Konkrétne používa transformátory.

V rámci riešenia práce som našiel vhodný dataset, ktorý obsahoval obrázky sietnic oka spolu s anotovanými patologickými nálezmi pre sémantickú segmentáciu. Tento dataset som si upravil, aby bol vhodný na natrénovanie. Vstupné obrázky sietnic oka, boli vo veľkom rozlíšení preto som ich musel jednotlive orezať na menšiu veľkosť. Na začiatku som sa pokúšal trénovať iba jednu patológiu. Použitá bola architektúra U-Net. Úspešne sa mi podarilo natrénovať binárnu segmentáciu a v rámci rozšírenia som natrénoval všetky patologické nálezy, ktoré dataset obsahoval. Po binárnej segmentácií som vyskúšal natrénovať multiclass segmentáciu. Tá spočíva v tom, že všetky 4 patológie som spojil a trénoval som ich vzájomne rozpoznanie. Rovnako som využil U-Net architektúru. V rámci tejto úlohy som využil druhý dataset na predtrénovanie neurónovej siete. Za pomoci transfer learning som dotrénoval sieť na mojom datasete. Sieť sa podarilo natrénovať, dokázala predikovať nálezy, ale s určitou nepresnosťou.

V ďalšom bode som sa pokúsil natrénovať na rovnakom datasete aj architektúru SegFormer. Bohužiaľ aj po niekoľkých pokusoch sa mi nepodarilo túto sieť natrénovať. V práci bolo teoretický opísané ako by táto sieť mohla fungovať.

V rámci pokračovania tejto práce, môže byť dokončené trénovanie architektúry SegFormer. Následne by sa dalo porovnať architektúru U-Net a SegFormer pre patologické nálezy na sietnici oka. Výsledky by mohli pomôcť v budúcnosti pri určení čo najlepšej architektúry na daný problém.

Literatúra

[1] *How U-net works?* [online]. [cit. 2022-12-27]. Dostupné z: https://developers.arcgis.com/python/guide/how-unet-works/.

[2] *What is the difference between drusen and exudates?* [online]. [cit. 2023-02-23]. Dostupné z: https://www.aao.org/eye-health/ask-ophthalmologist-q/what-is-difference-between-drusen-exudates.

[3] Badrinarayanan, V., Kendall, A., Cipolla, R. a Senior Member, I. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation* [online]. 2016 [cit. 2022-12-27]. Dostupné z: https://arxiv.org/pdf/1511.00561.pdf.

[4] Caesar, H., Uijlings, J. a Ferrari, V. *COCO-Stuff: Thing and Stuff Classes in Context* [online]. 2018 [cit. 2023-03-19]. Dostupné z: https://arxiv.org/abs/1612.03716.

[5] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M. et al. *The Cityscapes Dataset for Semantic Urban Scene Understanding* [online]. 2016 [cit. 2023-03-19]. Dostupné z: https://arxiv.org/abs/1604.01685.

[6] Costea, A. D., Petrovai, A. a Nedevschi, S. *Fusion Scheme for Semantic and Instance-level Segmentation* [online]. 2018 [cit. 2022-12-26]. Dostupné z: https://www.researchgate.net/profile/Andra-Petrovai/publication/329616112/figure/fig2/AS:739657281175556@1553359446925/Panoptic-segmentation-by-unifying-semantic-and-instance-segmentation.ppm.

[7] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. et al. *Imagenet: A large-scale hierarchical image databases* [online]. 2009 [cit. 2023-02-23]. Dostupné z: https://www.image-net.org/.

[8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X. et al. *AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE* [online]. 2021 [cit. 2023-03-16]. Dostupné z: https://arxiv.org/pdf/2010.11929.pdf.

[9] Hariharan, B., Arbeláez, P., Girshick, R. a Malik, J. *Simultaneous Detection and Segmentation* [online]. 2014 [cit. 2023-01-15]. Dostupné z: https://arxiv.org/pdf/1407.1808.pdf.

[10] He, K., Gkioxari, G., Dollár, P. a Girshick, R. *Mask R-CNN* [online]. 2018 [cit. 2023-01-16]. Dostupné z: https://arxiv.org/pdf/1703.06870.pdf.

[11] Hložánek, M. *Přístrojová technika v oftalmologii.* 1. vyd. Univerzita Karlova,2. lékařská fakulta, 2006. ISBN 80-902160-9-9.

[12] Iakubovskii, P. *Segmentation Models* [online]. GitHub, 2019 [cit. 2023-02-23]. Dostupné z: https://github.com/qubvel/segmentation_models.

[13] Kolář, P. *Věkem podmíněná makulární degenerace.* 1. vyd. GRADA, 2008. ISBN 978-80-247-2605-2.

[14] Kuchynka, P. *Oční lékařství.* 2. vyd. Grada, 2016. ISBN 978-802-4750-798.

[15] Kvapilíková, K. *Vyšetřování oka.* 1. vyd. Institut pro další vzdělávání pracovníků ve zdravotnictví, 1995. ISBN 80-7013-195-0.

[16] Liu, H., Peng, C., Yu, C., Wang, J., Liu, X. et al. *An End-to-End Network for Panoptic Segmentation* [online]. 2019 [cit. 2022-12-27]. Dostupné z: https://arxiv.org/pdf/1903.05027.pdf.

[17] Liu, S., Qi, L., Qin, H., Shi, J. a Jia, J. *Path Aggregation Network for Instance Segmentation* [online]. 2018 [cit. 2023-01-17]. Dostupné z: https://arxiv.org/pdf/1803.01534.pdf.

[18] Liu, X., Deng, Z. a Yang, Y. *Recent progress in semantic image segmentation* [online]. 2018 [cit. 2023-03-20]. Dostupné z: https://arxiv.org/ftp/arxiv/papers/1809/1809.10198.pdf.

[19] Loesch, A. a Audigier, R. *Describe Me If You Can! Characterized Instance-Level Human Parsing* [online]. 2021 [cit. 2023-04-19]. Dostupné z: https://ieeexplore.ieee.org/document/9506509/authors#authors.

[20] Mašek, P., Cholevík, D. a Němčanský, J. *Oftalmologie a diagnostické metody a přístroje v oftalmologii : studijní opora.* 1. vyd. Ostrava : Ostravská univerzita v Ostravě, 2014. ISBN 978-80-7464-569-3.

[21] Mishra, A. *Contrast Limited Adaptive Histogram Equalization (CLAHE) Approach for Enhancement of the Microstructures of Friction Stir Welded Joints* [online]. 2021 [cit. 2023-01-20]. Dostupné z: https://arxiv.org/ftp/arxiv/papers/2109/2109.00886.pdf.

[22] Munoz, E. *Attention is all you need: Discovering the Transformer paper* [online]. Towardsdatascience, 2020 [cit. 2023-01-15]. Dostupné z: https://towardsdatascience.com/attention-is-all-you-need-discovering-the-transformer-paper-73e5ff5e0634.

[23] Noh, H., Hong, S. a Han, B. *Learning Deconvolution Network for Semantic Segmentation* [online]. 2015 [cit. 2022-12-27]. Dostupné z: https://arxiv.org/pdf/1505.04366.pdf.

[24] Parkhi, O. M., Vedaldi, A., Zisserman, A. a Jawahar, C. V. *The Oxford-IIIT Pet Dataset* [online]. 2012 [cit. 2023-03-19]. Dostupné z: https://www.robots.ox.ac.uk/~vgg/data/pets/.

[25] Pinheiro, P. O., Collobert, R. a Dollár, P. *Learning to Segment Object Candidates* [online]. 2015 [cit. 2023-01-15]. Dostupné z: https://arxiv.org/pdf/1506.06204.pdf.

[26] Pinheiro, P. O., Lin, T.-Y., Collobert, R. a Dollár, P. *Learning to Refine Object Segments* [online]. 2016 [cit. 2023-01-16]. Dostupné z: https://arxiv.org/pdf/1603.08695.pdf.

[27] Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G. et al. *INDIAN DIABETIC RETINOPATHY IMAGE DATASET (IDRID)* [online]. 2019 [cit. 2023-03-12]. Dostupné z: https://ieee-dataport.org/open-access/indian-diabetic-retinopathy-image-dataset-idrid.

[28] Ronneberger, O., Fischer, P. a Box, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation* [online]. 2015 [cit. 2022-12-27]. Dostupné z: https://arxiv.org/pdf/1505.04597.pdf.

[29] Skouta, A., Elmoufidi, A., Jai Andaloussi, S. a Ouchetto, O. *Hemorrhage semantic segmentation in fundus images for the diagnosis of diabetic retinopathy by using a convolutional neural network* [online]. 2022 [cit. 2023-03-22]. Dostupné z: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00632-0.

[30] Sultana, F., Sufian, A. a Dutta, P. *Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey* [online]. 2020 [cit. 2022-12-27]. Dostupné z: https://arxiv.org/pdf/2001.04074.pdf.

[31] Synek, S. a Skorkovská Šárka. *Fyziologie oka a vidění*. 2. vyd. GRADA, 2014. ISBN 978-80-247-3992-2.

[32] Tran, M. *Understanding U-Net* [online]. Towardsdatascience, 2022 [cit. 2023-01-10]. Dostupné z: https://towardsdatascience.com/understanding-u-net-61276b10f360.

[33] Tsang, S.-H. *Review: DeconvNet — Unpooling Layer (Semantic Segmentation)* [online]. [cit. 2022-12-27]. Dostupné z: https://towardsdatascience.com/review-deconvnet-unpooling-layer-semantic-segmentation-55cf8a6e380e.

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. *Attention Is All You Need* [online]. 2017 [cit. 2022-12-30]. Dostupné z: https://arxiv.org/pdf/1706.03762.pdf.

[35] Walia, M. *Semantic Segmentation vs. Instance Segmentation: Explained* [online]. Roboflow, 2022 [cit. 2023-01-05]. Dostupné z: https://blog.roboflow.com/difference-semantic-segmentation-instance-segmentation/.

[36] Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A. et al. *Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation* [online]. 2020 [cit. 2022-12-30]. Dostupné z: https://arxiv.org/pdf/2003.07853.pdf.

[37] Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K. et al. *Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions* [online]. 2021 [cit. 2023-03-15]. Dostupné z: https://arxiv.org/pdf/2102.12122.pdf.

[38] Wikipedia. *Oko* [online]. 2022 [cit. 2022-12-26]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/thumb/a/a4/Schematic_diagram_of_the_human_eye_sk.svg/800px-Schematic_diagram_of_the_human_eye_sk.svg.png/.

[39] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M. et al. *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers* [online]. 2021 [cit. 2023-03-15]. Dostupné z: https://arxiv.org/pdf/2105.15203.pdf.

[40] Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M. et al. *UPSNet: A Unified Panoptic Segmentation Network* [online]. 2019 [cit. 2022-12-27]. Dostupné z: https://arxiv.org/pdf/1901.03784.pdf.

[41] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z. et al. *Rething Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers* [online]. 2021 [cit. 2022-12-29]. Dostupné z: https://arxiv.org/pdf/2012.15840.pdf.

[42] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A. et al. *Scene Parsing through ADE20K Dataset* [online]. 2017 [cit. 2023-03-19]. Dostupné z: https://ieeexplore.ieee.org/document/8100027.

[43] Čihák, R. *Anatomie 3.* 2. vyd. GRADA, 2004. ISBN 978-80-247-1132-4.