

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020

Bc. Peter Horečný



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

METODY SEGMENTACE OBRAZU S MALÝMI TRÉNOVACÍMI MNOŽINAMI

IMAGE SEGMENTATION METHODS WITH LIMITED DATA SETS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Peter Horečný

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Radim Burget, Ph.D.

BRNO 2020

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Peter Horečný

ID: 186083

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Metody segmentace obrazu s malými trénovacími množinami

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodami segmentace obrazu používaných v posledních letech, včetně architektury neuronové sítě UNet. Zpracujte poslední vývoj v této oblasti. Aktuální metody vzájemně srovnajte a popište jejich výhody a nevýhody. Dále se zaměřte na metody, které umožňují přesné strojové učení i v případě datových množin malé velikosti (například ODE sítě, augmentace datových množin atp.). Navrhněte metodu pro segmentaci obrazu a dosažené výsledky vhodně srovnajte. Dosažené výsledky zanechte do tabulek a hodnoty diskutujte včetně vzájemného srovnání se stavem ve světě.

DOPORUČENÁ LITERATURA:

- [1] Chen, Tian Qi, et al. "Neural ordinary differential equations." Advances in neural information processing systems. 2018.
- [2] Garcia-Garcia, Alberto, et al. "A survey on deep learning techniques for image and video semantic segmentation." Applied Soft Computing 70 (2018): 41-65.

Termín zadání: 3.2.2020

Termín odevzdání: 1.6.2020

Vedoucí práce: doc. Ing. Radim Burget, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cieľom tejto práce bolo navrhnúť metódu segmentácie obrazu, ktorá dokáže efektívne pracovať aj s dátovými množinami malej veľkosti. Bola k tomu využitá nedávno publikovaná neurónová sieť ODE, ktorá by vďaka svojim vlastnostiam mala dosahovať lepšej generalizácie v prípade riešenia úlohy s malým množstvom trénovacích dát. Navrhnutá sieť ODE-UNet vznikla kombináciou architektúry UNet a siete ODE, pričom využíva výhody oboch sietí. Na ISBI dátovej sade bolo dosiahnutých presností Rand: 0,950272 a Info: 0,978061 na testovacích snímkoch, čím bola prekonaná presnosť siete UNet, ktorá bola takisto v tejto práci testovaná. Týmito výsledkami bolo ale preukázané, že pomocou ODE sietí nie je možné prekonať stav vedy a techniky. Avšak preukázaných bolo aj niekoľko výhod oproti testovanej UNet architektúre a aktuálne používaným metódam a v rámci ďalšieho rozširovania by bolo možné výsledky stále zlepšovať.

KLÚČOVÉ SLOVÁ

ISBI dátová sada, malé trénovacie množiny, metódy segmentácie, ODE neurónová sieť, počítačové videnie, segmentácia obrazu, spracovanie obrazu, strojové učenie, umelá inteligencia, UNet

ABSTRACT

The goal of this thesis was to propose an image segmentation method, which is capable of effective segmentation process with small datasets. Recently published ODE neural network was used for this method, because its features should provide better generalization in case of tasks with only small datasets available. The proposed ODE-UNet network was created by combining UNet architecture with ODE neural network, while using benefits of both networks. ODE-UNet reached following results on ISBI dataset: Rand: 0,950272 and Info: 0,978061. These results are better than the ones received from UNet model, which was also tested in this thesis, but it has been proven that state of the art can not be outperformed using ODE neural networks. However, the advantages of ODE neural network over tested UNet architecture and other methods were confirmed, and there is still a room for improvement by extending this method.

KEYWORDS

ISBI dataset, small training datasets, segmentation methods, ODE neural network, computer vision, image segmentation, image processing, machine learning, artificial intelligence, UNet

HOREČNÝ, Peter. *Metody segmentace obrazu s malými trénovacími množinami*. Brno, 2020, 70 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Radim Burget, Ph.D.

VYHLÁSENIE

Vyhlasujem, že svoju diplomovú prácu na tému „Metody segmentace obrazu s malými trénovacími množinami“ som vypracoval samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi doc. Ing. Radimovi Burgetovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Tato práce vznikla jako součást klíčové aktivity KA6 - Individuální výuka a zapojení studentů bakalářských a magisterských studijních programů do výzkumu v rámci projektu OP VVV Vytvoření double-degree doktorského studijního programu Elektronika a informační technologie a vytvoření doktorského studijního programu Informační bezpečnost, reg. č. CZ.02.2.69/0.0/0.0/16_018/0002575.



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY

Projekt je spolufinancován Evropskou unií.

Obsah

Úvod	11
1 Neurónové siete	12
1.1 Umelá neurónová sieť	12
1.2 Konvolučná neurónová sieť	15
1.3 ODE neurónová sieť	19
1.4 Malé tréningové množiny	24
2 Segmentácia obrazu	26
2.1 Metódy segmentácie	26
2.2 Architektúry hlbokého učenia	28
3 Popis experimentov	35
3.1 ISBI dátová sada	35
3.2 Porovnanie známych architektúr	37
3.3 Návrh novej architektúry	39
4 Výsledky a diskusia	43
4.1 Jednotlivé experimenty	43
4.2 Porovnanie architektúr	55
Záver	61
Literatúra	62
Zoznam symbolov, veličín a skratiek	69

Zoznam obrázkov

1.1	Umelý neurón	12
1.2	Jednoduchá plne-prepojená neurónová sieť	13
1.3	Číselná reprezentácia filtra a vizualizácia krivky	16
1.4	Architektúra UNet	18
1.5	Zmeny stavu neurónovej siete / ODE siete	20
1.6	Závislosť stavu - klasická neurónová sieť / ResNet	21
1.7	Vyhodnocovacie body pre rôzne zložité vývoje stavu siete	22
1.8	Podtrénovanie a pretrénovanie modelu	25
2.1	Použitie siete FCN pre sémantickú segmentáciu. Zdroj: [24]	29
2.2	Prepojenia siete FCN medzi hrubými informáciami z vyšších vrstiev a detailnými informáciami z hlbokých vrstiev. Zdroj: [24]	29
2.3	Architektúra siete SegNet. Zdroj: [27]	30
2.4	Architektúra siete PSPNet. Zdroj: [31]	31
2.5	Architektúra siete Mask R-CNN. Zdroj: [38]	32
2.6	Rozšírená konvolúcia - recepčné polia pri rôznych hodnotách miery učenia (mr)	33
3.1	Originálny obrázok a hraničná mapa (ground truth boundary map)	36
3.2	Architektúra siete ODE	39
3.3	Autoenkóder využívajúci kombináciu ODE a UNet siete	40
3.4	Výsledky prahovania výsledného snímku	41
3.5	Návrh architektúry UNet-1-ODE	42
4.1	Podobnosť troch tréningových snímok	43
4.2	Graf tréningovej a validačnej presnosti DICE siete UNet	44
4.3	Graf tréningovej a validačnej presnosti Accuracy siete UNet	44
4.4	Graf tréningovej a validačnej presnosti DICE siete ODE	45
4.5	Graf tréningovej a validačnej presnosti Accuracy siete ODE	45
4.6	Priebeh tréningovej presnosti modelu odeunet1 (s jednoduchými 2d konv.) a modelu odeunet2 (so zretazenými 2d konv.)	47
4.7	Priebeh validačnej presnosti modelu odeunet1 (s jednoduchými 2d konv.) a modelu odeunet2 (so zretazenými 2d konv.)	47
4.8	Priebeh tréningovej presnosti modelu ODE-UNet s rôznou hĺbkou (h_3 = hĺbka 3)	48
4.9	Priebeh validačnej presnosti modelu ODE-UNet s rôznou hĺbkou (h_3 = hĺbka 3)	49
4.10	Priebeh tréningovej presnosti modelu ODE-UNet s rôznou mierou uče- nia (lr_3 = miera učenia $1e^{-3}$)	50

4.11	Priebeh validačnej presnosti modelu ODE-UNet s rôznou mierou učenia ($lr_3 =$ miera učenia $1e^{-3}$)	50
4.12	Priebeh trérovacej presnosti modelu ODE-UNet s rôznou toleranciou ($tol_3 =$ tolerancia $1e^{-3}$)	52
4.13	Priebeh validačnej presnosti modelu ODE-UNet s rôznou toleranciou ($tol_3 =$ tolerancia $1e^{-3}$)	52
4.14	Priebeh trérovacej presnosti modelu UNet-1-ODE1 (s jednoduchými 2d konv.) a modelu UNet-1-ODE2 (so zretazenými 2d konv.)	53
4.15	Priebeh validačnej presnosti modelu UNet-1-ODE1 (s jednoduchými 2d konv.) a modelu UNet-1-ODE2 (so zretazenými 2d konv.)	54
4.16	Priebeh trérovacej presnosti modelu UNet-1-ODE s rôznymi hodnotami tolerancie ($tol_3 =$ tolerancia $1e^{-3}$)	55
4.17	Priebeh validačnej presnosti modelu UNet-1-ODE s rôznymi hodnotami tolerancie ($tol_3 =$ tolerancia $1e^{-3}$)	55
4.18	Porovnanie priebehu trérovacej presnosti DICE sietí UNet, ODE, ODE-UNet a UNet-1-ODE	56
4.19	Porovnanie priebehu trérovacej presnosti ACCURACY sietí UNet, ODE, ODE-UNet a UNet-1-ODE	57
4.20	Porovnanie priebehu validačnej presnosti DICE sietí UNet, ODE, ODE-UNet a UNet-1-ODE	57
4.21	Porovnanie priebehu validačnej presnosti ACCURACY sietí UNet, ODE, ODE-UNet a UNet-1-ODE	58
4.22	Príklad validačných snímok siete ODE-UNet	58
4.23	Generované testovacie snímky jednotlivých architektúr	59

Zoznam tabuliek

4.1	Tabulka prahových hodnôt	48
4.2	Tabulka výsledkov pre rôzne hodnoty hĺbky	49
4.3	Tabulka výsledkov pre rôzne hodnoty miery učenia	51
4.4	Tabulka výsledkov pre rôzne hodnoty tolerancie	51
4.5	Tabulka výsledkov pre rôzne hodnoty tolerancie	54
4.6	Tabulka výsledkov	60

Úvod

Umelá inteligencia je významnou súčasťou života každého z nás, či už si to uvedomujeme, alebo nie. Firmy pomocou aplikácií založených na strojovom učení ľuďom ponúkajú v internetových obchodoch najvhodnejšie produkty, ponúkajú predpovede cien leteniek alebo akcií, predpovedajú dopravné zápchy alebo zobrazujú v reálnom čase, predvídajú šírenie chorôb, rozoznávajú hlasové príkazy a veľa ďalšieho. Takmer všetky tieto aplikácie majú spoločné to, že na základe veľkej, historickej dátovej množiny sa učia predpovedať budúci vývoj alebo vyhodnocovať aktuálny stav. Napríklad pri analýze genetických údajov alebo medicínskych snímok z magnetickej rezonancie sa učia na predošlých analýzach a následne vyhodnocujú nové snímky.

Pri vývoji ale často nastáva problém, že vyžadujú veľké množstvo trénovacích dát, aby bolo možné dosiahnuť presných hodnôt. Pomocou nich sa totiž vytvára alebo trénuje model, ktorý sa následne používa s neznámymi dátami. Čím menšia je trénovacia množina, tým obtiažnejšie je získať zovšeobecnenú informáciu a natrénovať vhodný model, ktorý by dosahoval požadovaných presností. Voľba vhodného algoritmu je v takýchto prípadoch veľmi podstatná, keďže nie každý dokáže pracovať s obmedzenou dátovou množinou rovnako. Vznikajú aj metódy, ktoré pomáhajú s maximálne efektívnym využitím dostupných dát.

Táto práca sa zaoberá porovnávaním dostupných algoritmov a metód pre segmentáciu obrazu s malými trénovacími množinami. Využíva k tomu neurónové siete UNet [1] v kombinácii s Ordinary Differential Equations (ODE) sieťami [2] a množinu 30 obrázkov, ktoré boli zverejnené k súťaži o najlepší model pre segmentáciu mikroskopických snímok mozgu octomilky.

Hlavným prínosom práce je návrh novej architektúry využívajúcej ODE sieť, u ktorej bola predpokladaná lepšia presnosť ako u aktuálne používaných riešení. Tento návrh vychádza z predpokladu, že nedávno odprezentovaný typ neurónových sietí ODE, ktoré sa ukázali ako presnejšie pri viacerých typoch úloh, by mohli zlepšiť výsledky aj pri segmentácii obrazu, čo doposiaľ nebolo overené.

Zbytok práce je štrukturovaný následovne. V prvej kapitole je popísaný všeobecný princíp neurónových sietí a funkcionality jednotlivých typov, ktorými sa práca následne zaoberá. Druhá kapitola sa venuje segmentácii obrazu, metódam, ktorými možno riešiť segmentačné úlohy a aktuálnym architektúram hlbokého učenia. V tretej kapitole sa popisuje celý experiment, použité dáta, spôsob trénovania a hlavne návrh novej architektúry. V poslednej kapitole sú zhrnuté výsledky vo forme grafov, tabuliek a vyhodnotení všetkých použitých architektúr a porovnanie ich presnosti a vhodnosti použitia.

1 Neurónové siete

Strojové učenie zahŕňa techniky, pomocou ktorých dokážeme predpovedať vývoj dát, klasifikovať, zoskupovať alebo segmentovať dáta, hľadať neznáme súvislosti apod. Jednou z týchto techník je hlboké učenie, ktoré využíva neurónové siete a množstvo ich variácií alebo architektúr.

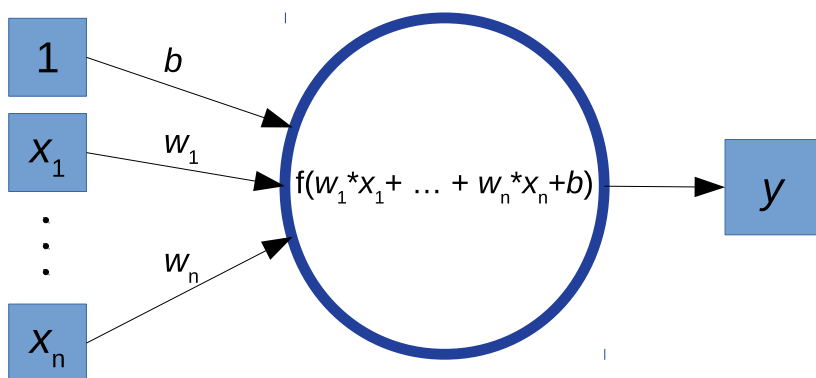
Neurónové siete sa ukázali ako veľmi efektívny nástroj napríklad pri rozpoznávaní hlasu, objektov v obraze alebo videu, a pri spracovávaní textu. Vychádzajú z funkcionality neurónových sietí v ľudskom mozgu.

1.1 Umelá neurónová sieť

Základnou výpočtovou jednotkou každej neurónovej siete je neurón (obr.1.1). Ten prijíma niekoľko vstupných hodnôt od iných neurónov alebo z vonkajšieho zdroja a vypočíta z nich jednu výstupnú hodnotu. Každá vstupná hodnota má priradenú váhu, čo je relatívna hodnota významnosti voči ostatným vstupom. Výstupná hodnota neurónu sa počíta ako funkcia súčtu súčinov vstupov a ich váh [3].

$$y = f(w_1 * x_1 + \dots + w_n * x_n + b)$$

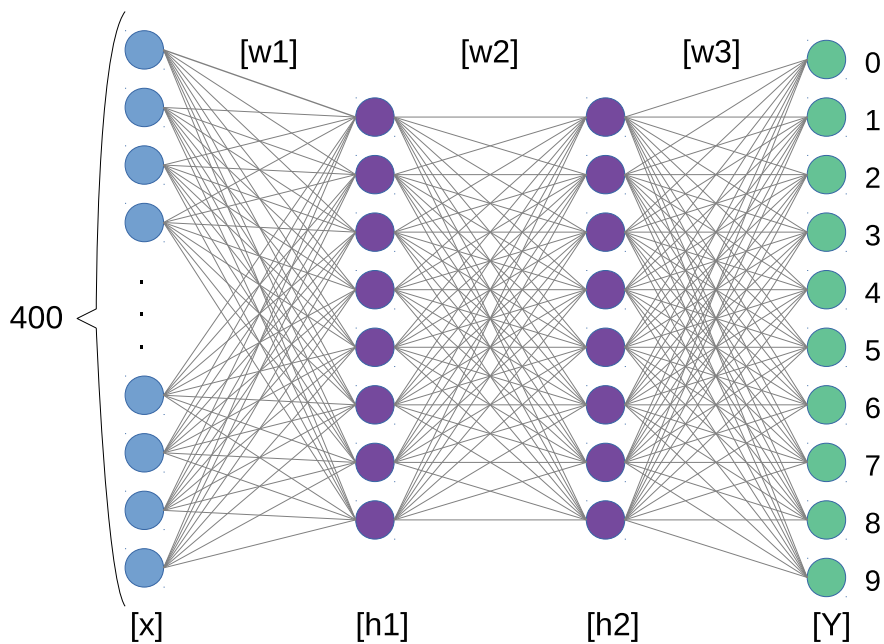
Funkcia f sa nazýva aktivačná funkcia a používa sa pre zavedenie nelinearity do výstupu neurónu. To je potrebné, lebo dáta v reálnom svete sú nelineárnej povahy. Najčastejšie sa stretávame s aktivačnými funkciami Sigmoid, tanh a Rectified Linear Unit (ReLU) [9]. Premenná b vo vzorci označuje vychýlenie (angl. bias), ktoré ovplyvňuje výstup aktivačnej funkcie neurónu.



Obr. 1.1: Umelý neurón

Neurónová sieť je viac-vrstvová sieť neurónov. Na obr.1.2 je znázornená jednoduchá sieť, ktorá je schopná rozoznávať ručne písané číslice od 0 po 9. Na vstupnej

vrstve sa nachádza 400 neurónov, ktoré reprezentujú každý jeden pixel čierneho-bieleho obrázku o rozmeroch 20x20 pixelov, na ktorom je číslica napísaná. Každý neurón udržiava určitú číselnú hodnotu (v intervale od 0 do 1), ktorú nazývame aktivácia. Aktivácia v tomto prípade značí odtieň šedej daného pixelu. Výstupná vrstva (vektor hodnôt $[Y]$) pozostáva z desiatich neurónov, kde každý reprezentuje jednu číslicu. Aktivácia týchto neurónov určuje, do akej miery sieť predpovedá, že na obrázku je napísaná daná číslica. Podľa neurónu s najväčšou aktiváciou klasifikuje číslicu na obrázku. Fialové krúžky na obr.1.2 reprezentujú dve vnútorné vrstvy. Tie majú za úlohu postupne detekovať rôzne úsečky, hrany, oblúky a spájať ich až k výslednému tvaru číslice. Každá vrstva ovplyvňuje aktivácie neurónov v nasledujúcej vrstve, tak ako to je znázornené na obr. 1.1. Nastavovaním váh spojov medzi jednotlivými neurónmi sieť určuje, do akej miery sa neuróny ovplyvňujú. V plne prepojenej architektúre je každý neurón v jednej vrstve spojený s každým neurónom vo vrstve nasledujúcej. V tejto sieti sa teda jedná o 3344 spojov, čo znamená 3370 váh a vychýlení. Celú sieť si možno predstaviť ako naozaj komplikovanú funkciu, ktorá má 400 vstupných hodnôt, 3370 parametrov a 10 výstupných hodnôt. Správnym nastavením parametrov možno sieť natréňovať k požadovanému chovaniu s výbornými výsledkami [4].



Obr. 1.2: Jednoduchá plne-prepojená neurónová sieť

Trénovanie

Cielom tréovania je nachádzanie čo najvhodnejších váh a vychýlení. Spomínané číslo 3370 býva vo väčšine prípadoch oveľa väčšie a preto je tréovanie časovo a pamäťovo veľmi náročný proces. K tréovacím dátam je priradené tzv. označenie (label), čo je napríklad už známa klasifikácia obrázku. Všeobecne sú to vždy už známe alebo historické dáta, pomocou ktorých sieť natrénujeme, aby potom dokázala napr. predikovať alebo klasifikovať nové, neznáme dáta.

Tréovanie siete začína náhodným priradením váh všetkým spojmom medzi neurónmi. Na vstupnú vrstvu je privedených 400 pixelov obrázku, a po postupnom prechode vnútornými vrstvami siete, je na výstupnej vrstve niekoľko nejednoznačných hodnôt, z ktorých nie je možné obrázok klasifikovať. Sieť ale pozná z označenia obrázku, aká má byť jeho klasifikácia, a teda na výstupnej vrstve by mal mať neurón, ktorý zodpovedá danej kategórii hodnotu aktivácie 1, a ostatné neuróny hodnotu aktivácie 0. Následne sa spočíta cena (angl. cost) tréovacieho kroku, čo je rozdiel ideálneho vektora výstupných hodnôt a tých aktuálnych. Toto prebehne na všetkých obrázkoch z tréovacej množiny a vypočíta sa priemer z hodnôt ceny. Čím menší je tento priemer, nazývaný aj strata (angl. loss), tým lepšie sú nastavené váhy.

Hodnota straty je výsledkom loss funkcie, čo je funkcia, ktorá má na vstupe všetkých 3370 parametrov siete a ako parametre jej slúžia všetky obrázky z tréovacej množiny. Ďalším cieľom siete je nájsť minimum tejto funkcie. K tomu sa používa klesanie podľa gradientu (gradient descent [5]), pomocou ktorého sa pozmenia váhy tak, aby táto zmena spôsobila najväčší pokles v loss funkcii smerom k minimu. Algoritmus, ktorý počíta gradient funkcie sa nazýva spätná propagácia (angl. backpropagation [6]), čo je základom celého tréovania siete. Celý tento proces tvorí jednu epochu. Týchto epoch môže sieť potrebovať aj niekoľko stoviek, kým sa dostane k požadovanej presnosti, ktorú zväčša zisťujeme pomocou validácie.

Validácia a Testovanie

Všetky dostupné dáta sa väčšinou rozdeľujú na tréováciu, validačnú a testovaciu množinu. Všetko sú to historické, známe dáta, u ktorých poznáme ich označenie, ale každá množina sa používa na niečo iné. Počas tréovania na tréovacej množine, sa zvyčajne po každej epoche počíta presnosť na množine validačnej. Podľa tejto presnosti sa kontroluje, či nedochádza k pretrénovaniu siete, čo znamená prispôbenie sa siete len na tréovacie dáta, čím sa stáva sieť nepresná na dáta neznáme. K pretrénovaniu dochádza ak je tréovacia presnosť oveľa väčšia ako validačná. Takisto možno nastaviť, aby sa sieť tréovala až kým nedosiahne určitú hranicu validačnej presnosti.

Testovanie siete musí prebiehať na úplne neznámych dátach. Preto sa testovacia množina nepoužíva počas tréningu, ale až po jeho ukončení. Potom sa na použije vytvorený model a na testovacej množine sa vypočíta presnosť siete.

1.2 Konvolučná neurónová sieť

V roku 2012 spôsobil tento typ sietí významný posun najmä v oblasti spracovania obrazu [7]. Jedná sa hlavne o klasifikáciu objektov a segmentáciu. Kapitola 1.1 popisuje typický príklad klasifikácie, kde je sieť trébovaná na rozoznávanie objektov na obrázku, konkrétne číslíc. Segmentácia funguje na podobnom princípe, avšak nezaraďuje celý obrázok do určitej kategórie, ale každý jeden pixel a tým ho rozdeľuje na jednotlivé objekty a výstupom siete je celý obrázok s ohraničenými objektami.

Konvolučná neurónová sieť sa skladá z niekoľkých vrstiev, ktoré sa môžu opakovať. Po vstupnej vrstve sa zvyčajne striedajú konvulčné, podvzorkovacie (subsampling) a aktivačnej vrstvy. Ako výstupná vrstva sa používa plne prepojená vrstva, ktorá z výstupu poslednej vnútornej vrstvy vytvorí N -rozmerný vektor, kde N je počet tried v prípade klasifikácie, alebo to môže byť $N \times N$ rozmerný obrázok v prípade segmentácie.

Konvolučná vrstva

Je to vždy prvá vnútorná vrstva a využívajú sa v nej filtre. Vstupom môže byť napr. pole o rozmeroch $20 \times 20 \times 3$ (obrázok o rozmeroch 20×20 pixelov s tromi farebnými zložkami). Filter je pole, ktoré sa definuje pri vytváraní siete a môže mať rozmery $3 \times 3 \times 3$, $5 \times 5 \times 3$, $7 \times 7 \times 3$ a podobne. Tento filter sa použije na vstupné pole pixelov, a postupne sa posúva od začiatku až po koniec obrázku. Rovnica 2D konvolúcie je:

$$y[m, n] = h[m, n] \otimes f[m, n] = \sum_j \sum_k h[j, k] * f[m - j, n - k],$$

kde h, f sú vstupný obrázok, filter a m, n sú stĺpce, riadky matíc. V každej pozícii filtra sa vynásobí každý pixel z oblasti originálneho obrázku, ktorú momentálne označuje filter (recepčné pole) s hodnotou filtra na danej pozícii. Súčet týchto násobkov je výstupom konvolúcie v danej pozícii filtra. Následne sa filter posunie a proces sa opakuje až do konca obrázku. Výsledkom celej konvolúcie je pole čísel o rozmeroch $18 \times 18 \times 1$, ktoré sa nazýva príznaková mapa. V praxi sa používa viac ako jeden filter a od toho sa potom odvíja tretí rozmer príznakovej mapy [8].

Týmto postupom dokážu konvulčné vrstvy rozoznávať rôzne tvary. Každý filter možno považovať za identifikátor určitého tvaru. Tieto tvary sú zo začiatku jednoduché, ako napr. priame čiary, rohy, oblúky a postupne sa v ďalších vrstvách spájajú do komplexnejších tvarov od geometrických útvarov až po konečnú triedu obrázku,

napr. číslice alebo konkrétne druhy zvierat. Funkcia filtrov spočíva v usporiadaní väčších číselných hodnôt do polí, ktoré predstavujú požadovaný tvar. Na obr.1.3 je číselná reprezentácia filtra a vizualizácia krivky, ktorú filter detekuje. Pri konvolúcii sa hodnoty recepčného pola násobia s hodnotami filtrov a ak sa v aktuálnom recepčnom poli nachádza tvar, ktorý tento filter dokáže detekovať, výsledná hodnota konvolúcie v tejto oblasti je vysoká. Ak sa naopak v danom poli podobný tvar nenachádza, výsledná hodnota je nízka. Popísaný filter je zjednodušený a čím viac ich konvolučná vrstva obsahuje, tým viac získava informácii o obrázku [3].



Obr. 1.3: Číselná reprezentácia filtra a vizualizácia krivky

Parametre konvolučnej vrstvy

Dva hlavné parametre, ktorými možno ovplyvňovať konvolučnú vrstvu sú krok (stride) a výplň (padding). Krok určuje, o koľko miest sa filter pri konvolúcii posúva. Čím väčší je krok, tým viac sa vstupnú pole zmenší. Potrebné je ale dávať pozor, aby celé recepčné pole vždy pasovalo do vstupného pola.

Použitím výplne sa pridajú na vonkajšok obrazu navyše pixely, napríklad nulové. Veľkosť výplne určuje počet týchto pixelov. V konvolučných sieťach sa používa výplň kvôli tomu, že pri každom prechode konvolučnou vrstvou sa veľkosť obrazu zmenší, ale na výstupe siete je požadovaný obraz v pôvodnej veľkosti. [10]

Aktivačná vrstva

Po každej konvolučnej vrstve sa aplikuje aktivačná vrstva, ktorá zavádza do systému nelinearitu. Pre rýchlosť tréningu je v súčasnosti najviac preferovaná ReLU vrstva. Matematicky je ReLU funkcia definovaná nasledovne:

$$y = \max(0, x)$$

Podvzorkovacia vrstva

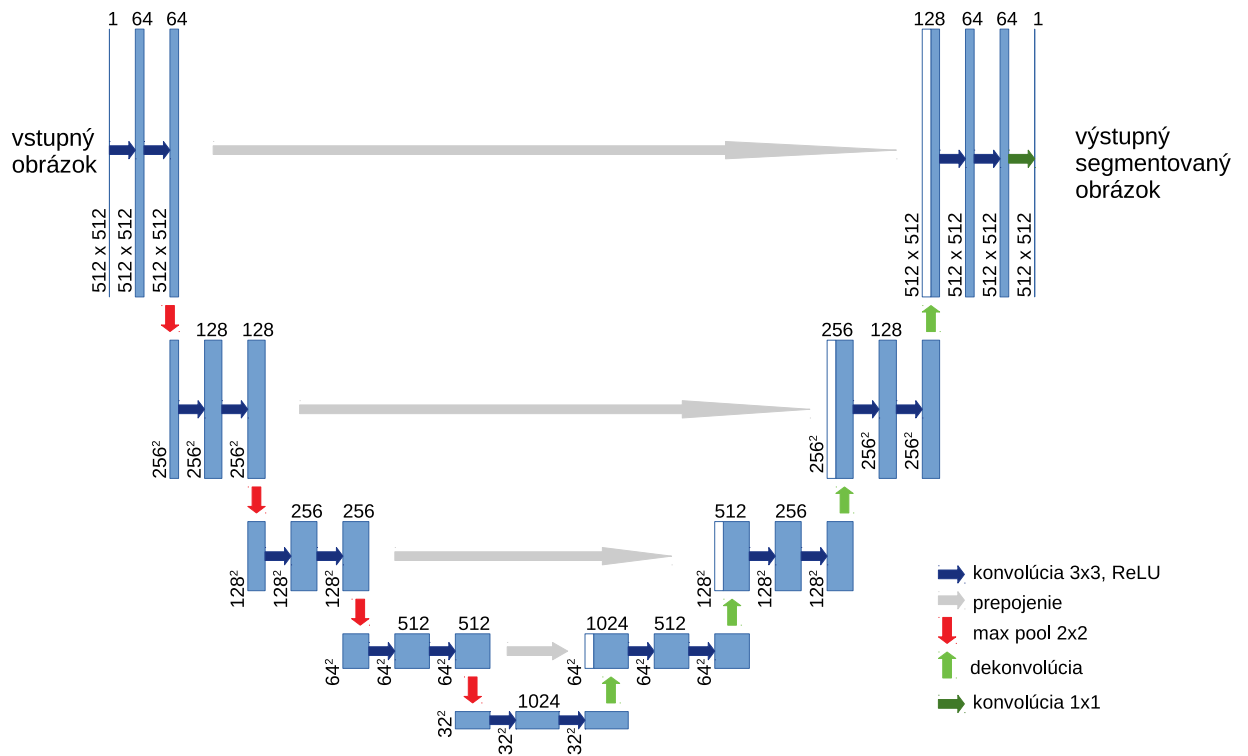
Filter podvzorkovacej alebo pooling vrstvy vykonáva matematickú operáciu nad oblasťami vstupného pola o rozmeroch $N \times N$ a z tejto oblasti je výstupom jedno výstupnú číslo. Filter sa postupne posúva po vstupnom poli s krokom o veľkosti N . Najpoužívanejším je maxpooling, ktorý vyberie z oblasti najväčšiu hodnotu, ale možností je viac [10]. Podstatou tejto vrstvy je to, že ak sieť pozná že v danej oblasti sa nachádza určitý tvar, tak nie je jeho presná poloha až taká podstatná, ako jeho pozícia voči ostatným tvarom. Týmto sa výrazne zníži počet parametrov siete a s tým aj potrebný výpočtový výkon.

UNet architektúra

UNet architektúra bola vytvorená pre segmentáciu bio-medicínskych snímok. Skladá sa z enkóderu a dekóderu a názov vychádza z jej tvaru pri grafickom znázornení. Celý autoenkóder pozostáva z postupne prepojených konvolučných a podvzorkovacích vrstiev. Na obr.1.3 je znázornený UNet, ktorý je použitý aj v praktickej časti tejto práce. Vstupné obrázky majú rozmery 512×512 pixelov. Každým prechodom konvolučnou vrstvou v enkóderi rastie hĺbka obrázku na základe počtu použitých filtrov. Po konvolučnej vrstve je vždy použitá ReLU vrstva a následne podvzorkovanie metódou max pool s rozmermi 2×2 . Pri tomto podvzorkovaní sa rozmery obrázku zmenšujú na polovicu. Týmto je dosiahnuté to, že informácia, ktorú filter získal z pôvodnej veľkosti, je po podvzorkovaní obsiahnutá na menšej oblasti. Navyše sa postupne zväčšuje recepčné pole, čo umožňuje filtrom konvolučnej vrstvy pohľad z väčšej perspektívy. Hĺbka obrázku, alebo počet kanálov sa prechodom cez enkóder zvyšuje, čo umožňuje rozoznávanie komplikovanejších tvarov.

Výstupom autoenkóderu je ale celý segmentovaný obrázok, o rovnakých rozmeroch ako na vstupe. Navyše počas prechodu enkóderom sa stráca informácia, kde sa rozoznané tvary a objekty nachádzajú v pôvodnom obrázku. Preto musí následovať opačná operácia. Úlohou dekóderu je obnoviť pôvodnú veľkosť obrázku a zistiť, kde sa objekty nachádzajú. Pre nadvzorkovanie existuje niekoľko techník, najpoužívanejšou je ale transponovaná konvolúcia, niekedy nazyvaná aj dekonvolúcia. Pri prechode dekóderom sa teda rozmery obrázkov zväčšujú a počet kanálov sa znižuje. Pre získanie presnejších pozícií objektov sa v každom kroku dekóderu používa ešte prepojenie, čo je spojenie výstupu transponovanej konvolúcie s výstupom konvolúcie na rovnakej úrovni enkóderu [1].

Siete UNet sú v posledných rokoch často používané a dosahujú veľmi dobrých výsledkov najmä v oblasti medicíny. Pre jednotlivé problémy sa používajú rôzne



Obr. 1.4: Architektúra UNet

variácie tejto architektúry, prispôbené charakteru problému. Príkladom je publikácia [51] z decembra roku 2018, v ktorej autori predstavujú H-DenseUNet. Táto metóda je vytvorená pre asistenciu doktorom pri diagnóze a plánovaní liečby rakoviny pečene. Jedná sa o automatickú segmentačnú metódu, ktorá prekonáva problém časovej a pamäťovej náročnosti pri spracovaní 3D snímok metód, ktoré boli používané dovtedy a takisto dosahuje lepšej presnosti. Pri skúmaní nádorových ochorení obličiek sa tiež používajú automatické segmentačné nástroje na snímkoch z Computed Tomography (CT) prístrojov a aj v tejto oblasti majú zastúpenie siete UNet [52]. V novembri 2019 bola predstavená ďalšia metóda založená na sieti UNet, konkrétne BiSC-UNet, ktorá segmentuje CT snímky obličiek, ktoré sa používajú aj pri príprave na operačné zákroky [53]. Pri predpoklade kolorektálnej rakoviny sa robí prehliadka tráviaceho traktu bezdrátorovu kapsulovou endoskopiou (WCE). Doktori následne musia manuálne prehliadnúť okolo 20 000 snímok. Tento proces je časovo veľmi náročný, preto bol v [54] predstavený diagnostický model využívajúci sieť UNet pre automatickú detekciu polypov v tráviacom trakte. Táto metóda dosahuje dostatočujúcich presností pre detekciu polypov s rýchlosťou 23,25 snímok za sekundu a tak by mohla poskytnúť efektívnu asistenciu pri diagnózach. Využitie sietí UNet siaha aj do úplne iných oblastí ako je medicíny, napríklad monitorovanie pobrežia. Segmentácia hladiny mora je kľúčová pri detekcii lodí prichádzajúcich k pobrežiu. V práci [55]

je použitý k riešeniu tejto úlohy model UNet v kombinácii s ResNet - Res-UNet. Model je testovaný na snímkoch získaných z aplikácie Google Earth[56] a dosahuje kvalitných výsledkov.

1.3 ODE neurónová sieť

Publikácia [2] zverejnená na konferencii NIPS v roku 2018 prvýkrát predstavuje ODE (Ordinary Differential Equations) neurónovú sieť, ktorá využíva obyčajné diferenciálne rovnice. Autori v nej prezentujú, že táto sieť dokáže počítat predikcie na dátach závislých na čase alebo časových radách presnejšie ako používané rekurentné neurónové siete. Príkladmi takýchto dát sú napríklad vývoj cien akcií na trhu, zmeny počasia, údaje z prístrojov na sledovanie životných funkcií pacienta, spotreba energií, teda všetky dáta, ktoré sú závislé na čase.

Obyčajné diferenciálne rovnice

Obyčajné diferenciálne rovnice [11] popisujú zmenu systému závislú na jednej nezávislej premennej veličine a túto zmenu možno popísať deriváciou. Riešením diferenciálnej rovnice je pôvodná funkcia, z ktorej vznikla táto derivácia. Príkladom diferenciálnej rovnice popisujúcej pozíciu pohybujúceho sa objektu je:

$$x(t + h) = x(t) + h \frac{dx}{dt},$$

kde x predstavuje pozíciu objektu, t je počiatkový čas, h je počet uplynulých sekúnd a derivácia pozície v závislosti na čase $\frac{dx}{dt}$ popisuje rýchlosť objektu.

Pre vyriešenie takejto diferenciálnej rovnice je potrebné zadať počiatkový stav systému a cieľový stav alebo bod, do ktorého chceme poznať priebeh pôvodnej funkcie. Pre riešenie sú najčastejšie používané numerické metódy. Jednou z najstarších a najjednoduchších je Eulerova metóda. Jej vstupom je diferenciálna rovnica a počiatkový stav. V tomto počiatkovom bode sa vypočíta tangens (alebo sklon) priamky, ktorá aproximuje hľadanú funkciu až po nasledujúci bod. V ďalšom bode sa vypočíta ďalší tangens priamky, tá sa v tomto bode pripojí na prvú priamku a takto sa pokračuje po všetkých zadaných bodoch, až k výslednej aproximácii pôvodnej funkcie do cieľového bodu. Matematický popis jedného kroku Eulerovej metódy je:

$$x(n + 1) = x(n) + h,$$

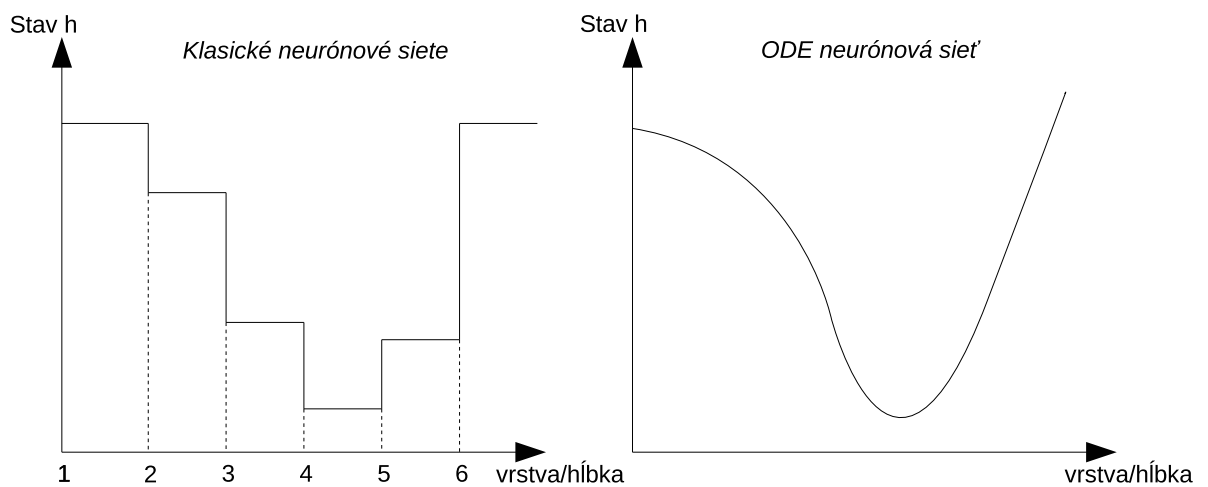
$$y(n + 1) = y(n) + hf(x(n), y(n)),$$

kde x sú body, v ktorých sa počíta tangens priamky, y sú výsledné hodnoty a h je veľkosť kroku medzi bodmi.

Aj pri dostatočnom počte bodov je ale táto metóda nepresná a v súčasnosti sa používajú presnejšie a komplikovanejšie metódy. Nástroje, ktoré používajú tieto metódy a dokážu veľmi efektívne počítať diferenciálne rovnice tohto typu sa nazývajú ODE kalkulátory (ODE solver).

Vrstvy

Hlavným rozdielom medzi ODE a ostatnými neurónovými sieťami je charakter vrstiev, viz. obr.1.5. Zakoreneným konceptom neurónových sietí je diskretný počet vrstiev, ktorý je určený vopred. Každá vrstva udržiava určitý stav, alebo hodnoty aktivácií a tento stav sa pri každom prechode vrstvou mení. Pri získavaní dát napríklad meraním sledovanej veličiny sa takisto zaznamenávajú hodnoty v časových intervaloch, teda výsledkom je konečný počet hodnôt. Svet ako taký ale funguje kontinuálne. Zmeny väčšiny veličín nie sú diskretné ale kontinuálne, počet hodnôt je teda nekonečný a žiadny prístroj nie je schopný zaznamenať všetky. Avšak ich vývoj možno popísať funkciou, ktorá ich všetky obsahuje. Výstupom ODE kalkulátora je práve takáto funkcia, ktorá popisuje vývoj stavu siete. Nie je teda špecifikovaný presný počet vrstiev siete, ale jedna kontinuálna vrstva, kde stav siete je funkciou hĺbky. Z toho vyplývajúca spojitosť gradientu sa používa na učenie neurónovej siete.

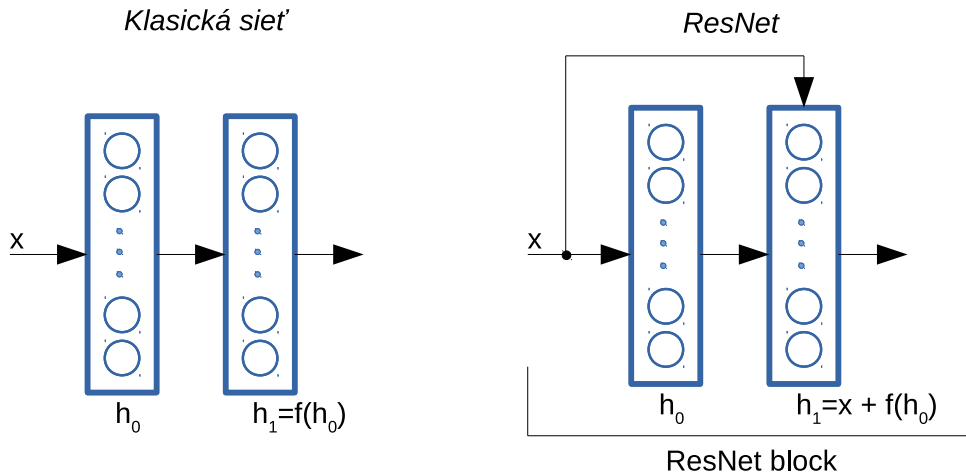


Obr. 1.5: Zmeny stavu neurónovej siete / ODE siete

ResNet

Určenie počtu vrstiev v klasických neurónových sieťach závisí na viacerých faktoroch. Zvyšovaním počtu vrstiev začne za určitou hranicou dochádzať k zhoršovaniu

presnosti alebo k zbytočnému narastaniu náročnosti. Tento problém sa podarilo vyriešiť spoločnosti Microsoft, keď v roku 2015 vyhrala klasifikačnú súťaž ILSVRC s hlbokou residuálnou neurónovou sieťou, alebo ResNet [12]. Pomocou ResNet bolo možné zostrojiť sieť s hĺbkou až 1000 vrstiev so stále sa zlepšujúcou presnosťou.



Obr. 1.6: Závislosť stavu - klasická neurónová sieť / ResNet

Hlavnou myšlienkou ResNetu je, že na vstup aktivačnej funkcie sa privádza súčet vstupu aj výstupu predchádzajúcej vrstvy. Pridaním vstupu sa zaistí, že sa nestratia dôležité informácie z vrstvy, ktorá tomuto bloku predchádzala. Na obr.1.6 je znázornený jeden ResNet blok. Výpočet stavu nasledujúceho bloku je:

$$x(n+1) = x(n) + f(x(n)),$$

kde f je funkcia, ktorá reprezentuje výpočet stavu danej vrstvy, napríklad konvolučnej. Po pridaní konštanty $h = 1$, ktorá nezmení pravdivosť vzorca, dostávame tvar:

$$x(n+1) = x(n) + hf(x(n)).$$

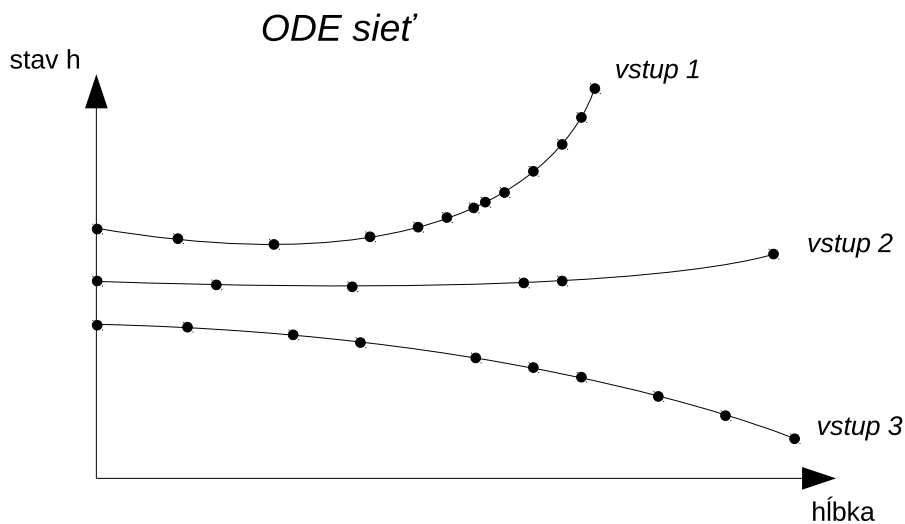
Táto upravená rovnica má rovnaký tvar ako diferenciálna rovnica, ktorú možno počítať Eulerovou metódou alebo iným ODE kalkulátorom. Preto všade, kde je použitý ResNet, možno použiť aj ODE siete. V publikácii [2] takto nahradili 6 ResNet blokov iba jedným ODE blokom, pričom dosiahli podobných výsledkov, ale s menším množstvom parametrov. Rozdiel je ten, že v sieti ResNet sa stav siete mení ako diskretná sekvencia konečných transformácií, zatiaľ čo v sieti ODE je to vektorové pole, ktoré kontinuálne transformuje stav siete.

ODE siete

Podobnosť vzorcov ResNetu a kroku Eulerovej metódy priviedla tvorcov k ODE sieťam. Dôležitou myšlienkou je fakt, že každú neurónovú sieť, ktorá používa pre-

pojenia (skip connections) ako ResNet, možno reprezentovať diferenciálnu rovnicu. Takúto rovnicu reprezentujúcu časť alebo celú neurónovú sieť potom dokážu veľmi efektívne riešiť adaptívne ODE kalkulátory. V úvode kapitoly 1.3 sa píše o veľkej presnosti ODE sietí na dátach závislých na čase alebo časových radách. Ak si ale predstavíme neurónovú sieť ako kontinuálny systém, bez diskretných vrstiev, ktorý sa mení v závislosti na hĺbke, tak za čas možno považovať hĺbku siete a vývoj siete je funkciou hĺbky. Takisto si možno predstaviť, že váhy siete sú funkciou hĺbky.

ODE neurónová sieť používa adaptívne ODE kalkulátory, ktoré sú oveľa komplikovanejšie ako Eulerova metóda [13] a riešenia diferenciálnej rovnice reprezentujúcej samotnú neurónovú sieť sú presnejšie. Vstupnými hodnotami ODE kalkulátoru sú počiatočný stav siete (napríklad vstupný obrázok) v čase 0 (na vstupnej vrstve), funkcia reprezentujúca časť alebo celú neurónovú sieť, a časové (hĺbkové) body, v ktorých má byť diferenciálna rovnica vypočítaná. ODE kalkulátor postupne vo vyhodnocovacích bodoch vyhodnocuje stav siete až k zadanému cieľovému bodu a výsledkom je funkcia, ktorá popisuje tento vývoj stavu. Vyhodnocovacie body sú v prípade klasického ResNetu jednotlivé vopred zadané vrstvy. Analogicky sú to pevne zadané body Eulerovej metódy. Avšak komplikovanejšie ODE kalkulátory sú adaptívne, a počet aj vzdialenosť vyhodnocovacích bodov si určujú samy podľa zložitosti aktuálneho vstupu. Pri ODE sieti sa teda nedá určiť presný počet vrstiev, keďže je premenný a takisto ich nemožno považovať za diskretné vrstvy v pravom slova zmysle, lebo funkcia spájajúca tieto body má kontinuálny charakter.



Obr. 1.7: Vyhodnocovacie body pre rôzne zložité vývoje stavu siete

Na obr.1.7 sú znázornené tri priebehy vývoja stavu ODE siete pre tri vstupy rôznej zložitosti, kde čierne body znázorňujú vyhodnocovacie body. Analogicky by pre ResNet znázorňovali tieto body jednotlivé vrstvy a stav vektor aktivácií danej

vrstvy, avšak medzi nimi by nebol kontinuálny priebeh. Dôležitou vlastnosťou je, že počet bodov, ktoré adaptívny ODE kalkulátor zvolí pre daný priebeh, žiadno neovplyvňuje pamäťovú náročnosť výpočtov. Tým pádom je možné vytvoriť neobmedzene hlbokú a hustú sieť s konštantnou pamäťovou náročnosťou. Avšak každý vyhodnocovací bod, resp. každé vyhodnocovanie zaberá nejaký čas, preto zvyšovaním počtu týchto bodov sa dá zlepšovať presnosť na úkor času, a naopak [2]. Aj keď počet bodov si určuje ODE kalkulátor nezávisle, možno ho ovplyvniť hodnotou tolerancie, ktorá je jedným zo vstupných parametrov ODE kalkulátoru. Ten zaisťuje, že výstup bude približne v danej tolerancii ku skutočnému výsledku. Zmenšovaním hodnoty tolerancie sa teda zvyšuje počet vyhodnocovacích bodov a tým aj časová náročnosť výpočtov.

Pri tréňovaní ODE siete dochádza k problému počas spätnej propagácie cez ODE kalkulátor. Klasickou propagáciou gradientu dochádza k zvyšovaniu pamäťovej náročnosti a takisto sa ku gradientom pridávajú numerické chyby. Preto sa pre spätnú propagáciu používa adjoint metóda [14], ktorá je súčasťou ODE kalkulátoru. Táto metóda počíta gradienty výpočtom ďalšej rozšírenej diferenciálnej rovnice smerom späť v čase/hĺbke. Tento proces má takisto konštantnú pamäťovú náročnosť, lebo nemusí ukladať žiadne aktivácie z medzi-výpočtov.

Stav vedy a techniky

Možnosť zlepšenia výsledkov residuálnych sietí pomocou ODE sietí vedie k stále pribúdajúcemu množstvu článkov a experimentov s touto technológiou. Aj keď sa stále jedná o novinku v oblasti hlbokého učenia, tak už bolo dokázaných niekoľko výhod pri jej použití. Napríklad v porovnaní s konvolučnými sieťami bola zistená lepšia robustnosť testovaná spracovaním vstupov z rôznymi typmi vstupných odchýlok [66]. Nové techniky, ktoré zlepšujú robustnosť klasifikátorov časových radov takisto využívajú ODE siete [67].

Postupne vznikajú aj rôzne rozšírenia a nástroje pre podporu. Takýmto nástrojom je napríklad framework OURS [57], ktorý umožňuje vývoj váh aj aktivácií na základe obyčajných diferenciálnych rovníc a táto rozšírená ODE sieť dosahuje na dátovej sade CIFAR-10 [58] lepších presností ako klasická ResNet sieť a nedávno prezentovaná ODE sieť. Ďalšiu nadstavbu predstavujú rozšírené (augmented) ODE siete [59]. Táto práca poukazuje na to, že existujú funkcie, ktoré ODE siete nedokážu reprezentovať, a ako riešenie predstavuje práve rozšírené ODE siete, ktoré sú stabilnejšie, lepšie zovšeobecňujú a majú menšie výpočtové nároky.

V ostatných typoch neurónových sietí sa často používajú regulačné mechaniky založené na zavádzaní šumu. Ich použitie v rámci ODE sietí bolo takisto otestované za vzniku SDE neurónovej siete (Neural Stochastic Differential Equation)[59]. Model

SDE siete dosahuje väčšej robustnosti, generalizácie a je odolnejší voči vstupným odchýlkam.

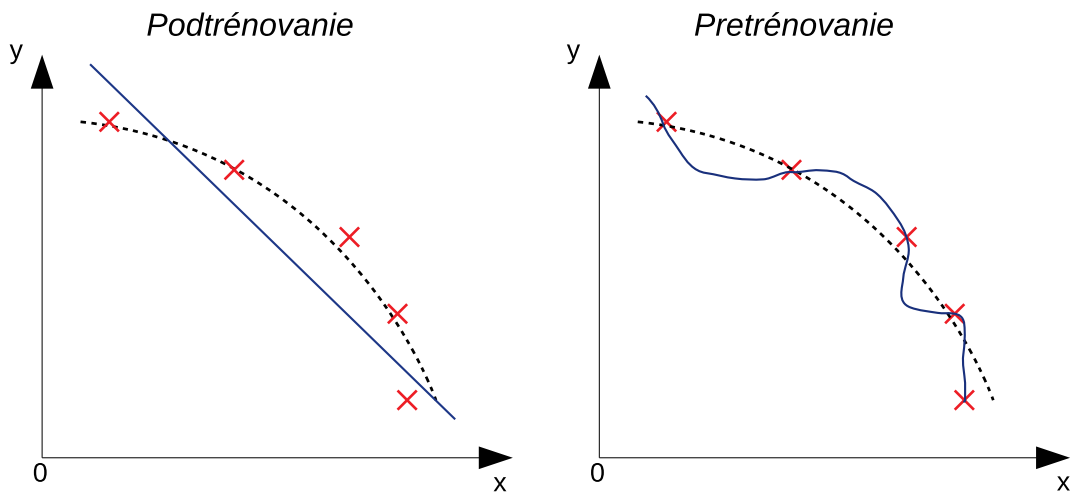
Príkladom využitia ODE neurónovej siete je metóda pre predpovedanie turbulencie tekutín [61]. Súčasný vývoj strojového učenia naštartoval snahu o získanie lepších presností tohto zložitého problému a prístup s použitím ODE siete prekonal doterajšie výsledky. Vďaka spojitému charakteru siete je možné ju využiť aj ako univerzálny aproximátor kontinuálnych funkcií [68]. Ďalšími príkladmi aktuálneho testovania ODE sietí sú predpovedanie dopytu elektrickej energie zo siete [69], kde sú ako hlavné výhody modelu uvedené časová nezávislosť a výpočtová škálovateľnosť, alebo detekcia, meranie a monitorovanie fyzikálnych aktivít veľkých oblastí zeme [70], kde bolo dosiahnutých výborných výsledkov aj pri použití malého množstva tréningových vzorkov.

1.4 Malé tréningové množiny

Pri riešení jednoduchých úloh, kde stačí poznať niekoľko údajov nie je veľké množstvo dát potrebné. Úlohy neurónových sietí ale bývajú často natoľko komplexné, že sa nedajú popísať jednoduchým a presným modelom. Obsahujú veľké množstvo faktorov, ktoré ovplyvňujú výsledok a ľudský mozog nie je schopný spracovať všetky závislosti a usporiadať ich do presného modelu. Počítač ale dokáže rozoznať a zachytiť tieto závislosti učením sa na veľkom množstve dát. Výsledný model takisto nieje úplne presný, ale vývoj stále nových architektúr sa k tomu snaží priblížiť. Množstvo tréningových dát je jedným z faktorov, ktorý ovplyvňuje presnosť takého modelu.

Malé tréningové množiny môžu spôsobovať niekoľko problémov pri tréningu. Na obr.1.8 sú znázornené problémy podtréningu a pretréningu modelu [47]. Červeno vyznačené body znázorňujú dátové body veličiny y , modré priebehy sú natréňované modely a čiarkovaná čiara predstavuje ideálnu aproximáciu funkcie, ktorá priebeh veličiny popisuje. Pri podtréningu je z dôvodu nedostatku informácií približne kvadratický priebeh aproximovaný ako lineárny, takže predpovede modelu na neznámych dátach budú veľmi nepresné. Pri pretréningu tvoria závislosť premennej kvadratické a vyššie mocniny. Model dosahuje na tréningových dátach veľmi vysokú presnosť, avšak na neznámych, testovacích alebo validačných bude veľmi nepresný. Je to z toho dôvodu, že model nie je dostatočne zovšeobecnený, aby dokázal pracovať s neznámymi dátami, je naučený len na tie, na ktorých trénoval. Pri zvyšovaní množstva tréningových dát sa počet takto nepresných modelov, ktoré by podobným spôsobom opisovali priebeh znižuje, pretože čím viac bodov má program k dispozícii, tým jednoduchšie sa odhaduje zovšeobecnený priebeh.

Neurónová sieť pracuje počas tréningu s miliónmi parametrov a preto potrebuje veľa iterácií, kým sa priblíži k ich ideálnej hodnote. Pri použití malej tréningovej



Obr. 1.8: Podtrénovanie a pretrénovanie modelu

množiny je veľká šanca, že sa model počas veľkého množstva iterácií pretrénuje. Dôležitým krokom pre zlepšenie procesu tréningu s malou tréningovou množinou je správna voľba optimalizačnej funkcie. Jej úlohou je minimalizácia stratovej funkcie siete a učenie siete zmenou váh. Pri obmedzenej dátovej množine sú vo väčšine prípadov najvhodnejšie optimalizéry, ktoré používajú metódy s adaptívnou mierou učenia (adaptive learning rate), ako je napríklad ADAM [48].

Augmentácia dátovej množiny

Augmentácia tréningovej dátovej množiny [49] je jednou z metód pre zlepšenie tréningového procesu, ktorá znižuje šancu pretrénovania modelu pri nedostatočnom množstve dát. Dosahuje toho jednoducho tým, že rozšíri tréningovú množinu o ďalšie dáta, ktoré sú vytvorené z pôvodných miernymi zmenami. Medzi najpoužívanejšie techniky v prípade obrazových dát patria:

- priblíženie alebo oddialenie obrazu,
- rotácia obrazu o určitý uhol,
- vertikálne alebo horizontálne posunutie obrazu,
- zmena jasnosti, kontrastu alebo ostrosti obrazu,
- a ďalšie.

2 Segmentácia obrazu

Počítačové alebo strojové videnie je vedný obor, ktorý sa venuje rozvoju technológií pre napodobenie častí komplexného ľudského vizuálneho systému a umožňuje počítačom rozoznávať a spracovávať objekty v obrazoch alebo videách podobným spôsobom ako ľudský mozog. Pre človeka je rozoznanie objektov v reálnom čase triviálnou úlohou, ale počítače to do nedávna dokázali len veľmi obmedzene. Avšak vďaka rozvoju umelej inteligencie a neurónových sietí v posledných rokoch bol dosiahnutý veľký pokrok v tejto oblasti, a v súčasnosti už dosahuje lepších presností ako ľudia v niektorých úlohách kde je nutná rýchla reakcia na obrazové vstupy [15].

K metódam spracovania obrazu patria hlavne klasifikácia obrazu, lokalizácia a segmentácia. Klasifikácia je proces, pri ktorom sa zaraďuje celý obraz do určitej kategórie, alebo je predpovedaná pravdepodobnosť že obraz patrí do určitej kategórie. Klasickým príkladom sú obrázky psov a mačiek, ktoré sú zaraďované do týchto dvoch kategórií [16].

Vyšší level predstavuje lokalizácia, ktorá detekuje objekt v obraze a zistí aj údaje o jeho lokalite v rámci obrazu. Interpretáciou umiestnenia daného objektu môže byť napríklad ohraničujúci rám okolo objektu, alebo farebné zvýraznenie všetkých pixelov, ktoré ho tvoria [17].

Najvyššou formou je segmentácia, pomocou ktorej je obraz rozdelený na segmenty. Segmenty potom reprezentujú objekty alebo ich časti, ktoré sú tvorené skupinami pixelov, nazývanými aj super-pixely. Každý jeden pixel obrazu je teda priradený do kategórie. Segmentácia môže byť sémantická alebo inštančná. Rozdiel je v tom, že inštančná oddeľuje od seba aj jednotlivé objekty rovnakej kategórie [19].

2.1 Metódy segmentácie

Existuje niekoľko segmentačných techník, pričom každá má určité výhody a nevýhody. V tejto časti sú zhrnuté a vysvetlené techniky založené na hľadaní oblastí, na detekcii hran a na zhlukovaní dát. V súčasnosti dosahujú výborných výsledkov segmentačné metódy využívajúce algoritmy hlbokého učenia, ktoré vo veľa prípadoch prekonávajú doterajšie presnosti [20]. Tieto metódy sú popísané v kap. 2.2.

Segmentácia založená na oblastiach

Tieto techniky rozdeľujú celý obraz na oblasti podľa určitého pravidla, ako napríklad rovnaká úroveň šedi pixelov, ktoré obsahuje. Závisia teda na spoločných vzoroch intenzity susedných pixelov, ktoré potom tvoria jednotlivé oblasti.

Najjednoduchšia metóda tohto typu je segmentácia prahovou hodnotou (threshold), ktorá priamo rozdeľuje obraz na základe rôznych odtieňov šedej jednotlivých objektov. Možno ju rozdeliť na lokálnu a globálnu prahovú metódu. Globálna rozdeľuje obraz na dve oblasti podľa jednej prahovej hodnoty, zvyčajne objekt v popredí a pozadie, medzi ktorými je výrazný kontrast. Lokálna prahová metóda využíva viacero prahových hodnôt pre rozdelenie obrazu na pozadie a niekoľko objektov v popredí. Výhody týchto metód sú jednoduché výpočty a rýchlosť, avšak nedokážu presne pracovať ak obraz neobsahuje výrazný kontrast medzi objektami.

Ďalším príkladom metódy tohto typu je segmentácia pomocou regionálneho rastu (regional growth segmentation). Cieľom je, podobne ako u prahovej metódy, zoskupenie pixelov s podobnými vlastnosťami, aby tvorili samostatnú oblasť. Prvým krokom je určenie centrálnych (seed) pixelov, ku ktorým sú následne pripájané podobné pixely z okolia. Kritérium podobnosti je prahová absolútna odchýlka medzi pixelom a centrálnym pixelom. Výhodou tejto metódy je, že často dokáže s dobrou presnosťou oddeliť oblasti s rovnakými charakteristikami a poskytuje tak dobré informácie o hraniciach objektov. Nevýhodou je, že je pomerne náročná na výpočet a náchylná na šumové hodnoty [21].

Segmentácia založená na detekcii hrán

Metódy založené na detekcii hrán využívajú diskontinuity určitej vlastnosti obrazu pre rozoznávanie hrán objektov a tým rozdeľujú obraz na jednotlivé objekty. Medzi dvomi susednými oblasťami alebo objektami s rôznymi odtieňmi šedej je vždy farebná hrana, čo znamená, že sa v danom mieste nachádza nesúvislosť hodnôt šedi. Rozoznaním týchto hrán možno definovať hranice objektov a tým určiť ich tvar. Metódy na detekciu hrán transformujú originálny obraz na obraz reprezentovaný hranicami objektov, ktorý obsahuje výrazne menej dát ako originál, ktoré treba následne spracovávať, ale stále si udržuje zásadné informácie o tvare objektov.

Nesúvislosti v obraze možno detekovať použitím derivačných operácií, a deriváty môžu byť počítané použitím diferenciálnych operátorov. Medzi často používané patria napríklad Prewittov operátor, Robertsov operátor, Sobelov operátor, Laplaceov operátor, Kirschov alebo Wallisov operátor [22]. Hrany objektov možno rozoznávať takisto pomocou filtrov a konvolúcií, ktorých princíp je popísaný v kap. 1.2.

Segmentácia založená na zhlukovaní

Pre segmentáciu obrazu môžu byť použité aj metódy zhlukovania dát. Zhlukovanie dát je jednou z techník strojového učenia, ktorá rozdeľuje dátové body do určitého počtu skupín alebo zhlukov. Dátové body sú rozdeľované na základe podobných vlastností.

Jedným z najpoužívanějších zhlukovacích algoritmov je K-means. Písmeno K reprezentuje počet vytvorených zhlukov, resp. počet centrálnych bodov. Centrálné body určujú stred zhluku, od ktorého sa počítajú vlastnosti ostatných bodov, v tomto prípade vzdialenosť k centrálnemu bodu. Postup celého algoritmu je nasledovný:

1. vytvorenie m-rozmerného priestoru, kde m je počet parametrov bodov
2. umiestnenie dátových bodov do tohto priestoru
3. náhodné umiestnenie centrálnych bodov do priestoru
4. výpočet vzdialenosti (napríklad Euklidovej) jednotlivých bodov ku každému centrálnemu bodu a priradenie k jednému z nich, čím sú vytvorené prvé zhluky
5. výpočet centra jednotlivých zhlukov a premiestnenie centrálnych bodov do týchto centier
6. nová iterácia, teda prepočet vzdialeností, vytvorenie nových zhlukov a premiestnenie centrálnych bodov

Tento proces pokračuje, až kým centrálné body prestanú meniť svoju polohu [23].

Výhodou tohto algoritmu je jeho rýchlosť a jednoduchosť a je takisto veľmi efektívny pri použití s veľkými dátovými množinami. Nevýhodou je, že počet zhlukov, teda parameter K musí byť dopredu určený a často je náročné odhadnúť ho správne [21].

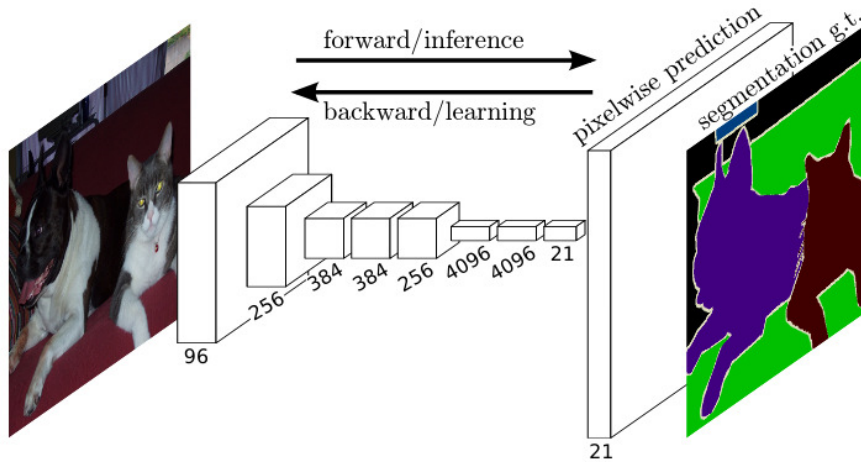
2.2 Architektúry hlbokého učenia

So stále narastajúcou popularitou hlbokého učenia vznikla v posledných rokoch celá rada nových segmentačných metód, ktoré využívajú práve algoritmy hlbokého učenia. Tieto metódy dosahujú výborných výsledkov. Veľká časť z nich je postavená na konvolučných neurónových sieťach. V tejto kapitole je popísaných niekoľko metód, ktoré boli kľúčové pri vývoji iných, alebo ktoré dosahujú v daných oblastiach v súčasnosti najlepších výsledkov.

Plne konvolučná sieť (FCN)

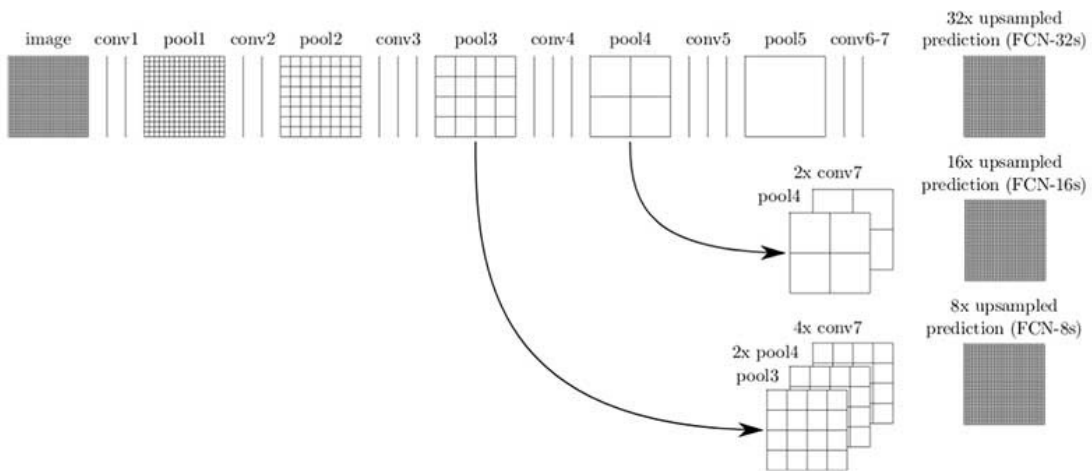
Jednou z najznámejších a najjednoduchších architektúr je plne konvolučná sieť [24]. Klasické konvolučné neurónové siete majú plne prepojené vrstvy, ktoré nedokážu spracovávať vstupné obrazy rôznych veľkostí, ale iba obrazy konkrétnej veľkosti. Autori siete FCN modifikovali existujúce architektúry konvolučných sietí tak, že nahradili plne prepojené vrstvy s plne konvolučnými vrstvami a tým dosiahli, že výsledkom siete je segmentačná mapa namiesto matice klasifikačných pravdepodobností [25]. FCN siete teda využívajú konvolučné vrstvy pre spracovávanie vstupov rôznych veľkostí. Vstupný obraz je najskôr postupne podvzorkovaný na menšie rozmery, pričom pomocou konvolučných vrstiev narastá hĺbka alebo počet kanálov obrazu.

Následne je obraz naopak nadvzorkovaný. Výsledkom tohto procesu je segmentačná mapa rovnakej veľkosti ako vstupný obraz.



Obr. 2.1: Použitie siete FCN pre sémantickú segmentáciu. Zdroj: [24]

Na obr. 2.1 je znázornený proces podvzorkovania a spätného nadvzorkovania vstupného snímku. Je tu takisto demonštrované použitie siete FCN na úlohe pracujúcej na úrovni pixelov (per-pixel task), konkrétne na sémantickej segmentácii.



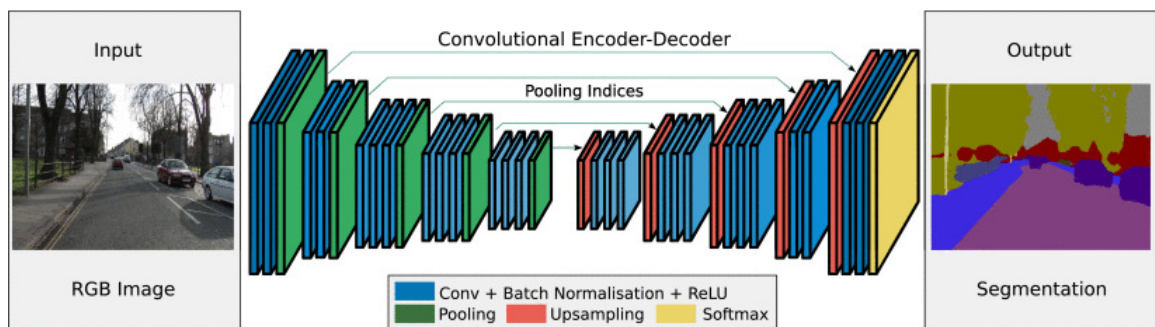
Obr. 2.2: Prepojenia siete FCN medzi hrubými informáciami z vyšších vrstiev a detailnými informáciami z hlbokých vrstiev. Zdroj: [24]

V sieti FCN vznikajú prepojenia (skip connections) tým, že výsledná príznaková mapa sa následne nadvzorkuje a informácie z nej sa zlúčia s informáciami z príznakovej mapy vyšších vrstiev. Týmto sú skombinované sémantické informácie z hlbokých vrstiev s informáciami z vyšších vrstiev, čím je dosiahnutá presnejšia a detailnejšia segmentácia.

Modely typu enkóder-dekóder

Veľká časť segmentačných techník používajúcich algoritmy hlbokého učenia pracuje s nejakou formou enkóderu a dekóderu. Jedna z prvých bola segmentácia založená na dekonvolúcii publikovaná v roku 2015 [26]. Tento model pozostáva z enkóderu s konvolučnými vrstvami a z dekonvolučnej siete, ktorej vstupom je príznaková mapa, z ktorej generuje mapu pravdepodobností jednotlivých pixelov.

Ďalším príkladom architektúry typu enkóder-dekóder je sieť SegNet [27], zobrazená na obr. 2.3. Podobne ako dekonvolučná sieť pozostáva z obdobného enkóderu a dekóderu nasledovaného klasifikačnou vrstvou jednotlivých pixelov. Hlavným rozdielom je spôsob, akým sieť nadvzorkuje príznakové mapy hlbších vrstiev. Využíva pri tom indexy vypočítané počas podvzorkovania napríklad pomocou metódy max-pool. Tieto indexy potom umožňujú nelineárne nadvzorkovanie. Sieť SegNet pracuje vo výsledku s oveľa menším množstvom parametrov.



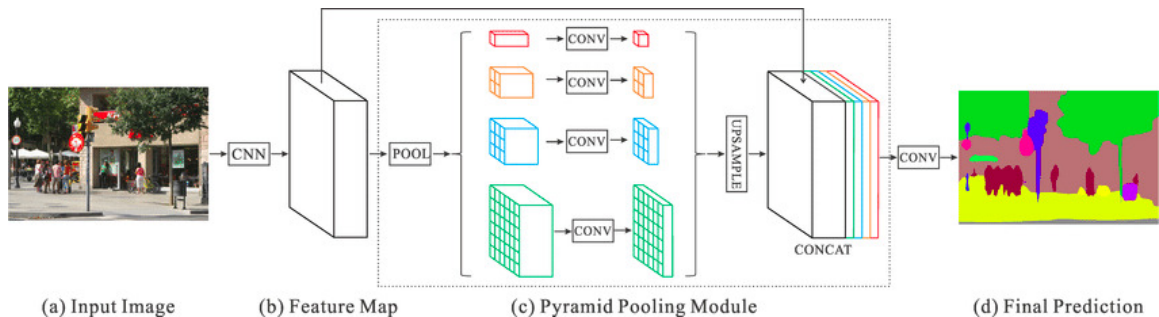
Obr. 2.3: Architektúra siete SegNet. Zdroj: [27]

Siete U-Net[1] a V-Net[28] takisto pozostávajú z častí enkóderu a dekóderu. Tieto architektúry boli vyvinuté za účelom segmentácie snímok pre oblasť medicíny a biomedicíny, ale sa začali uplatňovať aj mimo tieto sféry. Sieť U-Net je popísaná detailne v kap. 1.2. V-Net architektúra je takisto založená na plne konvolučnej sieti a bola vytvorená hlavne pre 3D segmentáciu medicínskych snímok.

Ďalšou U-Netu podobnou architektúrou je model Tiramisu [29]. Rozdiel je v tom, že Tiramisu model používa pre konvolúciu a pre transponovanú konvolúciu bloky siete DenseNet [30], ktorá v publikácii dosahovala výborných výsledkov na klasifikačných úlohách. Blok tejto siete pozostáva z niekoľkých konvolučných vrstiev, a jeho vstupom sú príznakové mapy všetkých predošlých vrstiev. Výsledná sieť je veľmi efektívna, pretože má jednoduchší prístup k informáciám všetkých vrstiev, avšak to spôsobuje veľkú pamäťovú náročnosť.

Viacúrovňové a pyramídové modely

Niektoré modely hlbokého učenia používajú metódy pre zlúčenie informácií z viacerých úrovní. Príkladom takéhoto modelu je PSPNet (Pyramid Scene Parsing Network) [31]. Vstupný obraz najskôr prechádza konvolučnou neurónovou sieťou, napríklad typu ResNet, kde je z poslednej konvolučnej vrstvy získaná príznaková mapa a tým sú zo vstupného obrazu extrahované vzory. Túto mapu využíva ďalej pyramídový podvzorkujúci modul (pyramid pooling module), ktorý rozozná jednotlivé vzory na rôznych úrovniach. V module je príznaková mapa podvzorkovaná na štyri úrovne a za každou nasleduje 1x1 konvolúcia pre zmenšenie dimenzií. Výstupy tohto pyramídového modulu sú nadvzorkované a spojené s pôvodnou príznakovou mapou. Týmto je dosiahnuté, že výstup obsahuje informácie lokálnej aj globálnej úrovne. Ná záver je výsledná príznaková reprezentácia poslaná do konvolučnej vrstvy pre získanie požadovanej predikcie na úrovni pixelov (per pixel prediction). Architektúra je znázornená na obr.2.4.



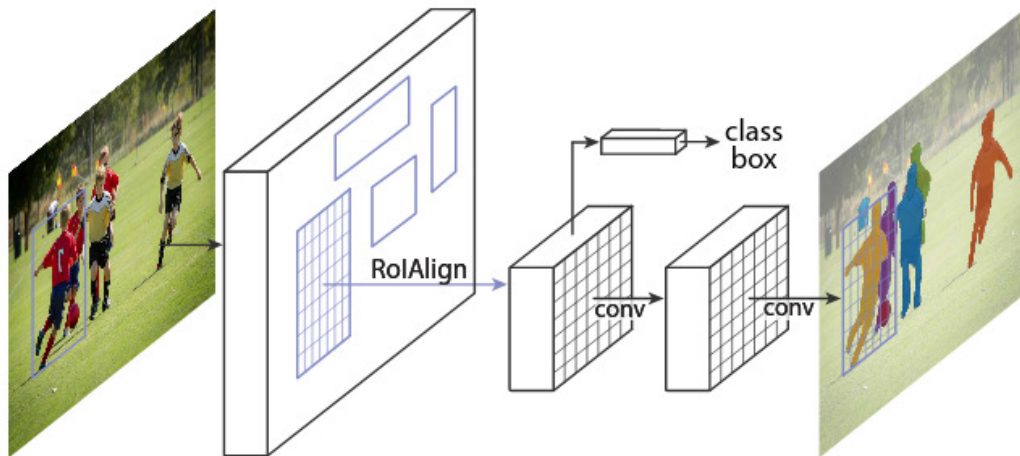
Obr. 2.4: Architektúra siete PSPNet. Zdroj: [31]

Ďalším príkladom architektúry, ktorá využíva viacúrovňovú analýzu je FPN (Feature Pyramid Network) [32], ktorá bola pôvodne vytvorená pre detekciu objektov, ale neskôr bola prispôbená a použitá aj na segmentačné úlohy [25]. Modely tohto typu sa stále vyvíjajú a toho dôkazom je množstvo aktuálnejších publikácií, ako napr.: DM-Net (Dynamic Multi-scale Filters Network) [33], APC-Net (Adaptive Pyramid Context Network) [34], MSCl (Multi-scale context intertwining) [35].

R-CNN modely

Architektúry typu R-CNN sa preukázali ako veľmi efektívne pri detekcii objektov. Niektoré rozšírenia boli modifikované pre inštančnú segmentáciu, teda úlohy vykonávajúce zároveň detekciu objektov a sémantickú segmentáciu a začali sa veľmi často pre riešenie týchto úloh používať [25]. Príkladmi týchto rozšírení sú Fast R-CNN [36], Faster R-CNN [37], alebo Mask R-CNN [38].

Model Faster R-CNN vráti pre každý objekt na vstupnom obraze označenie kategórie do ktorej spadá a súradnice obdĺžnikového rámu (bounding box), ktorý ohraňuje daný objekt. Na vstupe modelu sa nachádza konvolučná sieť, pomocou ktorej sú extrahované príznakové mapy z obrazov. Tieto príznakové mapy potom prechádzajú cez RPN (Region Proposal Network) sieť, ktorá vyberie oblasti, v ktorých sa nachádza objekt. Následuje RoI (Region of Interest) podvzorkovacia vrstva, ktorá oblasti upraví na rovnakú veľkosť. Následuje plne prepojená vrstva, ktorá klasifikuje a priradí ohraňujúce rámy jednotlivým objektom.



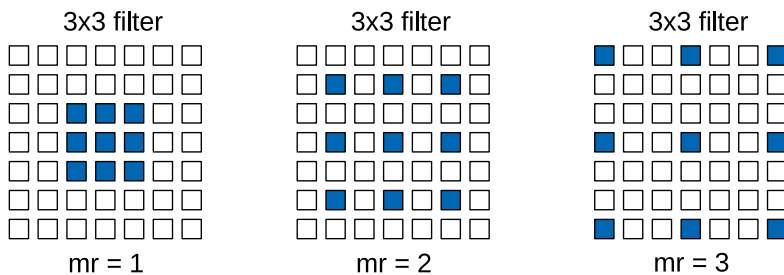
Obr. 2.5: Architektúra siete Mask R-CNN. Zdroj: [38]

Najnovším z uvedených modelov je Mask R-CNN (obr.2.5). Ten vznikol ako rozšírenie modelu Faster R-CNN a okrem klasifikácie a ohraňujúceho rámu je jeho výstupom aj maska objektu. Na vstupe modelu je takisto ako u Faster R-CNN sieť, ktorá extrahuje príznakové mapy zo vstupných obrazov, ale namiesto klasickej konvolučnej siete je to architektúra ResNet 101. Príznakové mapy sú ďalej použité v sieti RPN, ktorá predikuje, či sa v určitej oblasti nachádza nejaký objekt a výstupom siete sú oblasti alebo príznakové mapy, ktoré podľa predikcie objekt obsahujú. Tieto oblasti vstupujú do plne prepojenej siete, kde sú predikované kategórie a ohraňujúce rámy. Mask R-CNN model navyše v sieti RPN počíta parameter IoU (Intersection over Union), čo je prienik nad zjednotením s label obrazom s pravdivými rámami. Iba ak je tento parameter väčší ako 0,5 tak je oblasť použitá vo vetve pre výpočet masiek objektov. Predikcia masiek jednotlivých objektov je posledným krokom modelu [39].

Modely s rozšírenými (dilated) konvolúciami

Rozšírené konvolúcie sú často používané v aplikáciach, ktoré pracujú so segmentáciou v reálnom čase [25]. Pri takýchto aplikáciach je kľúčová rýchlosť a tá je priamo ovplyvnená výpočtovou náročnosťou.

Novým parametrom, ktorý prináša rozšírená konvolúcia je miera rozšírenia (dilation rate). Ten udáva mieru, akou sú rozšírené váhy použitého konvolučného filtra na recepcnom poli, s ktorým pracuje. Recepcné pole má normálne pri použití filtra s rozmermi 3×3 veľkosť 3×3 , pri použití filtra 5×5 veľkosť 5×5 apod. Avšak s využitím rozšírenej konvolúcie s mierou rozšírenia 2 a s filtrom s rozmermi 3×3 , bude recepcné pole rozšírené na veľkosť 5×5 , pričom počet parametrov ostáva 9 podľa veľkosti filtra, viz. obr.2.6. Týmto je dosiahnuté to, že recepcné pole je zväčšené bez navýšenia výpočtovej náročnosti.



Obr. 2.6: Rozšírená konvolúcia - recepcné polia pri rôznych hodnotách miery učenia (mr)

Jednými z najznámejších architektúr využívajúcich rozšírené konvolúcie sú modely DeepLab. Ako prvý bol vytvorený model DeepLabv1 [40], nasledovaný verziou DeepLabv2 [41]. Základom druhej verzie je niekoľko častí. V konvolučných vrstvách využíva rozšírené konvolúcie pre zmenšenie počtu potrebných parametrov pri výpočtoch. Ďalšou časťou je ASPP (Atrous Spatial Pyramid Pooling) vrstva, ktorá podvzorkuje konvolučné príznakové mapy na niekoľkých úrovniach s rôznym pomerom podvzorkovania a tým získava z príznakových máp lokálne informácie o objektoch a aj globálne informácie o celom kontexte obrazu. Špecifikom modelu DeepLabv2 je aj kombinácia metód hlbokých konvolučných sietí a pravdepodobnostných grafických modelov (probabilistic graphical models) pre lokalizáciu objektov. Ďalšou verziou tohto modelu je DeepLabv3 [42], ktorá kombinuje kaskádové a paralelné modely rozšírených konvolúcií. Do vrstvy ASPP je pridaná 1×1 konvolúcia a dávková normalizácia a všetky výstupy sú spojené a spracované ďalšou 1×1 konvolúciou. V roku 2018 bol predstavený model DeepLabv3+ [43]. Ten využíva architektúru typu enkóder-dekóder, ktorá zahŕňa rozšírené separovateľné konvolúcie a ako enkóder je použitý model DeepLabv3 [25].

Modely s rekurentnými neurónovými sieťami (RNN)

Všetky modely využívajúce algoritmy hlbokého učenia, ktorí boli popísané v tejto kapitole sú nejakým spôsobom založené na konvolučných neurónových sieťach. Avšak existujú aj modely založené na rekurentných neurónových sieťach, aj keď tieto siete boli primárne vytvorené pre iné úlohy ako segmentačné. RNN siete možno využiť pre modelovanie závislostí medzi pixelmi pre zlepšenie odhadu segmentačnej mapy [25]. Príkladmi takýchto architektúr sú napríklad model ReSeg [44], ktorý je postavený na sieti ReNet [45], alebo DA-RNN sieť (Data Associated Recurrent Neural Network) [46] vytvorený pre mapovanie 3D obrazu.

3 Popis experimentov

Cielom praktickej časti tejto práce je určenie najvhodnejšej metódy pre sémantickú segmentáciu obrazu s použitím dátovej množiny malej veľkosti. Použitá dátová množina ISBI pozostáva z 30 mikroskopických snímok mozgu octomilky. Prvou časťou experimentu je porovnanie UNet architektúry používanej v posledných rokoch, s jednoduchou architektúrou ODE siete. Ďalšou časťou je návrh novej architektúry pre segmentáciu obrazu s využitím metód, ktoré umožňujú efektívne strojové učenie aj v prípade malých dátových množín. Výsledky všetkých experimentov sú vyhodnotené v kap. 4.

3.1 ISBI dátová sada

V roku 2012 bola vyhlásená medzinárodná súťaž [50] v 2D segmentácii snímok mozgu zachytených elektrónovým mikroskopom (EM snímky). Nazýva sa ISBI challenge a jej cieľom bolo nájsť zlepšenie v oblasti rekonštrukcie neurónových obvodov. Víťazným sa stal tím s modelom založeným na konvolučnej neurónovej sieti a tento prístup hlbokého učenia sa postupom času stal štandardom pre segmentáciu EM obrázkov. Posielanie modelov na vyhodnotenie je ale možné dodnes a ľudia si tak môžu overiť ich presnosť. Táto súťaž bola náročnejšia v tom, že poskytuje záujmom iba obmedzenú tréningovú dátovú množinu, a tak je vhodnou voľbou pre účely tejto práce.

Pokrok v oblasti elektrónových mikroskopov viedol k zachyteniu väčších častí mozgového tkaniva a s tým aj k novým poznatkom o nervovom systéme. To viedlo k potrebe vývoju nových výpočtových technológií pre spracovanie a analýzu EM obrazov mozgu. V predchádzajúcej štúdií bolo zrekonštruovaných okolo 1000 neurónov za využitia 20 000 hodín ľudskej práce a výsledok aj tak nebol dostatočný. Pri získaní väčších obrazov je teda nevyhnutný vývoj automatických segmentačných techník a to bolo práve cieľom tejto súťaže.

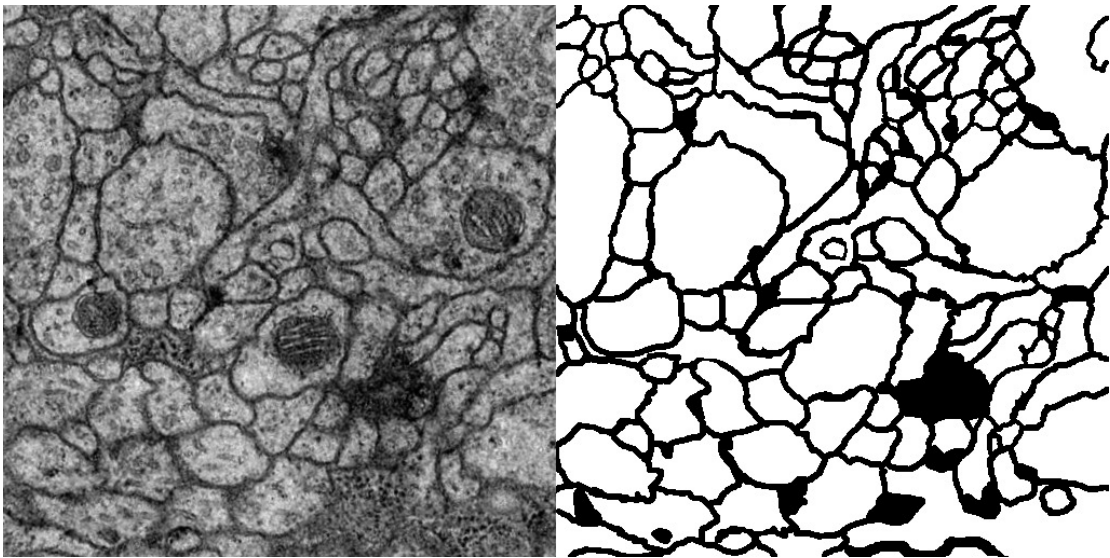
Dáta, ktoré sú poskytnuté záujmom sú postupné obrázky získané elektrónovým mikroskopom typu ssTEM o veľkosti 512x512 pixelov. Sú na nich zachytené časti nervového systému octomilky. Jednotlivé snímky zachytávajú oblasť o veľkosti 2x2x1,5 μ m s rozlíšením 4x4x50nm/pixel [50].

Dostupný dataset

Na obr.3.1 je jeden originálny obrázok z tréningovej množiny spolu s predpripravenou hraničnou mapou, ktorá slúži ako označenie (label) pri tréningu. Na webových stránkach súťaže je dostupná tréningová dátová množina aj testovacia, kde každá

obsahuje 30 obrázkov. V tréningovom datasete sú navyše spomínané hraničné mapy, ktoré boli vopred manuálne vytvorené expertmi na danú problematiku. Hraničné mapy pre testovaciu množinu neboli zverejnené, preto pre jej vyhodnotenie je nutné model odoslať a obratom získať výsledky.

Cieľom algoritmu je transformácia originálneho EM obrázku na čo najpresnejšiu hraničnú mapu, ktorá je definovaná ako binárny obraz, kde 1 indikuje pixel vnútri bunky a 0 hraničný pixel medzi bunkami. Nájdenie tejto mapy sa zhoduje so segmentáciou obrazu. Hľadanie hraníc je v tomto prípade náročnou úlohou, pretože hranice sú na viacerých miestach nejasné alebo rozmazané a navyše objekty vnútri buniek ako napríklad mitochondrie by mali byť programom ignorované.



Obr. 3.1: Originálny obrázok a hraničná mapa (ground truth boundary map)

Vyhodnocovanie výsledkov

Po natrénovaní modelu sú pre overenie presnosti použité snímky z dostupnej testovacej množiny. Po segmentácii sa musia pre vyhodnotenie presnosti odoslať organizátorom súťaže vo forme .tiff súboru. Organizátori snímky vyhodnotia, výsledky odošlú späť a zverejnia ich na oficiálnej stránke súťaže v tabulke výsledkov. Keďže sa jedná o hľadanie hraníc medzi jednotlivými bunkami snímku, jednoduché meranie chyby na úrovni pixelu (pixel error) by nebolo dostačujúce. Chybné klasifikovanie jedného pixelu ako hraničného nemusí spôsobiť celkovo zlý efekt na výsledné ohraničenie daného segmentu, pretože hrúbka hraníc môže byť rôzna a systém vyhodnocovania by mal byť dostatočne robustný na to, aby takého malé rozdiely vyhodnotil ako ne-dôležité. Preto tvorcovia zvolili metriku „Foreground-restricted Rand Scoring“ pre

oficiálne vyhodnocovanie a neskôr pridali aj metriku „Foreground-restricted Information Theoretic Scoring“ [50]. Prvá metrika je založená na Rand indexe, ktorý sa používa pre segmentáciu obrazu [64] a druhá je založená na variácii informácie, ktorá bola predstavená ako metrika pre zhlukovanie [63] a segmentáciu obrazu [62].

3.2 Porovnanie známych architektúr

V tejto časti sú popísané známe architektúry, ktoré boli použité v praktickej časti. Sieť UNet, ktorá je v súčasnosti používaná vo veľkej miere a dosahuje veľmi dobrých výsledkov v segmentačných úlohách, bola použitá pre porovnanie presnosti s jednoduchou architektúrou využívajúcou sieť ODE, ktorá je relatívne nová a málo používaná pre segmentáciu. Týmto bola overená možnosť použitia ODE siete pre tento typ úloh. Na základe výsledkov porovnania bolo následne rozhodnuté o využití tohto typu siete v návrhu novej architektúry pre segmentáciu dát s malými tréningovými množinami.

Spôsob tréovania

Tréovanie modelov prebieha na dostupnej dátovej množine s tým, že 24 snímok je použitých na tréovanie a 6 snímok na validáciu. Po každej epoche prebieha vyhodnotenie tréovacej aj validačnej presnosti na všetkých dostupných snímkoch. Pre vyhodnotenie presnosti bol použitý vyhodnocovací nástroj VISCERAL [65], ktorý počíta niekoľko metrík nad sadou snímok zoradených do .tiff súboru. V tejto práci sú využité metriky Dice a Accuracy. Dice koeficient je často používaná metrika pri segmentácii a je to pomer dvojnásobku zhodných pravdivých kladných pixelov medzi segmentovaným snímkom a hraničnou mapou (labelom) ku súčtu všetkých pixelov oboch snímok. Metrika presnosť (accuracy) je takmer totožná, akurát počíta aj s pravdivými zápornými pixelmi, je preto pre tento prípad smerodatnejšia.

Použitá UNet architektúra

Na obr. 1.4 je znázornená UNet architektúra, ktorá bola použitá pri tomto experimente. Hĺbka siete je 4, čiže pozostáva zo štyroch podvzorkovacích vrstiev, jedného mostu, a štyroch nadvzorkovacích vrstiev. Sieť je ukončená výstupnou konvolučnou vrstvou, ktorej výstupom je jeden segmentovaný obrázok v pôvodnej veľkosti 512x512 pixelov. Parametre pri tréovaní:

- počet epoch: 100
- veľkosť dávky: 1
- počet tréovacích obrázkov: 24
- počet validačných obrázkov: 6

- veľkosť vstupných obrázkov: 512x512p
- optimalizačná funkcia: Adam
- miera učenia: 0,01

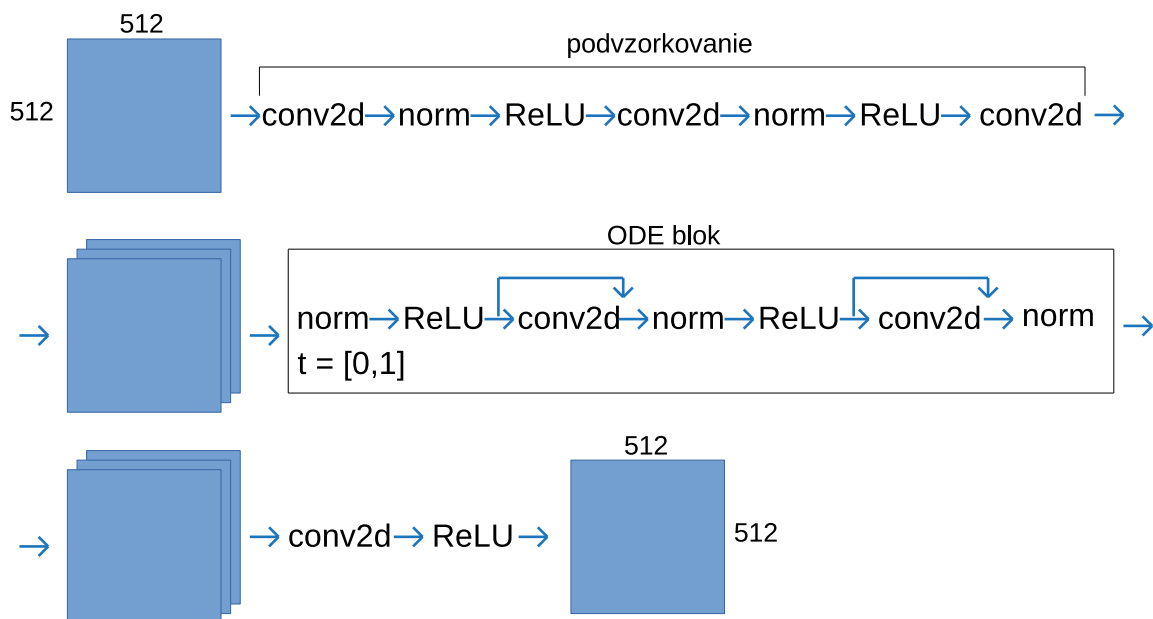
Miera učenia bola určená experimentálne. Pri hodnotách 0,001 a 0,0001 sieť dosahovala horších výsledkov.

Použitá ODE architektúra

Pre testovanie ODE siete na segmentáciu obrazu bola použitá ODE sieť, ktorú prví predstavitelia v publikácii [2] použili na klasifikáciu Modified National Institute of Standards and Technology (MNIST) databáze ručne písaných číslíc. Táto sieť bola mierne modifikovaná pre segmentáciu obrazu a dostupnú ISBI dátovú množinu. Jedná sa o pomerne jednoduchú sieť, s dvomi podvzorkovacími vrstvami na vstupe a jedným ODE blokom, ktorého vstupnou funkciou je malá residuálna sieť s dvomi ResNet blokmi. Tieto ResNet bloky pozostávajú z konvolučných, normalizačných a aktivačných vrstiev. Celá sieť je znázornená na obr.3.2.

Pre ODE model bola experimentálne zistená najlepšia presnosť pri použití miery učenia 0,001. K spätnej propagácii cez ODE kalkulator je využívaná adjoint metóda, ktorá je súčasťou dostupných ODE kalkulatorov. Hodnotou tolerancie možno ovplyvňovať požadovanú presnosť výpočtov, avšak s menšou hodnotou tolerancie narastá doba tréningovania, preto bude s menšími toleranciami otestovaný len najpresnejší model. Parametre pri tréningovaní:

- počet epoch: 100
- veľkosť dávky: 1
- počet tréningových obrázkov: 24
- počet validačných obrázkov: 6
- veľkosť vstupných obrázkov: 512x512p
- optimalizačná funkcia: Adam
- miera učenia: 0,001
- hodnota tolerancie: 0,001



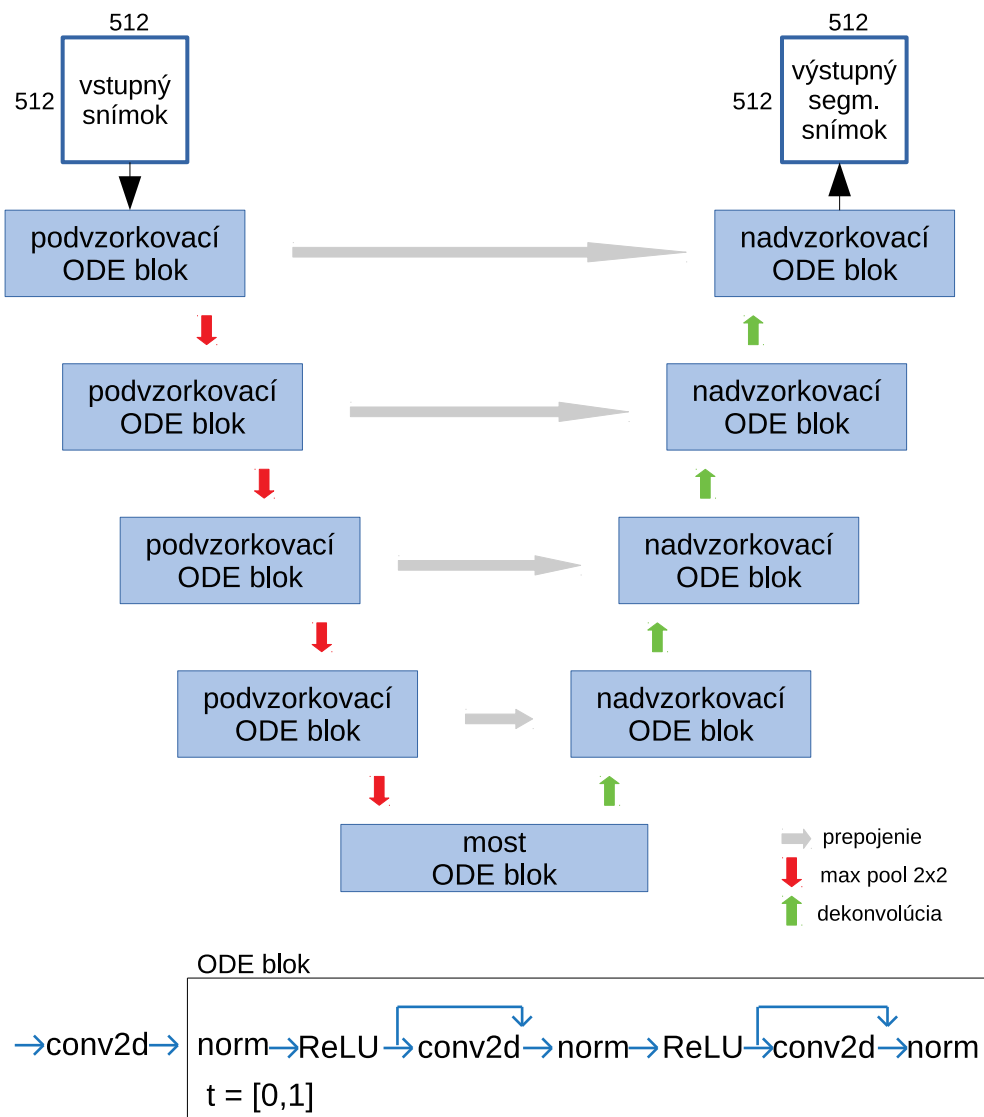
Obr. 3.2: Architektúra siete ODE

3.3 Návrh novej architektúry

Návrh novej architektúry a porovnanie jej presností s existujúcimi modelmi na ISBI dátovej množine je hlavnou časťou praktickej časti tejto práce. Navrhnuté boli 2 varianty architektúry využívajúcej kombináciu UNetu a siete ODE, na základe dobrých výsledkov z predchádzajúceho porovnania, čím sú využité výhody oboch sietí.

ODE-UNet

Prvý návrh novej architektúry pre efektívnu segmentáciu obrazu s dátovou množinou malej veľkosti je zobrazený na obr. 3.3. Je to UNet architektúra využívajúca ODE kalkulátory pre výpočet jednotlivých blokov, takže každá časť pôvodnej UNet siete je vstupnou funkciou pre ODE kalkulátor. Táto vstupná funkcia sa skladá z dvoch konvolučných vrstiev a niekoľkých normalizačných a aktivačných vrstiev. Pred každým ODE blokom sa nachádza konvolučná vrstva, ktorá vykonáva zmenu počtu filtrov, ktorá je daná pre vrstvu autoenkóderu. Táto zmena sa musí vykonať mimo samotný ODE blok, pretože vstup aj výstup ODE kalkulátoru musí mať rovnaké množstvo parametrov. V tomto návrhu sú využité výhody aj UNet architektúry aj presnosti ODE siete. Pre získanie čo najlepšieho modelu tejto architektúry bolo v práci otestovaných niekoľko variácií s rôznymi nastaveniami, usporiadaniami a vylepšeniami. Výsledky jednotlivých modelov sú v kapitole 4.1.



Obr. 3.3: Autoenkóder využívajúci kombináciu ODE a UNet siete

Parametre pri trénovaní siete ODE-UNet:

- počet epoch: 100
- veľkosť dávky: 1
- počet trénovacích obrázkov: 24
- počet validačných obrázkov: 6
- veľkosť vstupných obrázkov: 512x512p
- optimalizačná funkcia: Adam
- miera učenia: 0,001
- hodnota tolerancie: 0,001

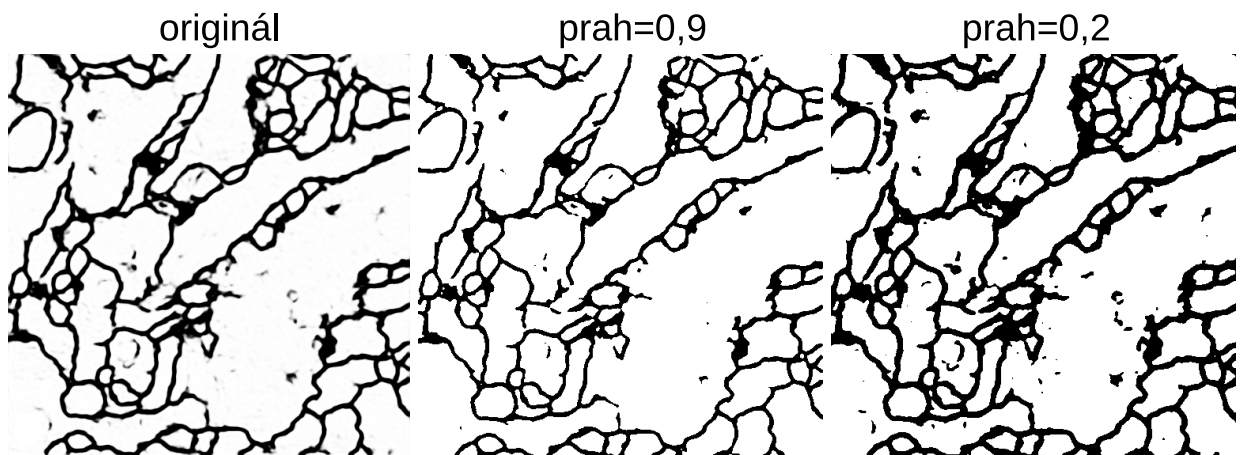
Metódy použité pri testovaní rôznych modelov ODE-UNet architektúry:

- **ConcatConv2d**

Jedná sa o rozšírenie 2d konvolučnej vrstvy, ktoré použili aj autori [2] pre demonštráciu ODE siete. Týmto rozšírením sa vytvoria prepojenia medzi konvolučnými vrstvami ODE bloku typické pre ResNet blok(1.6). Keďže ODE sieť bola vytvorená práve na základe toho, že dokáže vďaka ODE kalkulatorom efektívne počítať siete, ktoré môžu byť reprezentované ako residuálne, tak možno očakávať lepšiu presnosť pri použití tohto typu konvolučnej vrstvy v rámci ODE blokov.

- **Prahovanie (Thresholding)**

Výstupom modelu je snímok reprezentovaný dvojrozmerným polom normalizovaných hodnôt, kde každá hodnota reprezentuje odtieň šedej daného pixelu. Cieľom ISBI dátovej sady je hľadanie hraníc medzi bunkami, alebo hľadanie celistvých radov čiernych pixelov ohraničujúcich biele pixely, teda vnútro buniek. Možno teda predpokladať, že niektoré hodnoty vnášajú do výsledku neistoty.



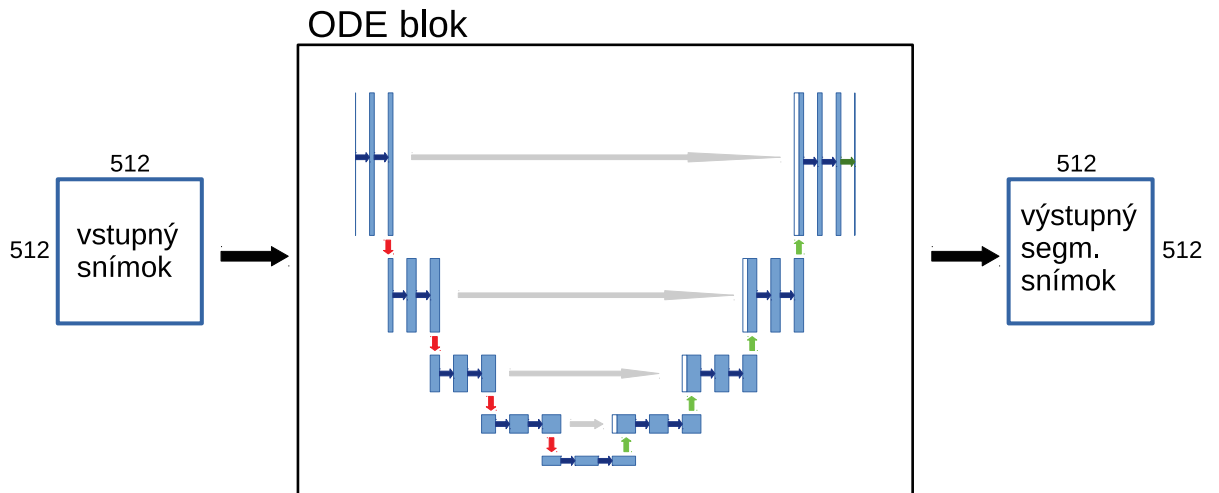
Obr. 3.4: Výsledky prahovania výsledného snímku

Prahovaním možno dosiahnuť to, že sa všetky hodnoty na základe prahovej hodnoty menia na 0 alebo 1, čiže na biely alebo čierny pixel. Na obr.3.4 je znázornený výsledný testovací snímok siete ODE-UNet a ten istý snímok s využitím prahových hodnôt 0,9 a 0,2. Všetky pixely s hodnotou menšou ako je prahová sú vyhodnotené ako 0 a všetky s väčšou hodnotou ako 1.

UNet-1-ODE

Tento návrh takisto využíva výhod architektúry UNet a aj ODE kalkulatorov, ale namiesto ODE blokov v každej vrstve UNet siete, pozostáva z jedného ODE bloku, ktorého vstupnou funkciou je celá UNet sieť. Priebeh siete je teda počítaný ako

jedna veľmi zložitá diferenciálna funkcia, riešená pomocou ODE kalkulatoru. Architektúra je znázornená na obr.3.5. Pri tréovaní bolo otestovaných niekoľko modelov, s použitím klasickej 2d konvolúcie alebo rozšírenej 2d konvolúcie. Keďže sa jedná o veľmi zložitú funkciu vstupujúcu do ODE bloku, možno predpokladať, že parameter tolerancia bude výrazne ovplyvňovať presnosť modelu, preto boli otestované rôzne hodnoty. Výsledky jednotlivých modelov sú v kap.4.1.



Obr. 3.5: Návrh architektúry UNet-1-ODE

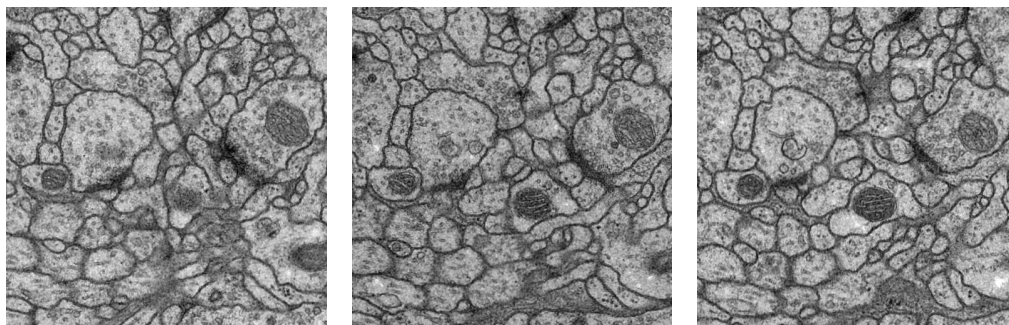
Parametre pri tréovaní siete UNet-1-ODE:

- počet epoch: 100
- veľkosť dávky: 1
- počet tréovacích obrázkov: 24
- počet validačných obrázkov: 6
- veľkosť vstupných obrázkov: 512x512p
- optimalizačná funkcia: Adam
- miera učenia: 0,0001

4 Výsledky a diskusia

K výsledkom je dôležité poznamenať niekoľko faktov. Všetky výsledky môžu byť zaťažené miernou chybou spôsobenou ľudským faktorom, keďže hraničné mapy slúžiace ako label snímok sú vytvorené manuálne niekoľkými expertmi. Takisto cieľom ISBI súťaže bolo nájsť bloky, resp. hranice jednotlivých buniek na snímku s tým, že organely vo vnútri buniek by mali byť ignorované. Od toho sa odvíja aj spôsob vyhodnocovania výsledkov a použité metriky, preto porovnanie so segmentačnými metrikami použitými pri tréningu a validácii nie sú úplne objektívne [50].

Na všetkých grafoch má tréningová a validačná presnosť takmer rovnaký priebeh. To je spôsobené tým, že snímky v dostupnej tréningovej sade sú veľmi podobné (viz. obr.4.1) a aj keď sú validačné snímky vyberané náhodne, podobnosť je stále značná. Časy tréningu zahŕňajú aj vyhodnocovanie niekoľkých metrick nad všetkými dostupnými snímkami, čo trvá určitú dobu.



Obr. 4.1: Podobnosť troch tréningových snímok

4.1 Jednotlivé experimenty

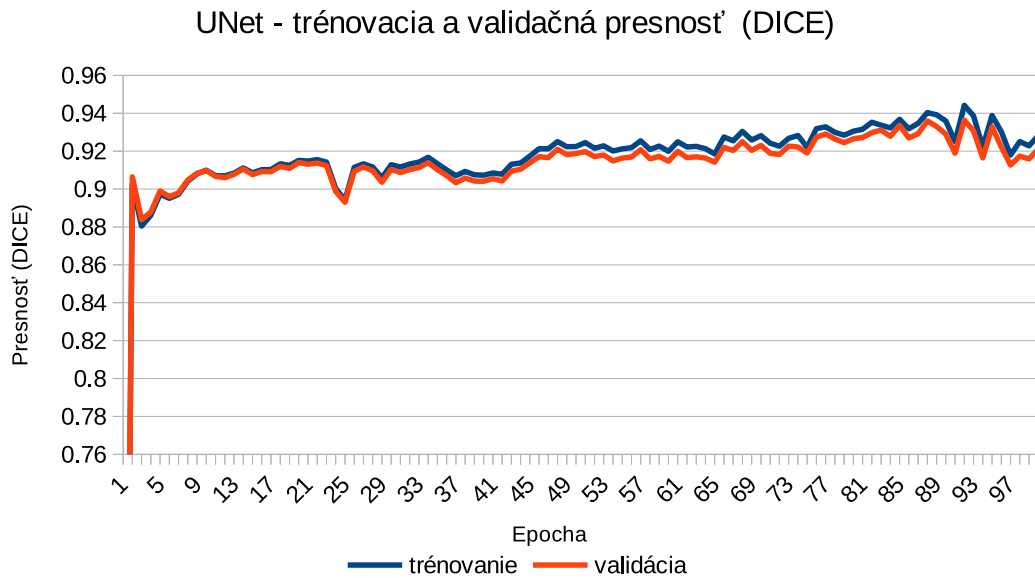
V tejto časti sú zobrazené a okomentované výsledky jednotlivých experimentov, rozdelené podľa použitého modelu. V rámci niektorých modelov bolo testovaných viacero parametrov, alebo bola experimentálne zisťovaná optimálna hodnota tréningových parametrov. Všetky výsledky sú zobrazené v príslušných grafoch a tabulkách.

UNet

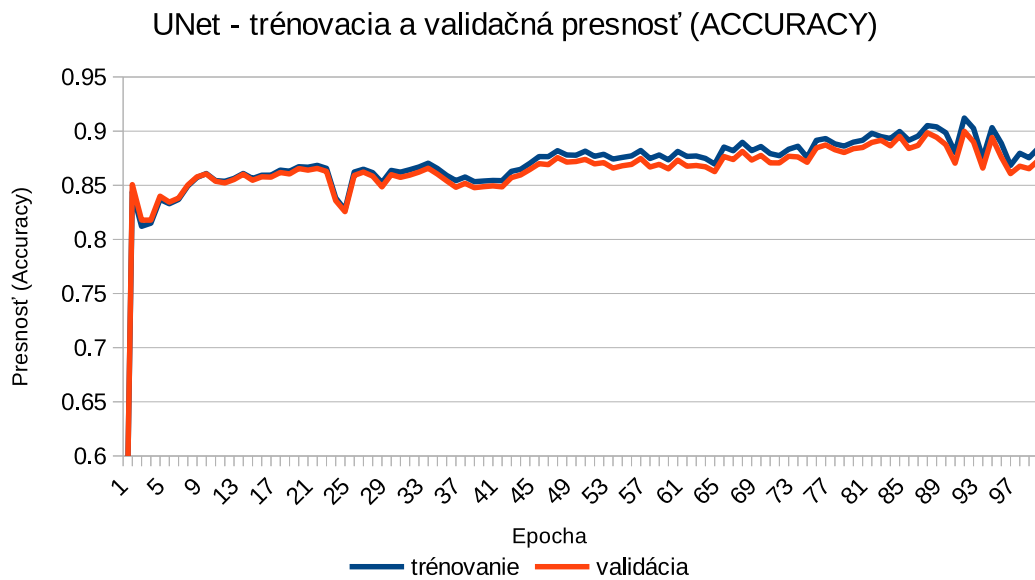
Priebeh tréningu a validácie UNet siete je znázornený na obr.4.2 a na obr.4.3. Presnosť vysoko stúpala hneď po druhej epoche a následne sa pomaly a mierne zlepšovala. Výsledné hodnoty siete sú nasledovné:

- čas tréningu: 24,27 min
- pamäťová náročnosť: 5207 MB

- najlepšia validačná presnosť dice: 0,936376
- najlepšia validačná presnosť accuracy: 0,899853
- Rand skóre: 0,902364
- Informačné teoretické skóre: 0,958347



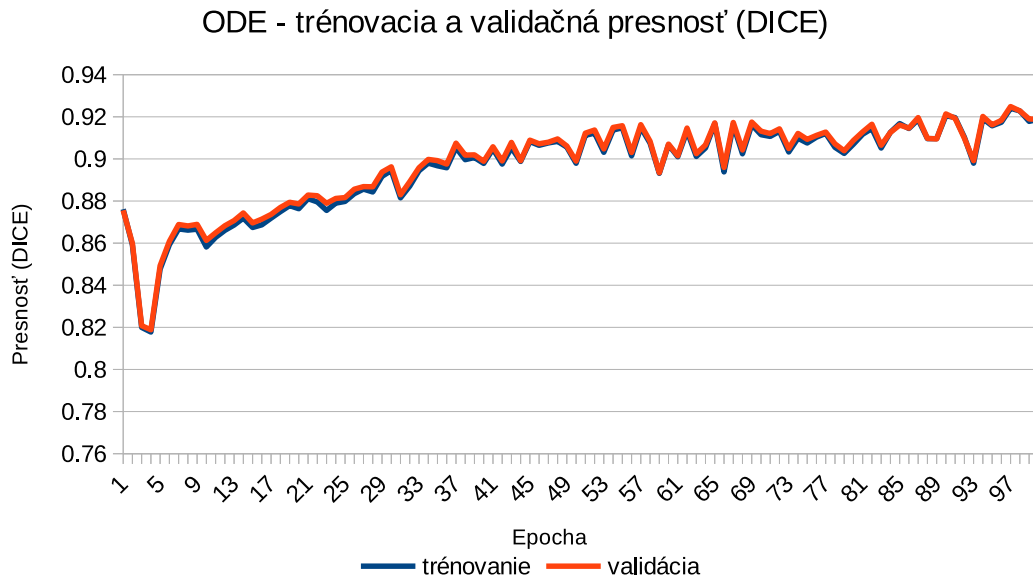
Obr. 4.2: Graf trénovacej a validačnej presnosti DICE siete UNet



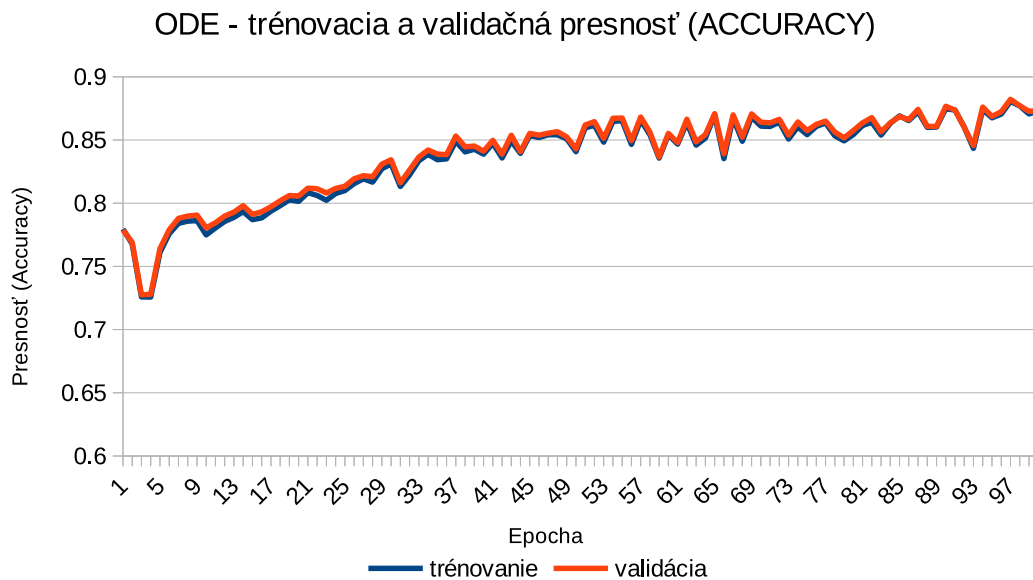
Obr. 4.3: Graf trénovacej a validačnej presnosti Accuracy siete UNet

ODE

Priebeh tréovania a validácie siete ODE je znázornený na obr.4.4 a na obr.4.5. Presnosť stúpala na pomerne vysokú hodnotu hneď po prvej epoche. Následne mierne stúpala a najlepšiu hodnotu dosiahla v posledných epochách tréovania.



Obr. 4.4: Graf tréovacej a validačnej presnosti DICE siete ODE



Obr. 4.5: Graf tréovacej a validačnej presnosti Accuracy siete ODE

Výsledne hodnoty modelu sú nasledovné:

- čas tréovania: 452,6 min
- pamäťová náročnosť: 4895 MB
- najlepšia validačná presnosť dice: 0,924852
- najlepšia validačná presnosť accuracy: 0,882176
- Rand skóre: 0,927354
- Informačné teoretické skóre: 0,967052

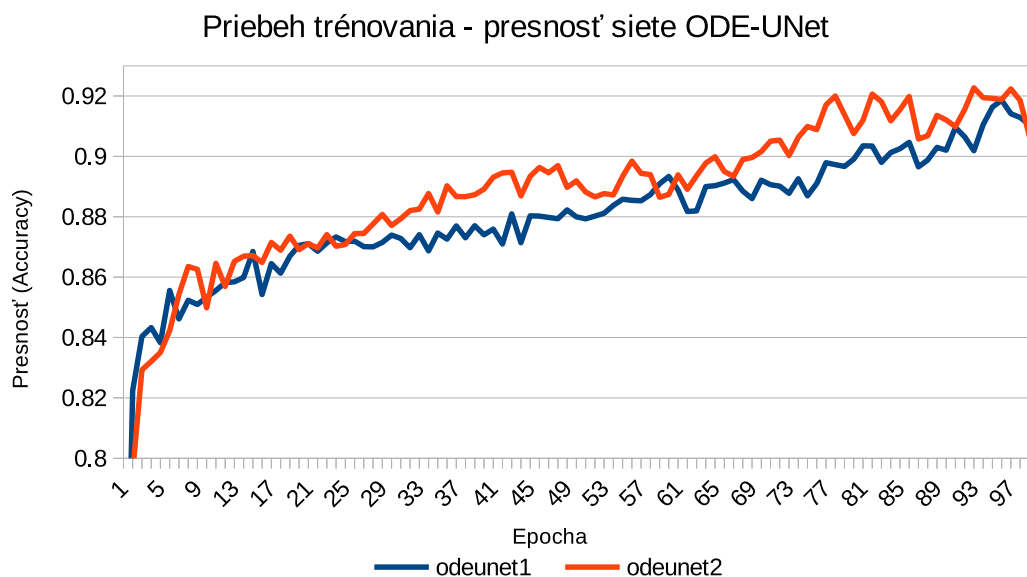
ODE-UNet

V rámci tejto architektúry bolo otestovaných niekoľko modelov s rôznymi nastaveniami parametrov a malými obmenami jednotlivých vrstiev ODE blokov. Výstupom týchto bodov je model architektúry ODE-UNet s usporiadaním a nastavením parametrov dosahujúci najlepších presností.

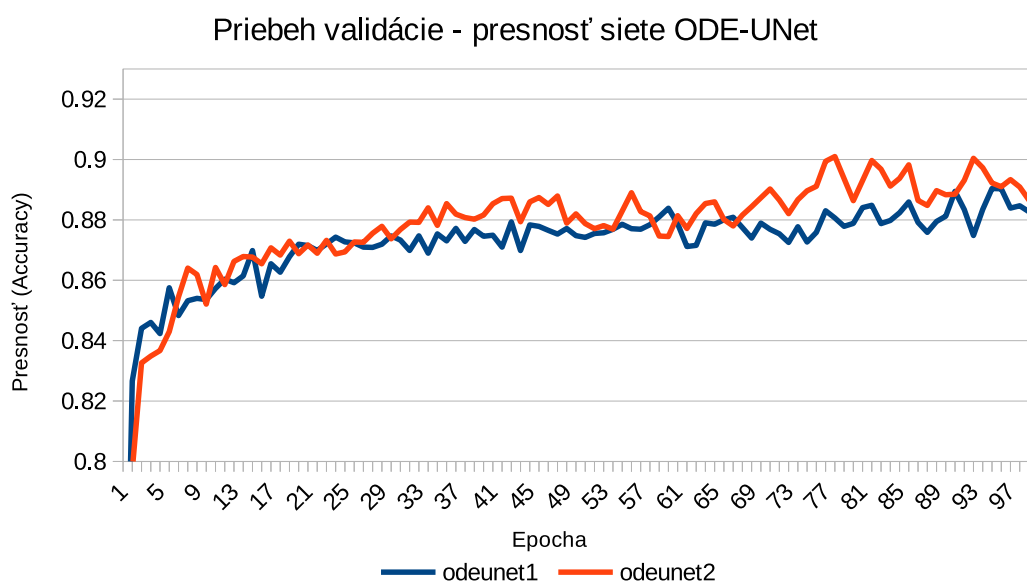
1. Ako prvý bol otestovaný model, ktorý využíva klasické 2d konvolúcie. Výsledne hodnoty modelu sú nasledovné:
 - čas tréovania: 319,9 min
 - pamäťová náročnosť: 8607 MB
 - najlepšia validačná presnosť dice: 0,932056
 - najlepšia validačná presnosť accuracy: 0,894390
 - Rand skóre: 0,880599
 - Informačné teoretické skóre: 0,938791
2. V tomto modeli boli nahradené klasické 2d konvolučné vrstvy ich rozšírenou verziou zretazené konvolučné vrstvy (concatenated). Takisto bola do každého ODE bloku pridaná jedna normalizačná a jedna aktivačná vrstva. Výsledne hodnoty modelu sú nasledovné:
 - čas tréovania: 403,9 min
 - pamäťová náročnosť: 8471 MB
 - najlepšia validačná presnosť dice: 0,938580
 - najlepšia validačná presnosť accuracy: 0,905035
 - Rand skóre: 0,950272
 - Informačné teoretické skóre: 0,978061

Na grafoch 4.6 a 4.7 je znázornené porovnanie priebehov tréovacej a validačnej presnosti modelov ODE-UNet z prvých dvoch bodov. Pri použití rozšírených 2d konvolúcií dosahuje model už po niekoľkých epochách výrazne lepších výsledkov.

3. V tomto bode bolo otestované prahovanie výsledných snímok a následne boli porovnané testovacie výsledky pre jednotlivé prahové hodnoty, uvedené v tab.4.1. Presnosť s rastúcou prahovou hodnotou postupne klesala voči ori-



Obr. 4.6: Pribeh tréovacej presnosti modelu odeunet1 (s jednoduchými 2d konv.) a modelu odeunet2 (so zretazenými 2d konv.)



Obr. 4.7: Pribeh validačnej presnosti modelu odeunet1 (s jednoduchými 2d konv.) a modelu odeunet2 (so zretazenými 2d konv.)

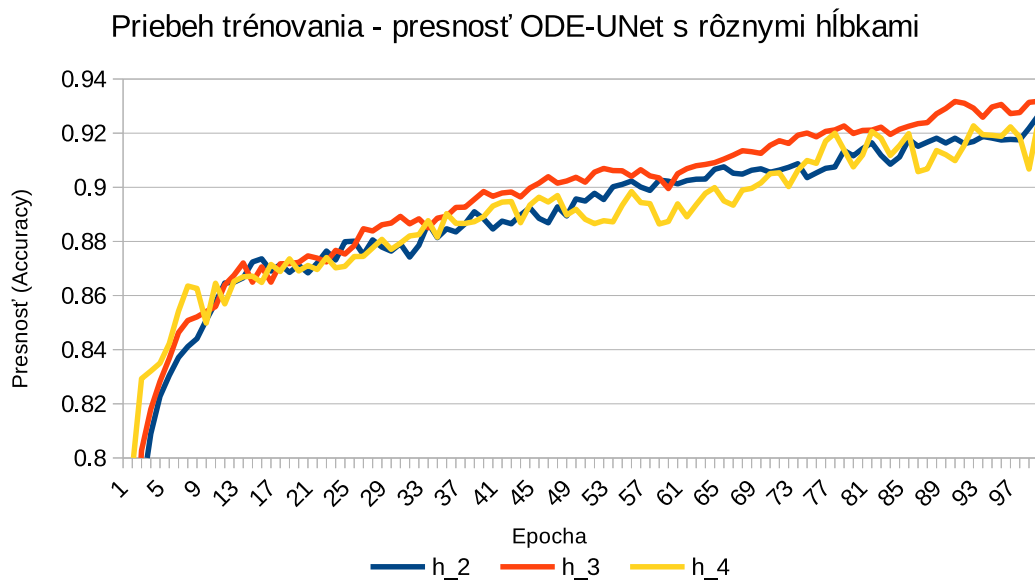
ginálnym snímkom, z čoho vyplýva, že metóda prahovania nie je vhodná pre tento problém.

4. Jedným z parametrov, ktorým možno pozmeniť tvar ODE-UNet architektúry je hĺbka UNet siete. Jeho zmenou nastane aj zmena presnosti siete, preto je

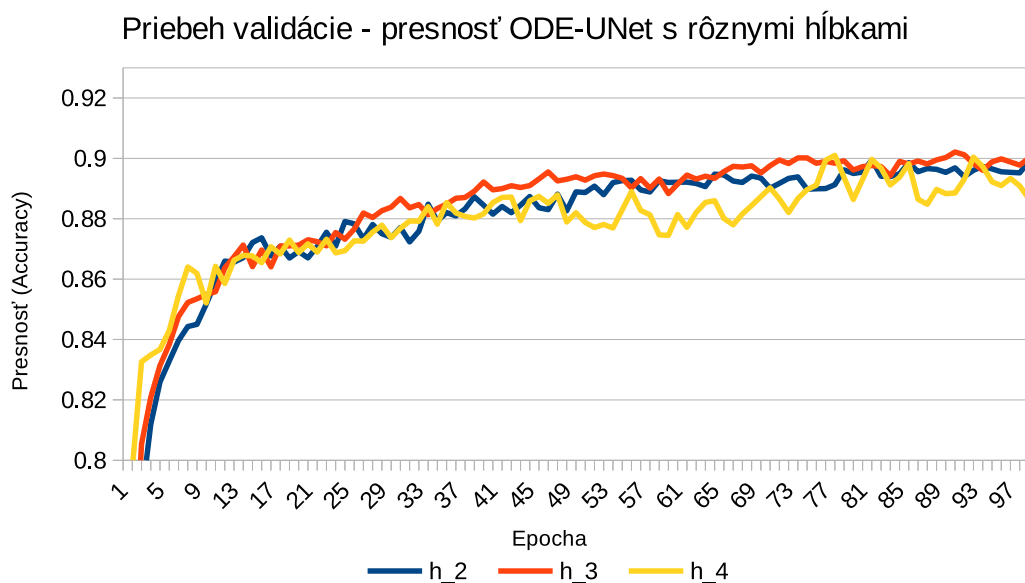
Prah [-]	Rand skóre [-]	Informačné teoretické skóre [-]
0	0,907928	0,963126
0,1	0,907884	0,963116
0,2	0,901925	0,960987
0,3	0,874843	0,955407
0,4	0,868860	0,952429
0,5	0,864646	0,951071
0,6	0,843835	0,945582
0,7	0,817346	0,935900
0,8	0,789252	0,925556

Tab. 4.1: Tabulka prahových hodnôt

v tomto bode otestovaných niekoľko hodnôt parametru hĺbka, viz. tab. 4.2. Testy sú počítané s nastaveniami modelu, ktorými bola dosiahnutá najlepšia presnosť do tohto bodu. Čas tréningovania v tejto tabulke nie je v prípade hĺbky 4 porovnateľný s ostatnými, pretože modely s hĺbkou 2 a 3 museli byť tréningované s menšou mierou učenia, inak dochádzalo k uviaznutiu v lokálnom minime, a model sa ďalej nezlepšoval.



Obr. 4.8: Pribeh tréningovej presnosti modelu ODE-UNet s rôznou hĺbkou (h_3 = hĺbka 3)



Obr. 4.9: Priebeh validačnej presnosti modelu ODE-UNet s rôznou hĺbkou (h_3 = hĺbka 3)

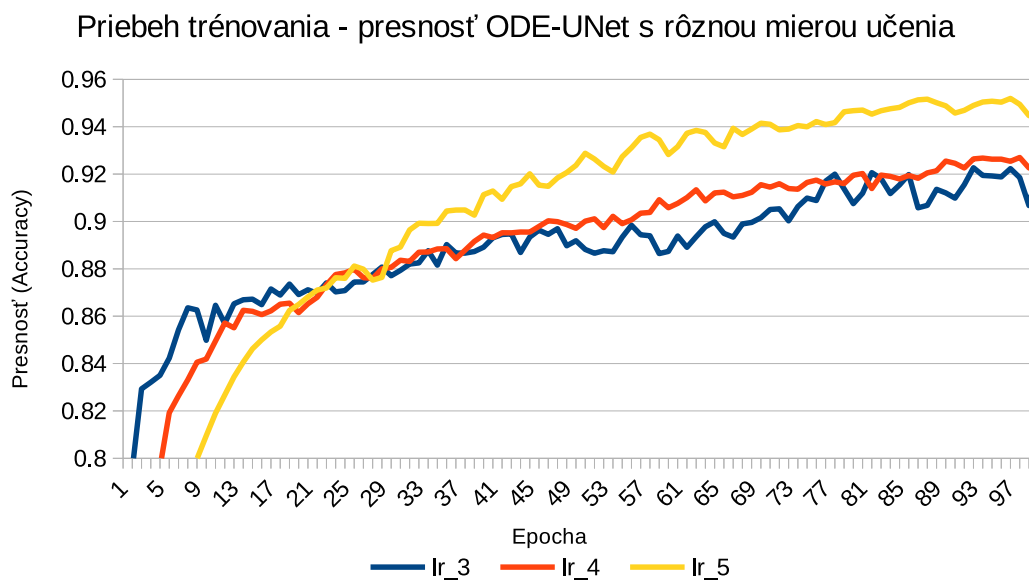
Hĺbka [-]	Čas tréno- vania [min]	Najlepšia val. pres- nosť [-]	Rand skóre [-]	Informačné teoretické skóre [-]
2	1154	0,903282	0,904764	0,968919
3	1220	0,910509	0,948470	0,971264
4	403,9	0,905035	0,950272	0,978061

Tab. 4.2: Tabuľka výsledkov pre rôzne hodnoty hĺbky

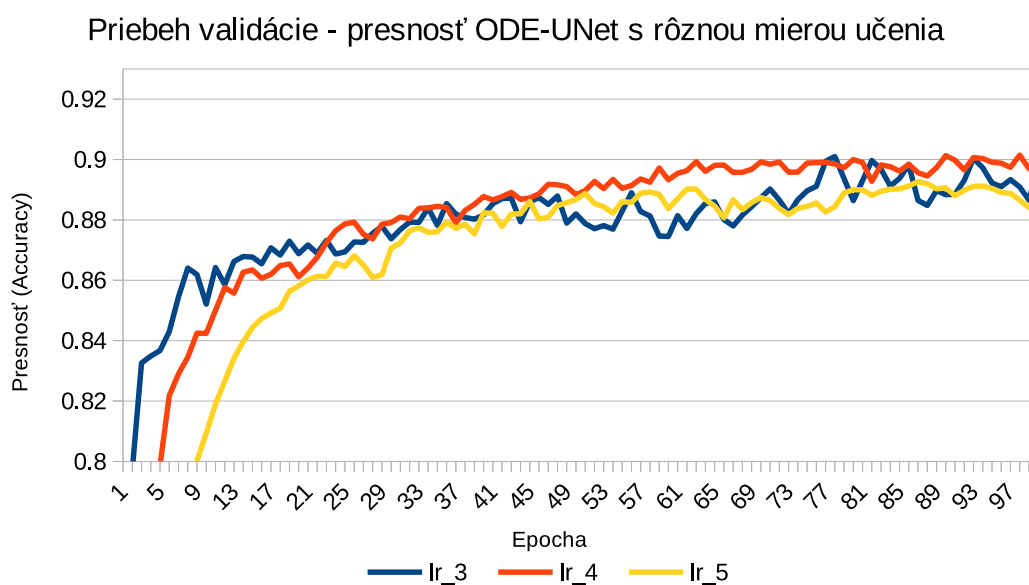
Na grafoch 4.8 a 4.9 je znázornené porovnanie priebehov trénovacej a validačnej presnosti modelov ODE-UNet s nastavením rôznej hĺbky. Najlepšiu trénovaciu aj validačnú presnosť v týchto priebehoch dosahuje model s nastavením hĺbky 3, avšak na testovacích dátach bol najpresnejší model s hĺbkou 4, viz. tab. 4.2.

5. Najhodnejšie nastavenie parametru miera učenia (learning rate) možno zistiť experimentálne. V tab. 4.3 sú porovnané výsledky pre rôzne hodnoty miery učenia. Testy sú počítané s nastaveniami modelu, ktorými bola dosiahnutá najlepšia presnosť do tohto bodu, a to konkrétne s hĺbkou 4 a bez použitia prahovania.

Na grafoch 4.10 a 4.11 je znázornené porovnanie priebehov trénovacej a validačnej presnosti modelov ODE-UNet s nastavením rôznej miery učenia. Pri trénovaní dosahoval model s mierou učenia $1e^{-5}$ výrazne lepších presností ako



Obr. 4.10: Priebeh tréovacej presnosti modelu ODE-UNet s rôznou mierou učenia (lr_3 = miera učenia $1e^{-3}$)



Obr. 4.11: Priebeh validačnej presnosti modelu ODE-UNet s rôznou mierou učenia (lr_3 = miera učenia $1e^{-3}$)

ostatné 2 modely, avšak validačná presnosť v tých istých epochách výrazne klesla. Takisto výsledná testovacia presnosť tohto modelu je nízka, možno teda predpokladať, že došlo k pretrénovaniu modelu. Najlepšiu testovaciu presnosť dosiahol model s mierou učenia $1e^{-3}$.

Miera učenia [-]	Čas tréovania [min]	Najlepšia val. presnosť [-]	Rand skóre [-]	Informačné teoretické skóre [-]
0,001	403,9	0,905035	0,950272	0,978061
0,0001	1259	0,901458	0,946784	0,975171
0,00001	1440	0,892354	0,857987	0,926803

Tab. 4.3: Tabulka výsledkov pre rôzne hodnoty miery učenia

6. Tolerancia je vstupný parameter ODE kalkulátorov, ktorý priamo ovplyvňuje presnosť výsledného modelu. Predstavuje jednu z hlavných výhod ODE sietí, pretože jeho zmenou dokážeme ovplyvniť prípustnú úroveň chyby pri výpočtoch ODE kalkulátoru, ktorý podľa tejto úrovne nastavuje počet vyhodnocovacích bodov, ktoré si možno predstaviť ako vrstvy siete. Dôležité je to, že pri tejto zmene nedochádza k zmene pamäťovej náročnosti, mení sa iba časová náročnosť. Testy sú počítané s nastaveniami modelu, ktorými bola dosiahnutá najlepšia presnosť do tohto bodu, a to hĺbka 4, miera učenia 0,001 a bez použitia prahovania.

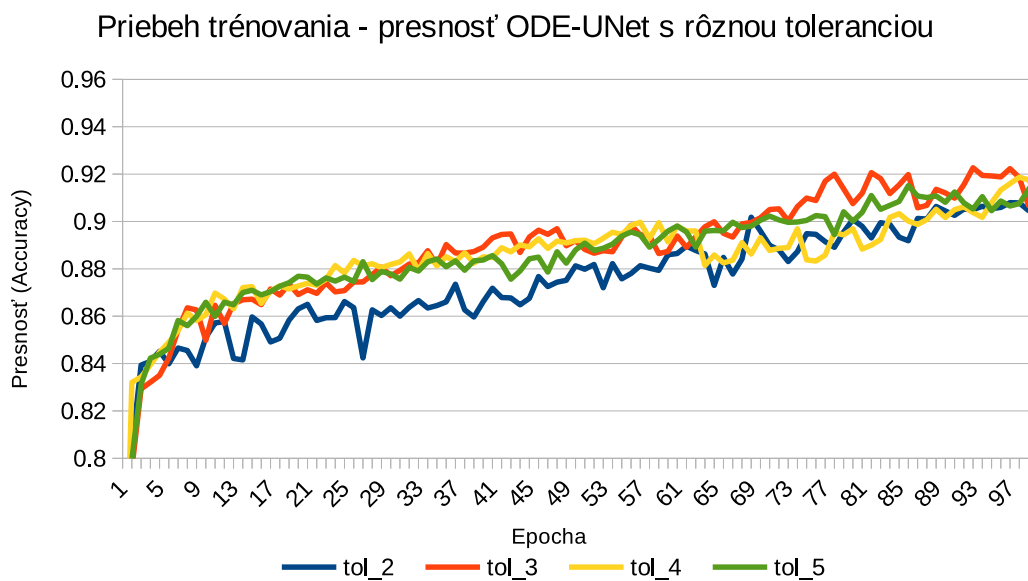
Na grafoch 4.12 a 4.13 je znázornené porovnanie priebehov trénovacej a validačnej presnosti modelov ODE-UNet s nastavením rôznej tolerancie. Priebehy majú veľmi podobný tvar, žiadny výrazný rozdiel pri zmene tolerancie nenastal. V tab. 4.4 sú porovnané výsledky pre rôzne hodnoty tolerancie. S klesajúcou hodnotou tolerancie narastala časová náročnosť, ale v dosiahnutých výsledkoch je len malý rozdiel a presnosť sa s klesajúcou toleranciou nezlepšovala, najlepšia presnosť bola dosiahnutá pri hodnote tolerancie 0,001.

Tolerancia [-]	Čas tréovania [min]	Najlepšia val. presnosť [-]	Rand skóre [-]	Informačné teoretické skóre [-]
0,01	288,6	0,902293	0,924655	0,962503
0,001	403,9	0,905035	0,950272	0,978061
0,0001	441,3	0,904556	0,953027	0,968715
0,00001	644	0,902643	0,929017	0,970479

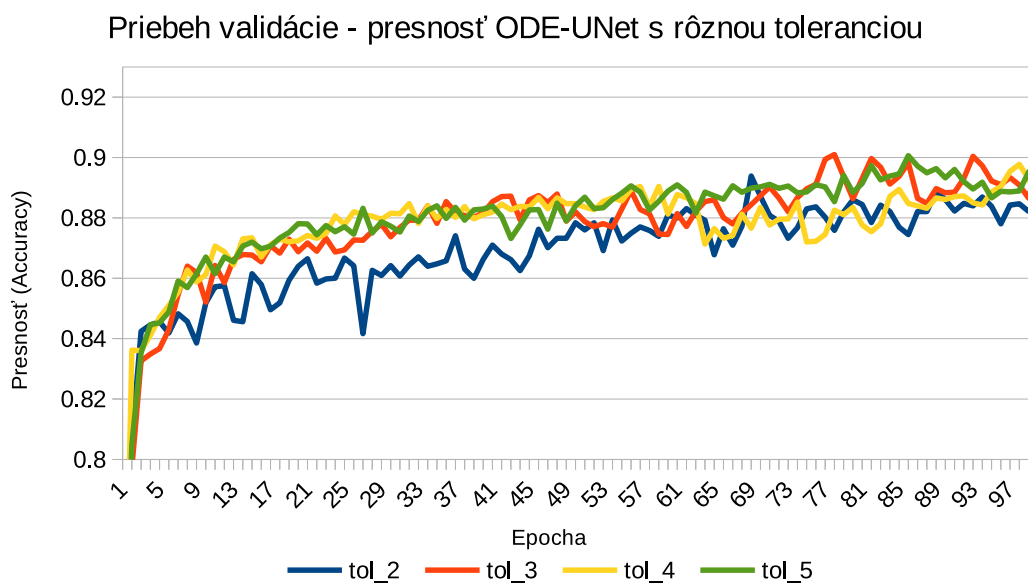
Tab. 4.4: Tabulka výsledkov pre rôzne hodnoty tolerancie

UNet-1-ODE

Pri tréovaní modelov tejto architektúry boli použité vstupné parametre, ktoré v architektúre ODE-UNet vykazovali najpresnejšie výsledky. Následne boli otestované



Obr. 4.12: Pribeh tréovacej presnosti modelu ODE-UNet s rôznou toleranciou (tol_3 = tolerancia $1e^{-3}$)



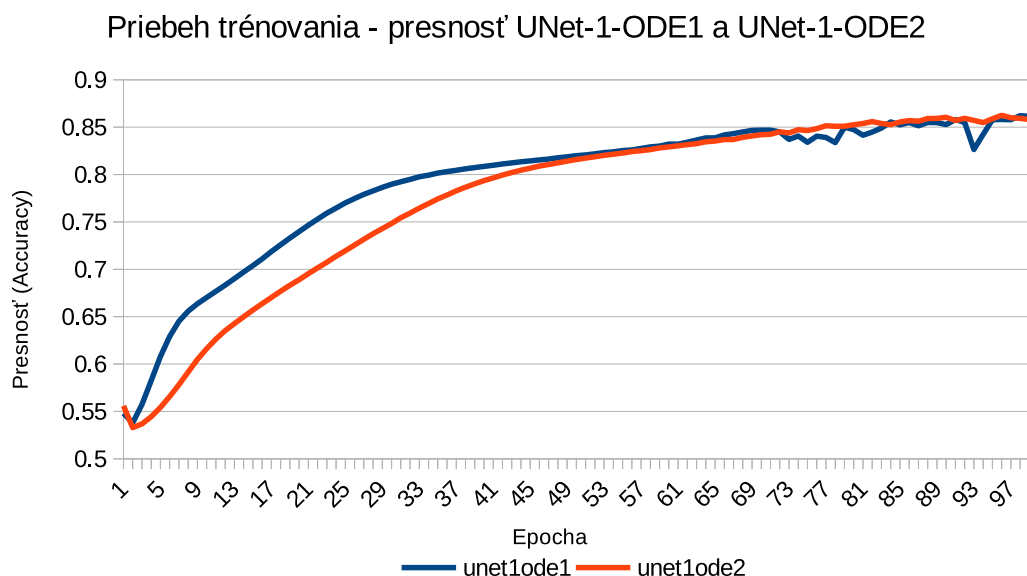
Obr. 4.13: Pribeh validačnej presnosti modelu ODE-UNet s rôznou toleranciou (tol_3 = tolerancia $1e^{-3}$)

dva modely.

1. Prvý model využíva jednoduché 2d konvolúcie. Výsledne hodnoty modelu sú nasledovné:

- čas tréovania: 456 min

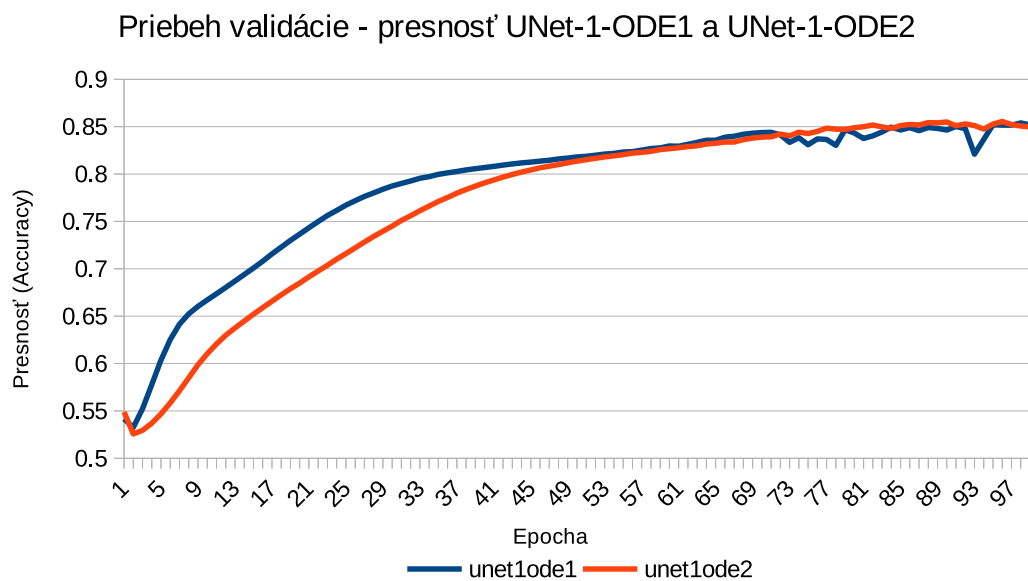
- pamäťová náročnosť: 7529 MB
 - najlepšia validačná presnosť dice: 0,905602
 - najlepšia validačná presnosť accuracy: 0,855049
 - Rand skóre: 0,781944
 - Informačné teoretické skóre: 0,906154
2. V druhom modeli boli jednoduché 2d konvolúcie nahradené rozšírenými, ktoré v rámci bloku vytvárajú medzi konvolučnými vrstvami prepojenia podľa vzoru residuálnych blokov. Výsledne hodnoty modelu sú nasledovné:
- čas tréningu: 527 min
 - pamäťová náročnosť: 8257 MB
 - najlepšia validačná presnosť dice: 0,912202
 - najlepšia validačná presnosť accuracy: 0,864408
 - Rand skóre: 0,845296
 - Informačné teoretické skóre: 0,934296



Obr. 4.14: Priebeh tréningovej presnosti modelu UNet-1-ODE1 (s jednoduchými 2d konv.) a modelu UNet-1-ODE2 (so zretazenými 2d konv.)

Na grafoch 4.14 a 4.15 je znázornené porovnanie priebehov tréningovej a validačnej presnosti modelov UNet-1-ODE s použitím jednoduchých 2d konvolúcií a s použitím rozšírených 2d konvolúcií. Na grafoch možno pozorovať plynulý, kontinuálny priebeh, pričom výsledky modelu s rozšírenými konvolúciami sú v celom priebehu bez výrazných výkyvov.

Keďže v tejto architektúre vstupuje do ODE bloku celá sieť UNet ako jedna



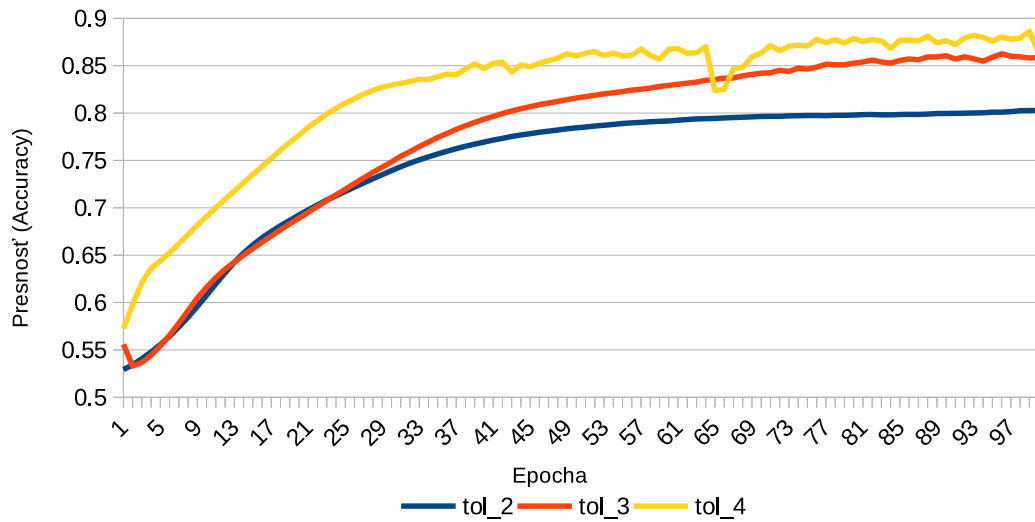
Obr. 4.15: Pribeh validačnej presnosti modelu UNet-1-ODE1 (s jednoduchými 2d konv.) a modelu UNet-1-ODE2 (so zrefaznými 2d konv.)

veľmi komplikovaná funkcia, výpočty ODE kalkulátoru sú takisto zložité. Preto sa dá predpokladať, že hodnota vstupného parametru tolerancia môže výrazne ovplyvniť výsledky tréovania. Výsledky tohto porovnania sú v tab. 4.5 a porovnanie priebehov tréovania a validácie sú znázornené v grafoch 4.16 a 4.17. Z výsledných grafov vyplýva, že pri malej hodnote tolerancie ($1e^{-2}$) sú výsledky horšie a pravdepodobne dochádza k uviaznutiu v lokálnom minime. Pri vyšších hodnotách už nedochádza k výrazným rozdielom, ale na rozdiel od porovnania výsledkov siete ODE-UNet pre rôzne tolerancie možno zhodnotiť, že presnosť sa s menšou hodnotou tolerancie zlepšovala.

Tolerancia [-]	Čas tréovania [min]	Najlepšia val. presnosť [-]	Rand skóre [-]	Informačné teoretické skóre [-]
0,01	369,2	0,821259	0,581072	0,819223
0,001	527	0,864408	0,845296	0,934296
0,0001	1062,5	0,867137	0,871101	0,950091

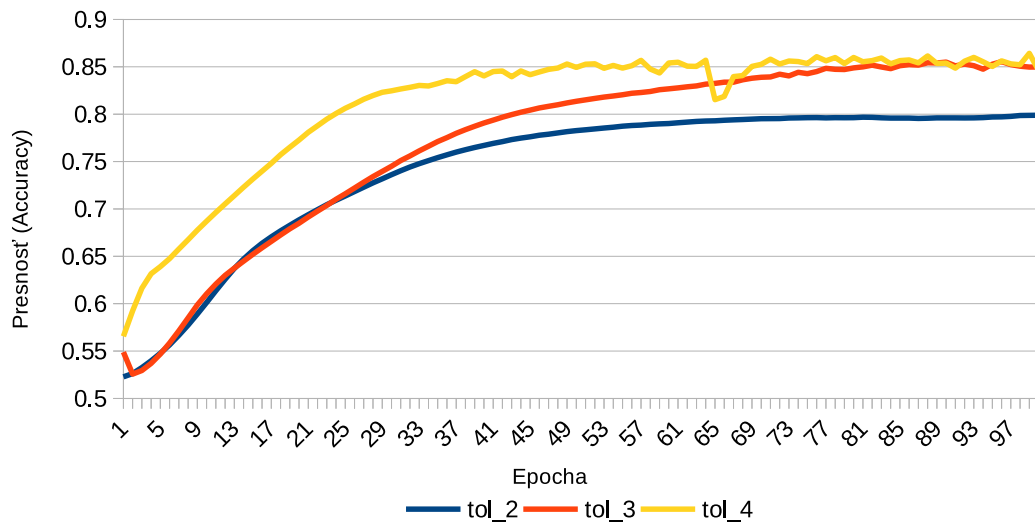
Tab. 4.5: Tabulka výsledkov pre rôzne hodnoty tolerancie

Priebeh tréovania - presnosť UNet-1-ODE pre rôzne hodnoty tolerancie



Obr. 4.16: Priebeh tréovacej presnosti modelu UNet-1-ODE s rôznymi hodnotami tolerancie (tol_3 = tolerancia $1e^{-3}$)

Priebeh validácie - presnosť UNet-1-ODE pre rôzne hodnoty tolerancie



Obr. 4.17: Priebeh validačnej presnosti modelu UNet-1-ODE s rôznymi hodnotami tolerancie (tol_3 = tolerancia $1e^{-3}$)

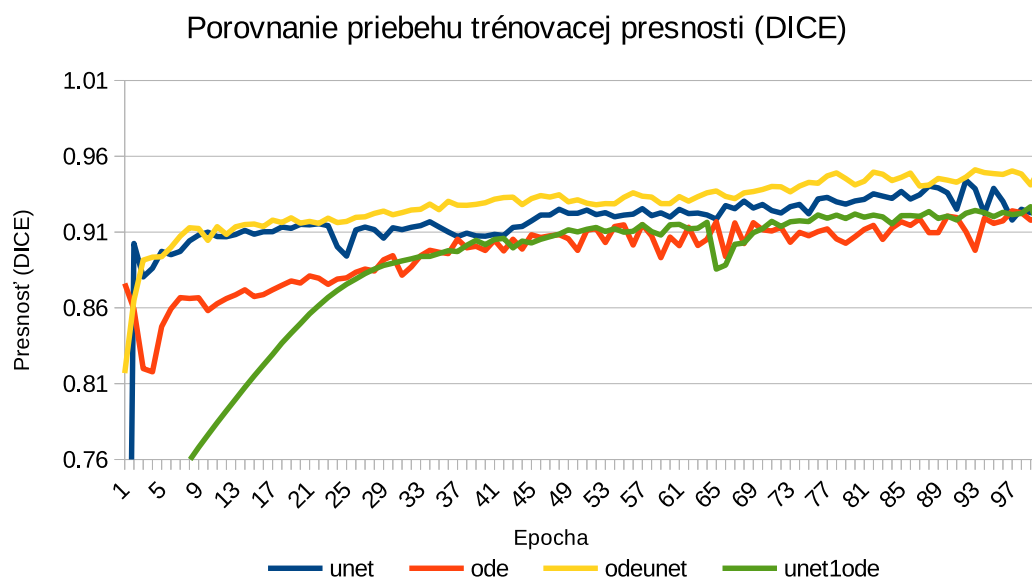
4.2 Porovnanie architektúr

V tejto časti práce sú navzájom porovnané výsledky architektúr UNet, ODE, ODE-UNet a UNet-1-ODE, ktorých detailný postup pri tréovaní je popísaný v kapitole 4. Porovnané sú modely architektúr, u ktorých boli dosiahnuté najlepšie výsledné

hodnoty, primárne na testovacích snímkoch a metrikách pre hľadanie hraničnej mapy snímkov.

Priebehy

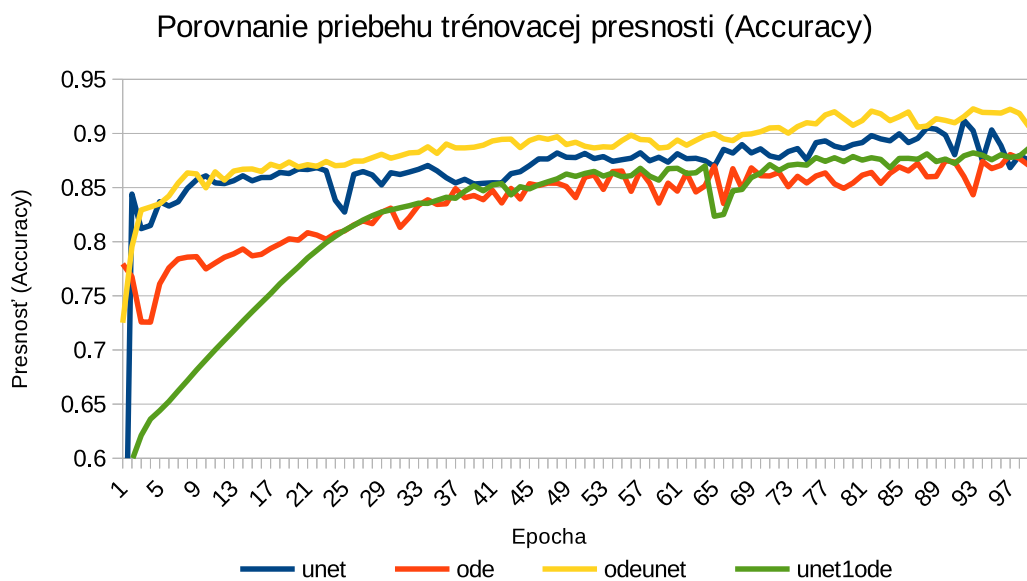
Na grafoch 4.18, 4.19, 4.20 a 4.21 sú znázornené priebehy trénovacích a validačných presností (metriek DICE a Accuracy) najpresnejších modelov jednotlivých architektur, ktoré boli v tejto práci testované. Na všetkých grafoch dosahuje najlepších výsledkov sieť ODE-UNet navrhnutá v tejto práci. Druhý najpresnejší je v týchto metrikách model architektúry UNet a modely sietí ODE a UNet-1-ODE zaostávajú výraznejším rozdielom.



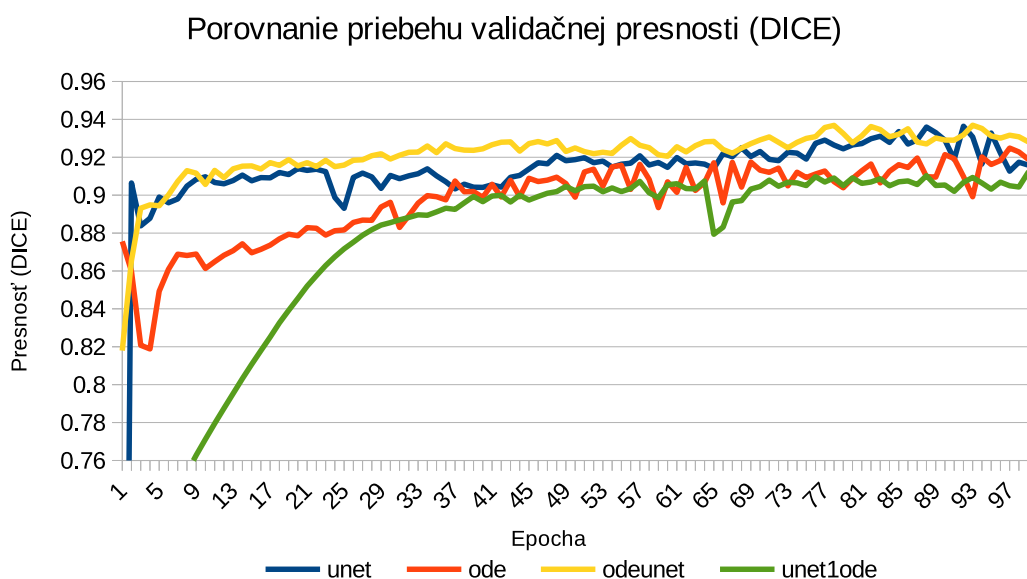
Obr. 4.18: Porovnanie priebehu trénovacej presnosti DICE sietí UNet, ODE, ODE-UNet a UNet-1-ODE

Výsledné hodnoty

Na obr. 4.22 sú zobrazené 3 validačné snímky z jednej epochy trénovania siete ODE-UNet. Jedná sa o originálny snímok ISBI dátovej sady, label snímok vytvorený tvorcami dátovej sady, ktorý určuje ako by mal vyzeráť výsledok segmentácie, a snímok, ktorý bol vygenerovaný po priechode originálneho snímku sieťou ODE-UNet v tejto epoche, a ktorý by sa mal čo najviac zhodovať s priradeným labelom. Keďže k testovacím snímkom tejto dátovej sady majú prístup iba tvorcovia a nemožno teda výsledky testovania vizuálne porovnať, tak je na tomto obr. demonštrované, čo je konkrétnym cieľom tejto úlohy s ISBI dátovou sadou. Podľa tohto potom možno následne aspoň približne vyhodnotiť testovacie snímky aj vizuálne.

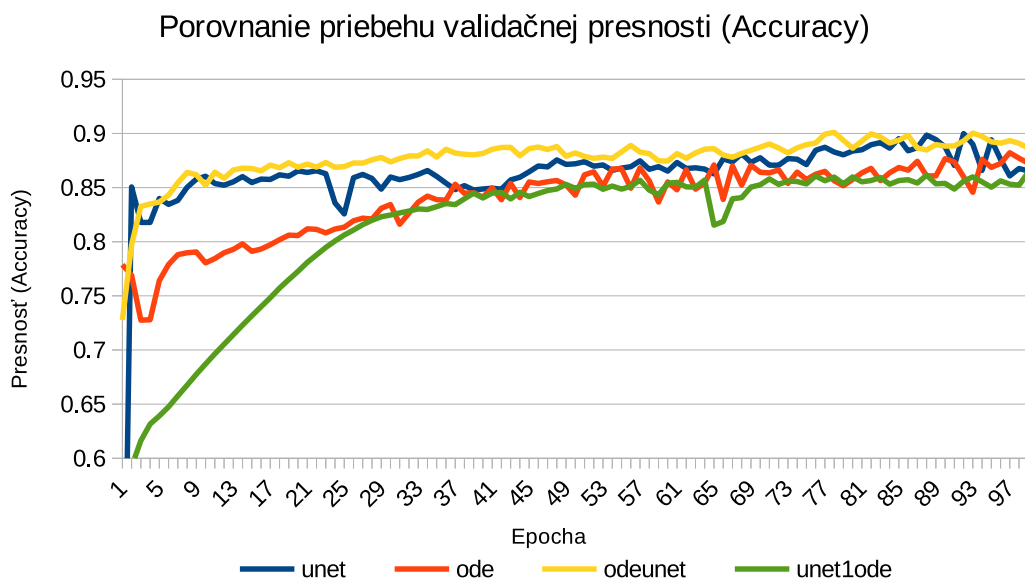


Obr. 4.19: Porovnanie priebehu trénovej presnosti ACCURACY sietí UNet, ODE, ODE-UNet a UNet-1-ODE

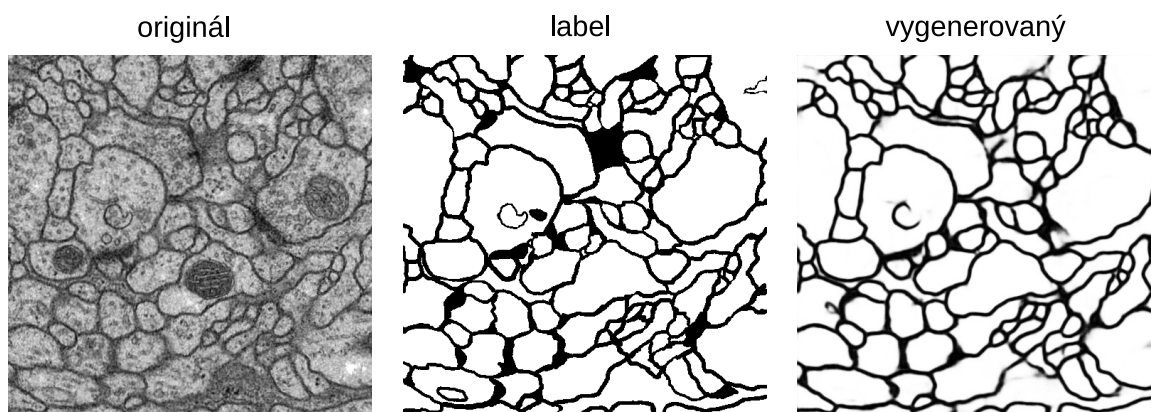


Obr. 4.20: Porovnanie priebehu validačnej presnosti DICE sietí UNet, ODE, ODE-UNet a UNet-1-ODE

Porovnaním originálneho a label snímku na obr. 4.22 sa potvrdzuje, že sa nejedná o čisto segmentačný problém. Cieľom siete je nájsť hraničnú mapu jednotlivých buniek, ale zároveň aj ignorovať vnútorné orgány buniek. Tie sú na originálnom snímku zobrazené ako väčšie tmavé kruhové oblasti, ktoré na label snímku nie sú vyznačené. Snímok, ktorý bol vygenerovaný sieťou ODE-UNet tomu takisto zod-



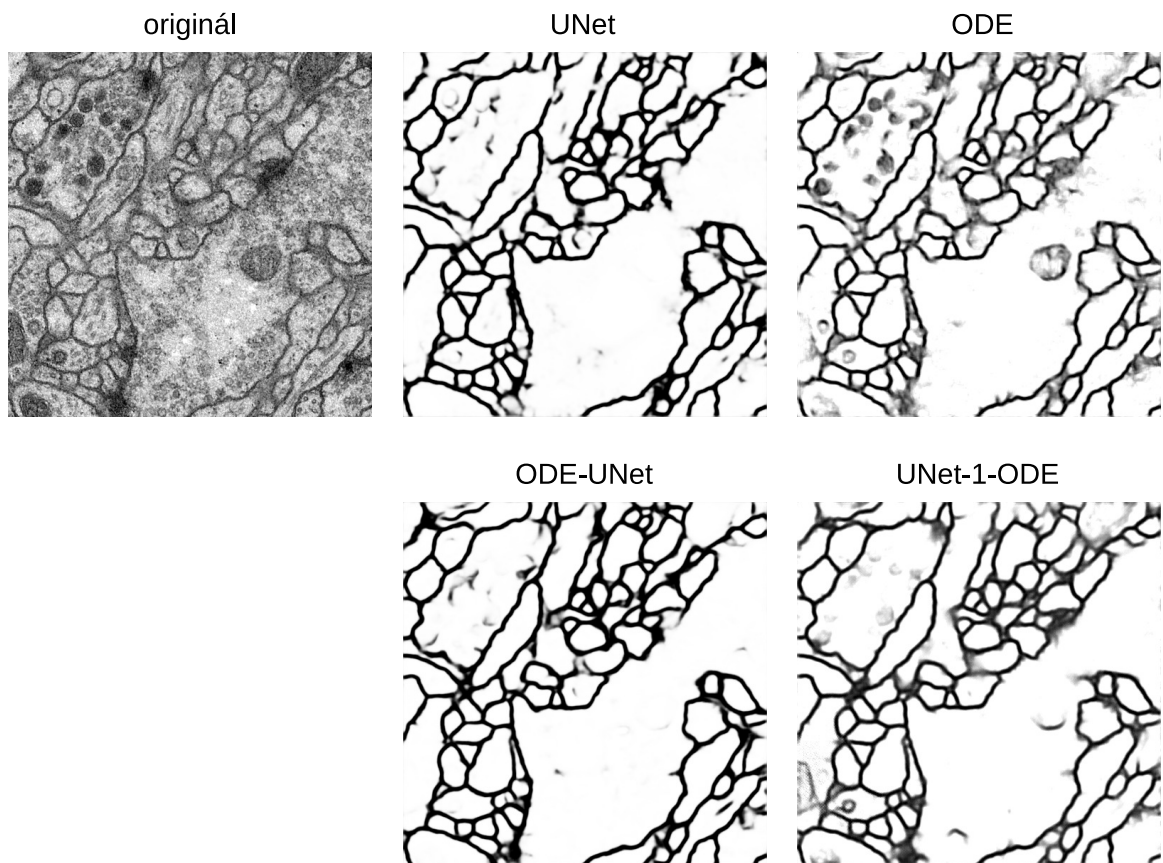
Obr. 4.21: Porovnanie priebehu validačnej presnosti ACCURACY sietí UNet, ODE, ODE-UNet a UNet-1-ODE



Obr. 4.22: Príklad validačných snímok siete ODE-UNet

povedá, vnútorné orgány sú v ňom ignorované, hranice buniek sa z veľkej časti podobajú hraniciam na label snímku, ale v niektorých miestach chýbajú oblasti, ktoré by mali byť vyznačené.

Podľa obr. 4.22 možno odhadnúť ako by mali ideálne vypadať výsledné snímky získané z testovacích originálnych snímok. K ich labelom totiž nie je prístup pre bežných riešiteľov úlohy. Na obr. 4.23 je zobrazený jeden originálny testovací snímok, a k nemu výsledky testovania najlepších modelov jednotlivých architektúr, ktoré boli v tejto práci použité. Hranice buniek sú na všetkých veľmi podobné a z veľkej časti odpovedajú originálnemu snímku. Veľký rozdiel je ale v tom, ako každý model vyhodnotil vnútorné prostredie a orgány buniek. Výstup sietí UNet a ODE-UNet je



Obr. 4.23: Generované testovacie snímky jednotlivých architektúr

velmi podobný, akurát výsledok UNetu obsahuje miestami viac nechcených tmavých pixelov v plochách, ktoré by mali byť biele. Sieť UNet-1-ODE lepšie vyhodnotila vnútorný priestor veľkej bunky v ľavom hornom rohu, ale celkovo výsledný snímok obsahuje viacero nežiadúcich zhlukov tmavých pixelov vnútri niektorých buniek. Výsledok siete ODE je najhorší, pretože neboli odfiltrované ani tie najvýraznejšie vnútorné organely.

V tab. 4.6 sú zhrnuté výsledky najlepších modelov použitých architektúr. Najlepšia presnosť všetkých štyroch použitých metrík na validačných aj testovacích dátach bola dosiahnutá modelom architektúry ODE-UNet, ktorá bola navrhnutá v tejto práci. Často používaná sieť UNet bola v tejto konkrétnej úlohe prekonaná čo sa týka presnosti, ale za cenu niekoľkonásobne väčšej časovej náročnosti. Kombinácia ODE blokov a siete UNet v oboch prípadoch znamenala takisto väčšiu pamäťovú náročnosť.

	Čas tréno- vania [min]	Presnosť DICE [-]	Presnosť ACC [-]	Rand skóre [-]	Informačné teoretické skóre [-]	Pamäťová náročnosť [MB]
UNet	24,27	0,936376	0,899853	0,902364	0,958347	5207
ODE	452,6	0,924852	0,882176	0,927354	0,967052	4895
ODE- UNet	403,9	0,938580	0,905035	0,950272	0,978061	8471
UNet-1- ODE	1062,5	0,913791	0,867137	0,871101	0,950091	8257

Tab. 4.6: Tabulka výsledkov

Záver

Diplomová práca sa zaoberá metódami segmentácie obrazu používaných v posledných rokoch a možnosťou využitia nového typu siete - ODE neurónových sietí pre tento typ úloh. V praktickej časti sú otestované a vyhodnotené štyri rôzne architektúry, ktoré boli vybrané alebo navrhnuté ako možné metódy, ktoré by umožňovali efektívne strojové učenie aj v prípade dátových množín malej veľkosti.

Cieľom práce bolo otestovať a porovnať dostupné architektúry na dátovej množine malej veľkosti a na základe týchto výsledkov navrhnúť a vytvoriť architektúru, ktorá by potenciálne mohla prekonať aktuálne používané metódy pre segmentáciu obrazu. Testovanie prebiehalo na 30 mikroskopických snímkoch mozgu octomilky.

Hlavným prínosom práce je návrh dvoch architektúr využívajúcich kombináciu UNet architektúry a ODE neurónovej siete a ich porovnanie s existujúcimi metódami segmentácie. Týmto bola otestovaná vhodnosť a presnosť siete ODE na segmentačnej úlohe v rámci použitia dátovej množiny malej veľkosti voči iným aktuálne používaným metódam. Po otestovaní jednoduchej ODE siete boli získané presnosti veľmi podobné tým, ktoré dosiahol model známej siete UNet. Takisto boli potvrdené výhody, ktoré boli prezentované pri publikácii ODE sietí v decembri 2018, a to napríklad konštantná pamäťová náročnosť pri rôznych hodnotách tolerancie, čo možno považovať v klasickom ponímaní neurónových sietí ako možnosť použitia neobmedzeného množstva vrstiev. Parametrom tolerancia možno teda meniť presnosť siete na úkor časovej náročnosti a naopak. Prísľub lepšej generalizácie na malých dátových množinách bolo po otestovaní jednoduchej ODE siete tiež možné považovať za možný, pretože na testovacích dátach dosiahla lepších výsledkov ako sieť UNet. Tieto výhody spolu so sľubnými výsledkami viedli k návrhu architektúr ODE-UNet a UNet-1-ODE, z ktorých bola sieť ODE-UNet tou, ktorá prekonalala v presnosti všetky ostatné testované architektúry. Na ISBI dátovej sade dosiahla hodnôt Rand: 0,950272 a Info: 0,978061. Pre porovnanie, použitá sieť UNet dosiahla hodnôt Rand: 0,902364 a Info: 0,958347. Veľkou prednosťou siete UNet ale ostáva časová náročnosť, keďže tá bola viac ako o jeden rád nižšia. Navrhnutá sieť ODE-UNet teda prekonalala použitú sieť UNet, avšak v rámci oficiálnych výsledkov ISBI dátovej sady stále zaostáva za niekoľkými desiatkami presnejších modelov. Nemožno teda tvrdiť, že by boli použitím siete ODE prekonané aktuálne používané metódy pre segmentáciu obrazu, ale jej výhody boli potvrdené a stále existuje veľký priestor pre rozširovanie a zlepšovanie tejto metódy.

Výstupom práce sú výsledné hodnoty použitých sietí zobrazené v grafoch a tabuľkách spolu s popisom a vyhodnotením každého testu. Tieto výsledky jednotlivých architektúr sú následne porovnané vo výsledných grafoch a tabuľke, v ktorej sú vyznačené najlepšie dosiahnuté hodnoty v rámci všetkých experimentov.

Literatúra

- [1] RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. *U-net: Convolutional networks for biomedical image segmentation*. In: International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015. p. 234-241.
- [2] CHEN, Tian Qi, et al. *Neural ordinary differential equations*. In: Advances in neural information processing systems. 2018. p. 6571-6583.
- [3] NIELSEN, Michael A. *Neural networks and deep learning*. San Francisco, CA, USA.: Determination press, 2015.
- [4] SCHMIDHUBER, Jürgen. *Deep learning in neural networks: An overview*. Neural networks, 2015, 61: 85-117.
- [5] BOTTOU, Léon. *Stochastic gradient descent tricks*. In: Neural networks: Tricks of the trade. Springer, Berlin, Heidelberg, 2012. p. 421-436.
- [6] CHAUVIN, Yves; RUMELHART, David E. *Backpropagation: theory, architectures, and applications*. Psychology Press, 2013.
- [7] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. *Imagenet classification with deep convolutional neural networks*. In: Advances in neural information processing systems. 2012. p. 1097-1105.
- [8] YANG, Jihoon. *Deep learning*. 2016.
- [9] ARORA, Raman, et al. *Understanding deep neural networks with rectified linear units*. arXiv preprint arXiv:1611.01491, 2016.
- [10] ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. *Understanding of a convolutional neural network*. In: 2017 International Conference on Engineering and Technology (ICET). IEEE, 2017. p. 1-6.
- [11] BUTCHER, John Charles. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.
- [12] HE, Kaiming, et al. *Deep residual learning for image recognition*. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778.
- [13] HONCHAR, Alexandr. *Neural ODEs: breakdown of another deep learning breakthrough*. Towards Data Science [online]. 2019 [cit. 2019-11-26]. Dostupné

- z URL:
 <<https://towardsdatascience.com/neural-odes-breakdown-of-another-deep-learning-breakthrough-3e78c7213795>>.
- [14] HOLLÄNDER, Branislav. *Paper Summary: Neural Ordinary Differential Equations. Towards Data Science [online]*. 2018 [cit. 2019-11-26]. Dostupné z URL:
 <<https://towardsdatascience.com/paper-summary-neural-ordinary-differential-equations-37c4e52df128>>.
- [15] Mihajlovic, Ilija *Everything You Ever Wanted To Know About Computer Vision [online]*. 2019. Dostupné z URL:
 <<https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e>>.
- [16] Kaeli, David et al. *Image Classification [online]*. 2015. Dostupné z URL:
 <<https://www.sciencedirect.com/topics/computer-science/image-classification>>.
- [17] Ganesh, Prakhar *Object Detection : Simplified [online]*. 2019. Dostupné z URL:
 <<https://towardsdatascience.com/object-detection-simplified-e07aa3830954>>.
- [18] LE, James. *How to do Semantic Segmentation using Deep learning. Nanonets.com [online]*. 2018 [cit. 2019-11-29]. Dostupné z URL:
 <<https://nanonets.com/blog/how-to-do-semantic-segmentation-using-deep-learning/>>.
- [19] DAI, Jifeng; HE, Kaiming; SUN, Jian. *Instance-aware semantic segmentation via multi-task network cascades*. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. p. 3150-3158.
- [20] MINAEE, Shervin, et al. *Image Segmentation Using Deep Learning: A Survey*. arXiv preprint arXiv:2001.05566, 2020.
- [21] YUHENG, Song; HAO, Yan. *Image segmentation algorithms overview*. arXiv preprint arXiv:1707.02051, 2017.
- [22] R, Muthukrishnan. (2011). *Edge Detection Techniques For Image Segmentation*. International journal of computer science and information technology. 3. 259-267. 10.5121/ijcsit.

- [23] HOREČNÝ, Peter. Výpočetní úlohy pro předmět Paralelní zpracování dat [online]. Brno, 2017 [cit. 2020-03-21]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/107243>. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Jan Mašek.
- [24] LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 3431-3440.
- [25] MINAEE, Shervin, et al. Image Segmentation Using Deep Learning: A Survey. arXiv preprint arXiv:2001.05566, 2020.
- [26] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1520–1528
- [27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.
- [28] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in 2016 Fourth International Conference on 3D Vision (3DV). IEEE, 2016, pp. 565–571.
- [29] JÉGOU, Simon, et al. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017. p. 11-19.
- [30] HUANG, Gao, et al. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 4700-4708.
- [31] ZHAO, Hengshuang, et al. Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 2881-2890.
- [32] LIN, Tsung-Yi, et al. Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 2117-2125.
- [33] J. He, Z. Deng, and Y. Qiao, "Dynamic multi-scale filters for semantic segmentation," in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3562–3572.

- [34] HE, Junjun, et al. Adaptive pyramid context network for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019. p. 7519-7528.
- [35] LIN, Di, et al. Multi-scale context intertwining for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). 2018. p. 603-619.
- [36] GIRSHICK, Ross. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2015. p. 1440-1448.
- [37] REN, Shaoqing, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. 2015. p. 91-99.
- [38] HE, Kaiming, et al. Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2017. p. 2961-2969.
- [39] SHARMA, Pulkit. *Computer Vision Tutorial: Implementing Mask R-CNN for Image Segmentation* Analytics Vidhya [online]. 2019. Dostupné z URL: <https://www.analyticsvidhya.com/blog/2019/07/computer-vision-implementing-mask-r-cnn-image-segmentation/>.
- [40] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutionalnets and fully connected crfs,”arXiv preprint arXiv:1412.7062, 2014.
- [41] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convo-lutional nets, atrous convolution, and fully connected crfs,”IEEEtransactions on pattern analysis and machine intelligence, vol. 40, no. 4,pp. 834–848, 2017.
- [42] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinkingatrous convolution for semantic image segmentation,”arXivpreprint arXiv:1706.05587, 2017
- [43] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam,“Encoder-decoder with atrous separable convolution for semanticimage segmentation,” inProceedings of the European conference oncomputer vision (ECCV), 2018, pp. 801–818.
- [44] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio,M. Matteucci, and A. Courville, “Reseg: A recurrent neuralnetwork-based model for semantic segmentation,” inProceedingsof the IEEE Conference on Computer Vision and Pattern RecognitionWorkshops, 2016, pp. 41–48.

- [45] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio, “Renet: A recurrent neural network based alternative to convolutional networks,” arXiv preprint arXiv:1505.00393, 2015
- [46] Y. Xiang and D. Fox, “Da-rnn: Semantic mapping with data associated recurrent neural networks,” arXiv preprint arXiv:1703.03098, 2017.
- [47] MAHESWARI, Jyoti Prakash. *Breaking the curse of small datasets in Machine Learning: Part 1*. Towards Data Science [online]. 2018 [cit. 2019-11-29]. Dostupné z URL:
<<https://towardsdatascience.com/breaking-the-curse-of-small-datasets-in-machine-learning-part-1-36f28b0c044d/>>.
- [48] WALIA, Anish Singh. *Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent*. Towards Data Science [online]. 2017 [cit. 2019-11-30]. Dostupné z URL:
<<https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f/>>.
- [49] MAHESWARI, Jyoti Prakash. *Breaking the curse of small data sets in Machine Learning: Part 2*. Towards Data Science [online]. 2019 [cit. 2019-11-30]. Dostupné z URL:
<<https://towardsdatascience.com/breaking-the-curse-of-small-data-sets-in-machine-learning-part-2-894aa45277f4/>>.
- [50] ARGANDA-CARRERAS, Ignacio, Srinivas C. TURAGA, Daniel R. BERGER, et al. *Crowdsourcing the creation of image segmentation algorithms for connectomics*. Frontiers in Neuroanatomy [online]. 2015, 9 [cit. 2019-11-30]. DOI: 10.3389/fnana.2015.00142. ISSN 1662-5129. Dostupné z URL:
<<http://journal.frontiersin.org/Article/10.3389/fnana.2015.00142/full/>>.
- [51] LI, Xiaomeng, et al. *H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes*. IEEE transactions on medical imaging, 2018, 37.12: 2663-2674.
- [52] TAO, Qingyi, et al. *3D Unet-based Kidney and Kidney Tumor Segmentation with Attentive Feature Learning*. 2019.
- [53] WANG, Chuanxia, et al. *BiSC-UNet: A fine segmentation framework for kidney and renal tumor*.

- [54] WANG, Liansheng; CHEN, Rongzhen; HU, Yanxing. *IDDF2018-ABS-0261 Polyp detection using an unet based model*. 2018.
- [55] CHU, Zhengquan, et al. *Sea-Land Segmentation With Res-UNet And Fully Connected CRF*. In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019. p. 3840-3843.
- [56] DAVIS, Mark. *How Does Google Earth Work?* Livescience.com [online]. 2019 [cit. 2019-12-08]. Dostupné z URL: [<https://www.livescience.com/65504-google-earth.html/>](https://www.livescience.com/65504-google-earth.html/).
- [57] ZHANG, Tianjun, et al. *ANODEV2: A Coupled Neural ODE Framework*. In: *Advances in Neural Information Processing Systems*. 2019. p. 5152-5162.
- [58] KRIZHEVSKY, Alex, et al. *Learning multiple layers of features from tiny images*. Technical report, University of Toronto, 2009.
- [59] DUPONT, Emilien; DOUCET, Arnaud; TEH, Yee Whye. *Augmented neural odes*. arXiv preprint arXiv:1904.01681, 2019.
- [60] LIU, Xuanqing, et al. *Neural SDE: Stabilizing Neural ODE Networks with Stochastic Noise*. arXiv preprint arXiv:1906.02355, 2019.
- [61] PORTWOOD, Gavin D., et al. *Turbulence forecasting via Neural ODE*. arXiv preprint arXiv:1911.05180, 2019.
- [62] Arbeláez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). *Contour detection and hierarchical image segmentation*. *IEEE Trans. Patt. Anal. Mach. Intell.* 33, 898–916. doi: 10.1109/TPAMI.2010.161
- [63] Meilă, M. (2005). “*Comparing clusterings: an axiomatic view*,” in *Proceedings of the 22nd International Conference on Machine Learning, ICML '05* (New York, NY: ACM), 577–584.
- [64] Unnikrishnan, R., Pantofaru, C., and Hebert, M. (2007). *Toward objective evaluation of image segmentation algorithms*. *IEEE Trans. Patt. Anal. Mach. Intell.* 29, 929–944. doi: 10.1109/TPAMI.2007.1046
- [65] DEL TORO, Oscar Alfonso Jiménez, et al. *VISCERAL–VISual Concept Extraction challenge in RAdioLogY: ISBI 2014 challenge organization*. *Proceedings of the VISCERAL Challenge at ISBI, 2014*, 1194: 6-15.
- [66] YAN, Hanshu, et al. *On Robustness of Neural Ordinary Differential Equations*. arXiv preprint arXiv:1910.05513, 2019.

- [67] Anindya Sarkar, Anirudh Sunder Raj, Raghu Sesha Iyengar. *Improving Robustness of time series classifier with Neural ODE guided gradient based data augmentation* arXiv preprint arXiv:1910.06813 , 2019.
- [68] ZHANG, Han, et al. *Approximation Capabilities of Neural Ordinary Differential Equations*. arXiv preprint arXiv:1907.12998, 2019.
- [69] XIE, Xiang; PARLIKAD, Ajith Kumar; PURI, Ramprakash Srinivasan. *A Neural Ordinary Differential Equations Based Approach for Demand Forecasting within Power Grid Digital Twins*. In: 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, 2019. p. 1-6.
- [70] PAOLETTI, Mercedes E., et al. *Neural Ordinary Differential Equations for Hyperspectral Image Classification*. IEEE Transactions on Geoscience and Remote Sensing, 2019.

Zoznam symbolov, veličín a skratiek

ODE	Ordinary Differential Equations - obyčajné diferenciálne rovnice
ReLU	Rectified Linear Unit
tanh	tangens hyperbolický
H-Dense	Hybrid Densely Connected UNet - hybridný husto prepojený UNet
3D	troj-dimenzionálny
CT	Computed Tomography - počítačová tomografia
SC-UNet	Skip Connection Unet - UNet s prepojeniami
WCE	Wireless capsule endoscopy - bezdrátová kapsulová endoskopia
ResNet	Residual Neural Network - residuálna neurónová sieť
NIPS	Neural Information Processing Systems - konferencia o systémoch spracovania neurónových informácií
ILSVRC	ImageNet Large Scale Visual Recognition Competition - súťaž v spracovaní obrazu
CIFAR-10	dátová sada pre segmentáciu obrazu
SDE	Neural Stochastic Differential Equation - Stochastické neurónové siete využívajúce diferenciálne rovnice
ADAM	Adaptive Moment Estimation - optimalizačná funkcia
ISBI	International Symposium on Biomedical Imaging - medzinárodné sympóziom o zobrazovaní bio-medicínskych snímok
EM	Electron Microscope - elektrónový mikroskop
ssTEM	Scanning Transmission Electron Microscope - typ elektrónového mikroskopu
MNIST	Modified National Institute of Standards and Technology - inštitút štandardov a technológií
norm	normalizačná vrstva
FCN	Fully Convolution Network - plne konvolučná vrstva
conv2d	vrstva 2d konvolúcia
PSPNet	Pyramid Scene Parsing Network - pyramídový model
FPN	Feature Pyramid Network - pyramídový model
DM-Net	Dynamic Multi-scale Filters Network - viacúrovňový model
APC-Net	Adaptive Pyramid Context Network - adaptívny pyramídový model
MSCI	Multi-scale context intertwining - viacúrovňový model
R-CNN	R-Convolutional Neural Network - R-konvolučná neurónová sieť
RPN	Region Proposal Network - sieť návrhu oblasti
RoI	Region of Interest - oblasť záujmu
IoU	Intersection over Union - prienik cez zjednotenie
ASPP	Atrous Spatial Pyramid Pooling - vrstva DeepLabv2 architektúry

DA-RNN	Data Associated Recurrent Neural Network - rozšírená rekurentná neurónová sieť
ConcatConv2d	Concatenated 2d Convolutional layer - rozšírená konvolučná vrstva
tol	tolerancia
lr	learning rate - miera učenia
Rand	Rand skóre
Info	Informačné teoretické skóre