

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020

Zuzana Koščová



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

**KONVOLUČNÍ NEURONOVÉ SÍŤ PRO DETEKCI
OBJEKTŮ V MEDICÍNSKÝCH OBRAZECH**

DEEP-LEARNING-BASED PATTERN DETECTION IN MEDICAL IMAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Zuzana Koščová

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Roman Jakubíček

BRNO 2020



Bakalářská práce

bakalářský studijní obor **Biomedicínská technika a bioinformatika**

Ústav biomedicínského inženýrství

Studentka: Zuzana Koščová

ID: 203197

Ročník: 3

Akademický rok: 2019/20

NÁZEV TÉMATU:

Konvoluční neuronové sítě pro detekci objektů v medicínských obrazech

POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se s problematikou konvolučních neuronových sítí se zaměřením na detekční architektury. 2) Vyhledejte informace o dvou nejnovějších algoritmech pro detekci využívající hlubokého učení (YOLO a R-CNN) a vypracujte rešerši a teoretický rozbor těchto metod. 3) Nastudujte princip učení těchto sítí a tvorbu učební databáze včetně její augmentace. Vytvořte libovolnou učební a trénovací databázi. 4) Natrénujte alespoň jeden z dostupných algoritmů ve vhodně zvoleném programovém prostředí. 5) Implementujte obě metody a trénujte na dostupných databázích. Proveďte srovnání těchto metod. 6) Aplikujte metody na medicínská data a diskutujte výsledky a možnosti využití v diagnostice.

DOPORUČENÁ LITERATURA:

[1] 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.

[2] REDMON, J, S DIVVALA, R GIRSHICK a A FARHADI. You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [online]. Los Alamitos, CA, USA: IEEE Computer Society, 2016, s. 779–788. ISSN 1063-6919. Dostupné z: doi:10.1109/CVPR.2016.91

Termín zadání: 3.2.2020

Termín odevzdání: 5.6.2020

Vedoucí práce: Ing. Roman Jakubiček

prof. Ing. Ivo Provazník, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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

Táto bakalárska práca sa zaoberá konvolučnými neurónovými sieťami (CNN) pre detekciu objektov v medicínskych obrazoch. Pre lepšie pochopenie problematiky sú najprv popísané umelé neurónové siete a CNN všeobecne, neskôr sú kapitoly zamerané na konkrétne metódy detekcie, ktoré využívajú CNN. V rámci práce bola vytvorená sada abdominálnych snímok z výpočetnej tomografie a z magnetickej rezonančnej tomografie na ktorých boli trénované a testované „Faster R-CNN“ a „You Only Look Once“ algoritmy pre detekciu pečene. Implementácia oboch modelov prebehla v programovacom jazyku Python s využitím knižnice Pytorch. Na záver sú vyvedené a diskutované výsledky detekcie a možné využitie v lekárstve.

KLÚČOVÉ SLOVÁ

konvolučné neurónové siete, detekcia objektov, CT, MRI, Faster R-CNN, YOLO, ohraňujúci rámček, pečeň

ABSTRACT

This Bachelor thesis deals with Deep-learning-based pattern detection in medical images. For better understanding of a subject artificial neural network and convolutional neural network (CNN) are described at first. Next chapter is focused on specific detection methods which use CNN. Within a thesis a dataset of abdominal CT a MRI scans was created. Faster R-CNN and YOLO algorithms were trained and tested on acquired scans for liver detection. Implementation of chosen methods took place in Python programming language using the Pytorch library. Finally, detection results and possible use in medicine are discussed.

KEYWORDS

Deep learning, pattern detection, CT, MRI, Faster R-CNN, YOLO, bounding box, liver

KOŠČOVÁ, Zuzana. *Konvoluční neuronové sítě pro detekci objektů v medicínských obrazech*. Brno, 2020, 66 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství. Vedúci práce: Ing. Roman Jakubíček

VYHLÁSENIE

Vyhlasujem, že svoju bakalársku prácu na tému „Konvoluční neuronové sítě pro detekci objektů v medicínských obrazech“ som vypracovala samostatne pod vedením vedúceho bakalárskej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autora uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušila autorské práva tretích osôb, najmä som nezasiahla nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomá 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 autorky

POĎAKOVANIE

Rada by som poďakovala vedúcemu práce pánovi Ing. Romanovi Jakubíčkovi za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Obsah

Úvod	11
1 Neurónové siete	12
1.1 Biologický neurón	12
1.2 Matematický model neurónu (perceptrón)	12
1.3 Viacvrstvová neurónová sieť	14
1.4 Topológia neurónovej siete	14
1.5 Učenie neurónovej siete	15
2 Konvolučné neurónové siete	17
2.1 Architektúra CNN	17
2.2 Učenie konvolučnej neurónovej siete	20
3 Detekcia objektov	21
3.1 Konvolučná neurónová sieť založená na návrhu regiónov	21
3.2 „Fast“ R-CNN	22
3.3 „Faster“ R-CNN	24
3.4 „You Only Look Once“	27
4 Experimentálne dáta	30
5 Implementácia algoritmov	33
5.1 Realizácia „Faster“ R-CNN	33
5.2 Realizácia YOLO v3	40
6 Hodnotenie a výsledky detekcie objektov	44
6.1 Metriky hodnotenia detekcie	44
6.2 Porovnanie dosiahnutých výsledkov	46
6.3 Diskusia dosiahnutých výsledkov	51
6.4 Využitie v medicíne	56
Záver	57
Literatúra	58
Zoznam príloh	62
A Tabuľka presností	63
B Ukážky detekcií z modelu „Faster“ R-CNN	64

C Ukázky detekcí z modelu YOLO v3	65
D Obsah příloh v informačnom systéme VUT	66

Zoznam obrázkov

1.1	Umelý neurón	12
1.2	Viacvrstvová neurónová sieť	14
2.1	Konvolučná neurónová sieť	18
2.2	Konvolučné vrstvy	19
2.3	ReLU	19
2.4	„MaXPooling“	20
3.1	R-CNN	22
3.2	„Fast“ R-CNN	23
3.3	„Faster“ R-CNN	24
3.4	„Region Proposal Network (RPN)“	25
3.5	YOLO	29
4.1	Ukážka CT a MRI rezov s anotovanými rámčekmi	31
4.2	Augmentácia	32
5.1	Model „Faster“ R-CNN	34
5.2	Architektúra siete MobileNet v2	35
5.3	Znázornenie „anchors“ rámčekov pre „Faster“ R-CNN	36
5.4	Ukážka priebehu chybových funkcií „Faster“ R-CNN	39
5.5	DarkNet-53	40
5.6	Architektúra YOLO v3	41
5.7	Ukážka anchors rámčekov pre YOLO	42
5.8	Priebeh YOLO chybovej funkcie	43
6.1	IoU	44
6.2	Zaradenie detekcií do skupín	46
6.3	Aplikácia funkcie „Non Max Suppression“ na obrázkoch	46
6.4	Výber jedného rámčeku na základe „confidence score“	47
6.5	Krivka závislosti presnosti detekcie na prahu IoU	48
6.6	Koláčové grafy ukazujúce percentuálne zastúpenie typov detekcií	48
6.7	Grafy závislosti presností na prahu IoU pre CT a MRI modalitu a oba algoritmy	49
6.8	Grafy závislosti presností na prahu IoU pre MRI sekvencie a oba algoritmy	49
6.9	Detekcie na CT rezoch s vysokou presnosťou	51
6.10	Príklady detekcií s nízkym IoU medzi predvídaným a pravdivým rámčekom na CT rezoch	52
6.11	Detekcie s vysokou presnosťou na MRI rezoch v T1 DUAL vstupnej fáze	53
6.12	Detekcie s nízkym IoU na MRI rezoch v T1 DUAL vstupnej fáze	54

6.13	Príklady detekcií na MRI rezoch v T2 váhovanej SPIR fáze	55
6.14	Príklady neúspešných detekcií na MRI rezoch v T2 váhovanej SPIR fáze	55
B.1	Ukážky výsledných detekcií z „Faster“ R-CNN modelu	64
C.1	Ukážky výsledných detekcií z YOLO v3 modelu	65

Zoznam tabuliek

6.1	Priemerné presnosti detekcií z validácií náhodného výberu pri IoU 50 %	47
A.1	Tabuľka presností pre opakované validácie náhodného výberu	63

Úvod

Konvolučné neurónové siete sú v dnešnej dobe jedným z najväčších trendov v oblasti spracovania obrazov, pretože využívajú špeciálnu architektúru dobre adaptovanú pre klasifikáciu a taktiež detekciu. Je to mimoriadny typ neurónových sietí, ktoré dosiahli významný pokrok v oblastiach počítačového videnia. Lekársky výskum v rovine konvolučných neurónových sietí nie je výnimkou. Zobrazovacie systémy v lekárstve poskytujú snímky, ktoré obsahujú dôležité vizuálne informácie o ľudskom tele, avšak ich hodnotenie voľným okom býva často časovo náročné a výsledok je subjektívny. Práve správne využitie konvolučných neurónových sietí k detekcii objektov by mohlo viesť k lepším výsledkom či už v diagnostike alebo pri plánovaní intervenčných výkonov.

Presná detekcia, ktorá v niektorých prípadoch slúži ako predspracovanie pre segmentačné úlohy má dôležitý vplyv na predbežné hodnotenie pečene pre transplantáčny výkon alebo pri operáciách brušnej aorty, kedy je nutné analyzovať a určiť presnú pozíciu orgánov brušnej dutiny a ciev do nich vstupujúcich. Toto poháňa prebiehajúci výskum, aby sa dosiahli lepšie výsledky a prekonalí mnohé výzvy, ktoré vyplývajú z vysoko flexibilných anatomických vlastností brucha a obmedzení modality odrážajúcich vlastnosti obrazu [1].

Prvá kapitola bakalárskej práce sa zaoberá problematikou neurónových sietí, ich stavbou, topológiami a učením. V druhej kapitole boli predstavené konvolučné neurónové siete všeobecne a to hlavne ich architektúra, a odlišnosť od klasických neurónových sietí. Nasledujúca časť bola zameraná na vytvorenie rešerši na tému detekcie objektov dvomi vybranými metódami, ktoré využívajú architektúru konvolučných neurónových sietí: Konvolučná neurónová sieť založená na návrhu regiónov (z angl. „Faster“ R-CNN) spolu s jej 3 verziami a „You Only Look Once“. V oboch týchto častiach boli popísané metódy detekcie a architektúra modelov.

V nasledujúcej kapitole je charakterizovaná zvolená sada dát pre detekciu, ktorú tvoria abdominálne snímky z výpočetnej tomografie (CT) a z magnetickej rezonančnej tomografie (MRI) a je tu zachytený hlavne spôsob získania rezov, tvorba anotácií a augmentácia. Ako detekovaný objekt bola vybraná pečeň, ktorej veľkosť a tvar sa významne odlišuje v rámci rezov získaných od 40 pacientov z oboch modalít. Sada dát taktiež obsahuje asi 15 % netypických tvarov pečene a v rámci CT snímkov je pozorované variabilné rozpätie Hounsfieldovej jednotky pre pečeňové tkanivo [1].

Realizácia „Faster“ R-CNN a „You Only Look Once“ metódy pre detekciu pečene z dostupných snímkov je predstavená v piatej kapitole, ktorá sa zaoberá hlavne tvorbou modelu, výpočtom chyby a učením týchto sietí so zvolenými parametrami. V poslednej kapitole sú zhrnuté a diskutované dosiahnuté výsledky detekcie získané implementovanými metódami a možné využitie v medicíne.

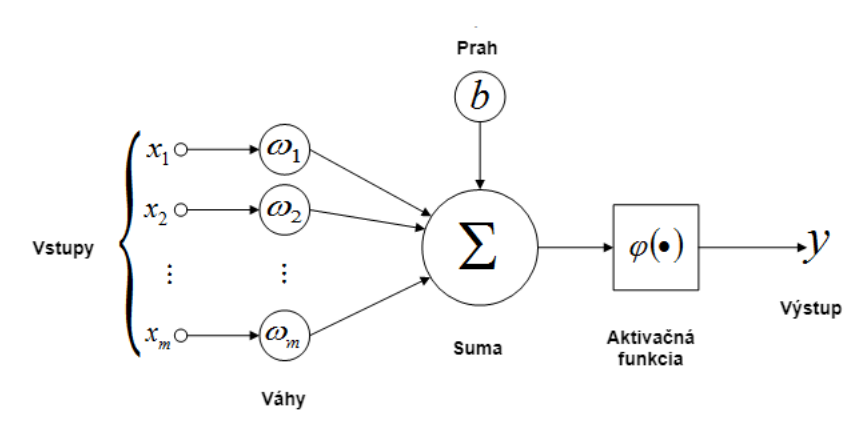
1 Neurónové siete

1.1 Biologický neurón

Neurón je základná stavebná a funkčná jednotka nervovej sústavy. Je schopný prijímať, viesť, spracovávať a odovzdávať určitú formu informácie. Vďaka svojim funkčným spojom tzv. synapsiám dokáže vytvárať kontakty s ostatnými neurónmi. Nervová bunka sa skladá z tela, axónu, ktorým sa šíri informácia smerom od tela neurónu a dendridov, ktoré prijímajú vstupné informácie [2].

1.2 Matematický model neurónu (perceptrón)

Matematický model neurónu je základná jednotka umelej neurónovej siete, ktorá bola inšpirovaná funkciou biologického neurónu. Najjednoduchší umelý neurón, perceptrón, bol predstavený pánom Frankom Rosenblattom v roku 1957 a bol inšpirovaný prácou bádateľov Warrena McCulloch a Waltera Pitts. Skladá sa len z jedného neurónu, ktorý má n -vstupov a so všetkými je prepojený [3].



Obr. 1.1: Matematický model umelého neurónu, (upravené z [4]).

Na obrázku 1.1 je zobrazená štruktúra umelého neurónu. Predpokladáme, že vstupy označené ako x_0, x_1, \dots, x_n reprezentujú informácie, ktoré do biologického neurónu vstupujú pomocou dendridov. Tieto hodnoty sú váhované vektorom váh w_0, w_1, \dots, w_n , ktorý modeluje synaptické spojenia a tým vyjadruje významnosť jednotlivých vstupov. Všetky váhované vstupy sú následne sčítané spoločne s tzv. prahom označovaným aj ako „bias“. Táto suma ďalej vstupuje do aktivačnej funkcie φ , ktorá podmieňuje, či neurón bude aktivovaný. Výsledok z tejto aktivačnej funkcie je odoslaný na výstup [3].

Rovnica 1.1 matematicky popisuje chovanie modelu umelého neurónu.

$$y = f\left[\sum_{i=1}^N w_i x_i - b\right] = f\left[\sum_{i=0}^N w_i x_i\right]; \quad x_0 = 1 \text{ a } w_0 = -b, \quad (1.1)$$

kde f reprezentuje aktivačnú funkciu neurónu, w_i reprezentuje i -tú váhu, x_i reprezentuje i -tý vstup, b reprezentuje prah (bias) a N reprezentuje počet vstupov.

Výber aktivačnej funkcie závisí na druhu riešeného problému. K najpoužívanejším patria tieto typy:

- Skoková aktivačná funkcia

$$f(\alpha) = \text{sign}(\alpha), \text{ kde } f(\alpha) = \{-1, 1\} \quad (1.2)$$

- Linerárna aktivačná funkcia

$$f(\alpha) = \alpha \quad (1.3)$$

Logická sigmoida

$$f(\alpha) = \frac{1}{1 + e^{-\frac{\alpha}{T}}}, \text{ kde } f(\alpha) = \langle -1, 1 \rangle \quad (1.4)$$

- Hyperbolický tangent

$$f(\alpha) = \frac{1}{1 + e^{-\frac{\alpha}{T}}} - 1, \text{ kde } f(\alpha) = \langle 0, 1 \rangle \quad (1.5)$$

kde α označuje vstup do aktivačnej funkcie a $f(\alpha)$ výstup z aktivačnej funkcie. Skoková funkcia je nelineárna aktivačná funkcia a používa sa zväčša pre binárnu klasifikáciu. Úloha riešená lineárnou aktivačnou funkciou je prevažne aproximácia ľubovoľnej funkcie alebo predikcia. Logická sigmoida a hyperbolický tangent sú funkcie, ktoré sa používajú na riešenie nelineárnych problémov, napríklad klasifikácia alebo spracovanie informácií v neurónoch skrytých vrstiev neurónovej siete [3].

Perceptrón pracuje ako binárny klasifikátor, jeho úlohou v procese tréningu je naučiť váhy rozdeľovať vstupnú dátovú množinu do dvoch kategórií. Na výstupe sa objaví číslo 0 alebo 1 podľa toho, či je váhovaný vstup menší alebo väčší ako prah.

$$výstup = \begin{cases} 0 & \text{ak } \sum_{i=1}^N w_i x_i - b \leq 0 \\ 1 & \text{ak } \sum_{i=1}^N w_i x_i - b > 0, \end{cases} \quad (1.6)$$

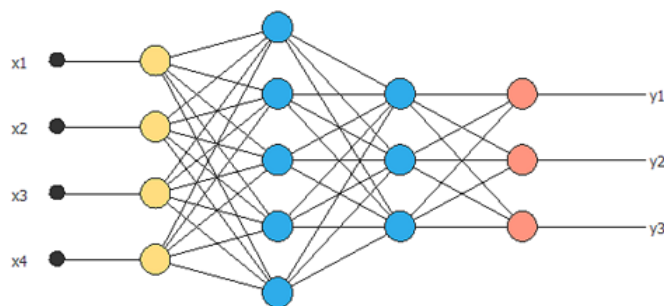
kde w_i znamená i -tá váha, x_i je i -tý vstup, b je prah (bias) a N je počet vstupov.

Perceptrón dokáže pracovať len so vstupnými dátami, ktorých množina je lineárne separovateľná. Problém exkluzívny OR (XOR) je jedným perceptrónom neriešiteľný [3].

1.3 Viacvrstvá neurónová sieť

Viacvrstvá neurónová sieť obsahuje minimálne 3 vrstvy, pričom každá z vrstiev predstavuje skupinku neurónov. Počet neurónov vo vstupnej a výstupnej vrstve závisí na počte vstupných a požadovaných výstupných parametrov. Okrem týchto dvoch spomínaných vrstiev sa medzi nimi nachádzajú aj tzv. skryté vrstvy.

Úlohou vstupnej vrstvy je distribuovať signál do ostatných vrstiev bez zmeny. Ako aktivačná funkcia je zvolená lineárna funkcia. Výstup zo vstupnej vrstvy je zároveň vstupom do vrstvy skrytej. Každý neurón z jednej vrstvy je prepojený s každým neurónom z vrstvy nasledujúcej, avšak nedochádza k prepojeniam neurónov v tej istej vrstve. S väčším počtom skrytých vrstiev sa stáva neurónová sieť sofistikovanejšou, zvyšuje sa jej presnosť, prehĺbuje sa. Vo výstupnej vrstve dochádza k finálnym výpočtom [3].



Obr. 1.2: Viacvrstvá neurónová sieť, (prevzaté z [5]).

1.4 Topológia neurónovej siete

Neurónové siete sa delia podľa šírenia signálu na dopredné neurónové siete a rekurentné neurónové siete. V doprednej neurónovej sieti sa informácia šíri od vstupnej cez skryté vrstvy až na výstup len jedným smerom. Nenájdeme tu žiadnu spätnú väzbu.

U rekurentnej neurónovej siete funguje spätnoväzobné šírenie signálu. Spojenie medzi jednotlivými vrstvami sú vedené oboma smermi pomocou spätných cyklov. Ich učenie prebieha nie len počas tréningu, ale aj počas vytvárania výstupu zo vstupu privedeného do siete. Tento typ neurónovej siete má schopnosť pamätať si výstup z predchádzajúceho kroku a to na základe prítomnosti tzv. skrytého stavu a rekurentného spojenia. Využívajú sa v oblasti rozpoznávania reči, rukopisov a to vďaka ich schopnosti spracovávať variabilnú dĺžku vstupnej sekvencie [6].

1.5 Učenie neurónovej siete

Po zostavení štruktúry neurónovej siete je model pripravený na tréning. Existujú dva prístupy k učeniu neurónovej siete a to učenie s učiteľom a učenie bez učiteľa. Pri tréningu s učiteľom sú prítomné 2 vektory: vstupné hodnoty a požadované výstupné hodnoty. Na začiatku sú váhy a prah nastavené náhodne a začína proces učenia. V procese tréningu bude sieť spočiatku poskytovať nedostačujúce výsledky a po niekoľkých epochách táto sieť bude dosahovať vyššiu presnosť. Epocha značí jednu iteráciu učenia, ktorá prebieha na všetkých tréningových vzorkách. Neurónová sieť spracuje vstupné hodnoty, privedie ich na výstup, ktorý sa následne porovná s požadovaným výstupným vektorom. Jedna z možných metrík stanovenia chybovej funkcie je stredná kvadratická odchýlka, ktorá indikuje ako veľmi sa líši výstup siete od optimálneho výsledku [7].

$$L(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2, \quad (1.7)$$

kde \hat{y}_i je požadovaný výstup z neurónovej siete, y_i je skutočný výstup z neurónovej siete a m je počet vstupov.

Chybová funkcia bude po prvej epoche poskytovať vysoké hodnoty. Na to, aby bol dosiahnutý akceptovateľný výstup je potrebné minimalizovať chybu. Jedným z možných postupov je metóda gradientného zostupu. Z matematického hľadiska je gradient vektorom prvých derivácií jednotlivých premenných funkcie. Gradient udáva smer najstrmsšieho rastu chybovej funkcie. Metóda gradientného zostupu sa využíva pre korekciu váh, pretože vďaka ich aktualizácii je neurónová sieť schopná poskytovať presnejšie výsledky. Ak sa po vypočítaní parciálnej derivácie chybovej funkcie podľa váhy budeme po malých krokoch učenia posúvať proti smeru gradientu, tak chybová funkcia bude postupne konvergovať do lokálneho minima. Krok učenia je zvyčajne malé číslo, ktoré umožňuje pomalú a hladkú aktualizáciu váh tak, aby lokálne minimum chybovej funkcie nebolo prekročené. Matematicky môže byť aktualizácia váh pomocou metódy gradientného zostupu zapísaná ako:

$$w_{nové} = w_{staré} - \mu \nabla L, \quad (1.8)$$

kde $w_{nové}$ sú aktualizované váhy, $w_{staré}$ sú pôvodné váhy, μ je krok učenia a ∇L je gradient chybovej funkcie.

Ak by bol gradient chybovej funkcie počítaný z každého vzorku tréningovej množiny dát, bolo by to neúčinné a zdlhavé, preto sa využíva metóda stochastického gradientného zostupu, pri ktorej je vypočítaný približný gradient chyby z určitej množiny vstupných dát. Nie je garantované, že chybová funkcia bude klesať v každej iterácii, je „zašumená“, bude kolísať, avšak po čase klesne [7].

Keďže informácia o požadovaných výstupoch zo skrytých vrstiev nie je dostupná, pre aktualizáciu parametrov neurónov zo skrytých vrstiev sa využíva metóda spätného šírenia chyby. Chyba vo vrstve n sa rozdelí medzi neuróny vo vrstve $n-1$. Takto je chyba pretláčaná späť k vstupu, proti smeru výpočtu výstupov siete a podľa nej sú aktualizované váhy v systéme. Tento postup sa opakuje niekoľkokrát so snahou minimalizovať výsledok chybovej funkcie až pokiaľ nie je dosiahnutý akceptovateľný výstup [7].

Pri tréningu bez učiteľa nie je prítomný očakávaný výstupný vektor. Algoritmus učenia spočíva v nastavovaní váh len z dostupných vstupných hodnôt [7].

2 Konvolučné neurónové siete

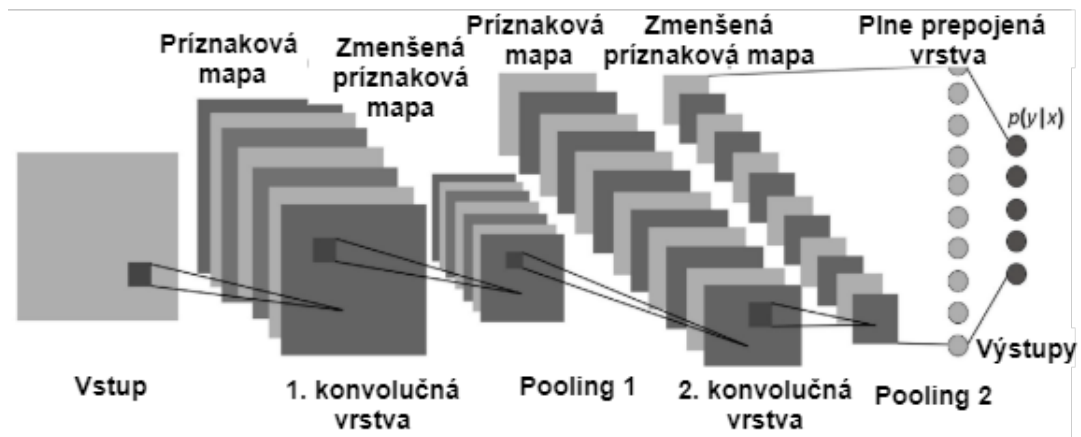
Konvolučné neurónové siete sú špeciálny typ neurónových sietí, ktoré dosiahli významný pokrok vo výskume počítačového videnia. Populárnymi sa stali hlavne po roku 2010 pretože prekonalí výkon doposiaľ známych architektúr neurónových sietí v oblasti klasifikácie a rozpoznávania obrazov. Proces učenia konvolučných neurónových sietí (CNN) simuluje organizáciu vizuálneho centra mozgovej kôry a úspešne natrénovaná CNN môže vytvoriť hierarchickú informáciu o rohoch, tvaroch alebo komponentoch objektov v klasifikácii obrázkov počas predspracovania [8].

Ako už bolo zmienené, celá myšlienka CNN je inšpirovaná biológiou oka. Za to, že svet vnímame ako farebný a detailný obraz vďaka súborom procesov odohrávajúcich sa v našom mozgu. Na vnútornej časti očnej steny, v zadnej časti oka sa nachádza vrstva citlivá na svetlo nazývaná sietnica lat. retina. Elementami, ktoré reagujú na množstvo dopadajúceho svetla sú u ľudí čapíky a tyčinky, spolu nazývané fotoreceptory. Fotoreceptory a nervové bunky v sietnici vytvárajú zložitú priečnu a pozdĺžnu neurónovú sieť. Niekoľko fotoreceptorov je synapticky viazaných na jednu bipolárnu bunku, ktorá reaguje len ak väčšina fotoreceptorov na ňu pripojených je aktívnych. Ďalej niekoľko bipolárnych buniek predáva informácie jednej gangliovej bunke, ktorá slúži na spracovanie informácii o veľkosti a intenzite svetelného podnetu. Takisto aj gangliová bunka reaguje len za predpokladu aktivity väčšiny bipolárnych buniek na ňu pripojených. Dochádza k výraznej priestorovej sumácii. Informácia je ďalej odvádzaná axonom do vyšších zrkových centier. Čím hlbšie do mozgu postupujeme, tým viac konkrétnych príznakov z obrazu sveta je detekovaných neurónmi [9].

Konvolučná neurónová sieť je druh neurónovej siete, ktorá obsahuje aspoň jednu konvolučnú vrstvu. CNN používajú koncept kombinovania príznakov nízkeho levelu z obrázku do príznakov levelu vyššieho až pokiaľ je sieť schopná reagovať na konkrétne objekty, podobne ako vo vizuálnom centre. Následne sa použije klasická viacvrstvová neurónová sieť, ktorá kombinuje extrahované príznaky do zmyslupného kontextu. CNN predstavujú oproti viacvrstvovej neurónovej sieti výhodu v získavaní lokálnych informácií napr. zo susedných pixelov a v zároveň znižujú počet jednotiek v sieti [10].

2.1 Architektúra CNN

Konvolučná neurónová sieť sa skladá z niekoľkých výpočetných vrstiev neurónov a to konvolučných vrstiev, podvzorkovacích vrstiev a plne prepojených vrstiev. Na obrázku 2.1 je možné vidieť návrh jednoduchšej konvolučnej siete zloženej zo spomínaných vrstiev.

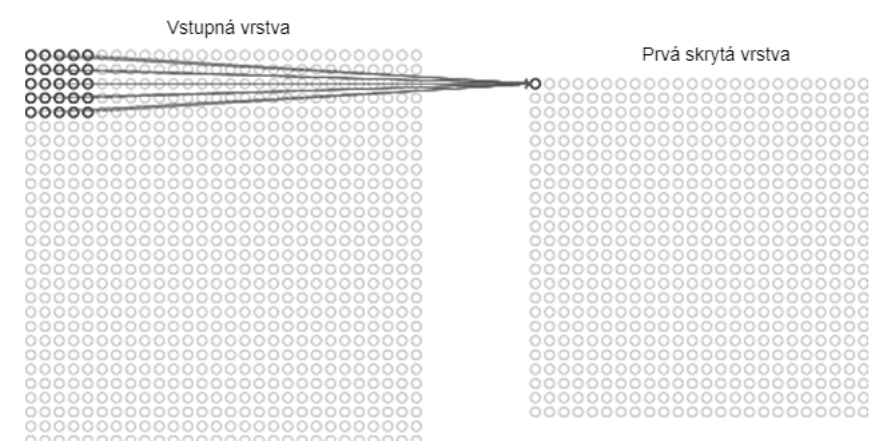


Obr. 2.1: Architektúra konvolučnej neurónovej siete v ktorej sa nachádzajú 3 typy vrstiev: konvolučná, podvzorkovacia („pooling“) a plne prepojená vrstva, (upravené z [11]).

2.1.1 Konvolučné vrstvy

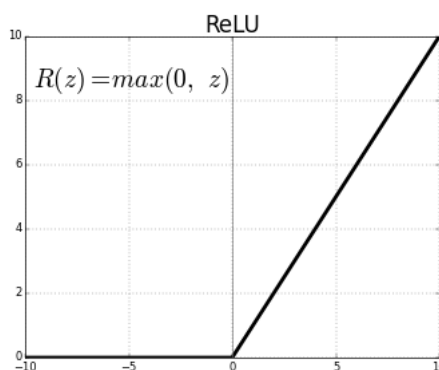
Hlavný zmysel konvolučných vrstiev je extrakcia rozlišujúcich príznakov zo vstupného obrazu. Matematická operácia, ktorá je aplikovaná na obraz v podobe filtru sa nazýva konvolúcia. „Kernel“ (jadro), reprezentujúci filter, je matica čísel, ktoré simulujú váhy. Matica s určitým počtom riadkov a stĺpcov sa posúva inak povedané konvuluje po vstupnom obraze s určitým počtom pixelov. Región v ktorom sa filter v danom čase nachádza sa nazýva recepčné pole. V jednotlivých polohách filtru sú vynásobené hodnoty filtru s hodnotami pixelov daného regiónu obrazu a výsledné súčiny sú následne sčítané a uložené do matice. Po posunutí filtru po všetkých pozíciách obrazu s určitým krokom, získame maticu hodnôt nazývanú príznaková mapa. Čím vyšší je krok tzv. „stride“ s ktorým sa filter posúva po obraze, tým menšiu príznakovú mapu získame [8].

Každá vrstva ma špecifické množstvo filtrov a ich počet udáva počet príznakových máp a tým hĺbku danej vrstvy. Každý z týchto filtrov môže byť považovaný za identifikátora pre určité príznaky. V počiatočných vrstvách siete sú rozlišované jednoduché vizuálne elementy ako hrany, rohy alebo krivky. V hlbších vrstvách nachádzame filtre špecializujúce sa na odlišenie sofistikovanejších objektov ako oči, uši, vlasy atď. Filtre sú zložené z neurónov, ktoré medzi sebou zdieľajú množinu váh, čo znižuje pamäťovú náročnosť a urýchľuje výpočet. Čím vyšší je počet filtrov v danej vrstve tým viac informácií získavame o vstupnom obraze. V niektorých prípadoch je vhodné vyplniť vstupnú maticu pixelov nulami okolo okrajov. Táto operácia je v anglickej literatúre nazývaná „zero padding“ a vďaka nej je možné extrahovať príznaky z hraničných častí obrázka a taktiež kontrolovať veľkosť príznakovkej mapy [10].



Obr. 2.2: Ukážka aplikácie filtru na vstupné pixely obrázka, (upravené z [10]).

Po každej konvolúcii je aplikovaná dodatočná nelineárna operácia nazývaná „Rectified Linear Unit“ (ReLU), ktorá umožňuje nahradiť každú zápornú hodnotu aktívacie príznakovej mapy nulou. Zmysel tejto operácie je zaradiť nelinearitu do konvolučnej neurónovej siete, pretože väčšina reálnych dát s ktorými bude sieť pracovať nie je lineárnych [10].

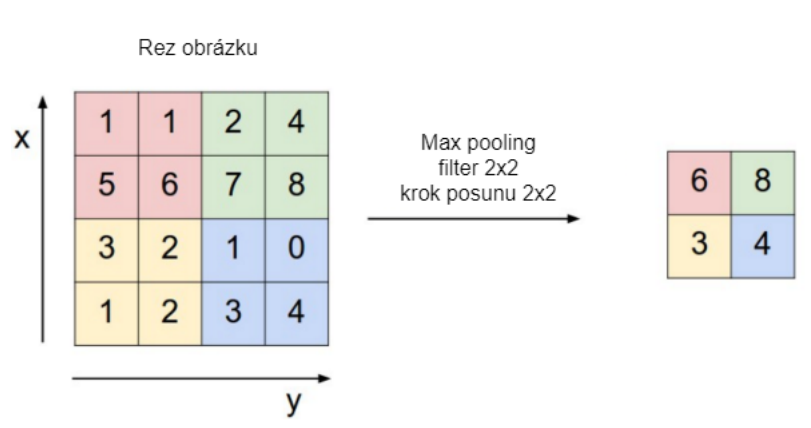


Obr. 2.3: Funkcia ReLU, (prevzaté z [12]).

2.1.2 Podvzorkovacie vrstvy („Pooling vrstvy“)

Úlohou podvzorkovacích (z angl. „pooling“) vrstiev je redukovať priestorové rozlíšenie príznakových máp, pričom ostanú nemenné voči tvaru a pozícii významných príznakov, ktoré boli v obraze detekované. Po aplikácii filtrov konvolučných vrstiev je definované okolie pixelov zjednotené do jednej hodnoty. Zvyčajne sa používajú „MaxPooling vrstvy“, ktoré zo susedných pixelov vyberajú maximum, ale môžu byť

použité aj iné metódy ako „MinPooling“, „AveragePooling“ alebo „MedianPooling“. Myšlienka ostáva stále tá istá a to vybrať susedné hodnoty pixelov z príznakovej mapy, porovnať ich pomocou funkcie (max, min, priemer atď.) a odoslať jednu hodnotu na výstup. Narozdiel od konvolučných vrstiev, pri aplikácii filtru na pixely príznakovej mapy a následnom posune, sa pixely zvyčajne neprekrývajú [8].



Obr. 2.4: Ukážka operácie „MaxPooling“, (upravené z [13]).

2.1.3 Plne prepojené vrstvy

Konvolučné a podvzorkovacie vrstvy sa opakujú niekoľkokrát a na záver sú spojené všetky príznaky z obrázka pomocou plne prepojených vrstiev, ktoré predstavujú tradičnú viacvrstvovú neurónovú sieť. Ako vstup do plne prepojených vrstiev je príznaková mapa z poslednej konvolučnej a podvzorkovacej vrstvy, ktorá reprezentuje sofistikované príznaky. Výstup z plne prepojených vrstiev je N-dimenzionálny vektor, ktorý v prípade klasifikačnej úlohy reprezentuje pravdepodobnosť výskytu N-tried v obrázku [8].

2.2 Učenie konvolučnej neurónovej siete

Pri učení konvolučných neurónových sietí sa využívajú rovnaké postupy ako pri učení jednoduchej viacvrstvovej neurónovej siete. Pre aktualizáciu váh je využívaná metóda spätného šírenia chyby a pre minimalizáciu chybovej funkcie metóda stochastického gradientného zostupu. Ako už bolo spomínané, CNN sa skladajú z niekoľkých typov vrstiev, preto je proces učenia modifikovaný pre každú z týchto vrstiev [7].

3 Detekcia objektov

Detekcia objektov ako súčasť počítačového videnia je založená na lokalizácii objektov v obraze a ich vymedzením pomocou ohraničujúceho rámčeka. Detekcia v sebe okrem lokalizácie zahŕňa aj klasifikáciu objektov do tried. Nie vždy je počet rámečkov reprezentujúcich objekt dopredu známy. Variabilný počet výstupov je hlavný dôvod, prečo sa táto výpočetná úloha nedá riešiť klasickou architektúrou konvolučnej neurónovej siete nasledovanou plne prepojenými vrstvami. Ako jeden z možných prístupov pre vyriešenie tohto problému sa ukázalo nachádzanie regiónov záujmu z obrazu a použitie CNN klasifikátora pre overenie prítomnosti objektu v danom regióne. Problém s týmto prístupom je v tom, že predmet záujmu môže mať rôzne priestorové umiestnenia či rozmery v obraze a bolo by nutné vybrať veľké množstvo regiónov, čo by bolo, samozrejme, výpočetne veľmi náročné [8, 14].

Do metód strojového učenia, ktoré sa používajú na detekciu objektov môžeme zaradiť histogramovo orientované gradienty (HOG), Haarove príznaky alebo škálovo invariantné transformácie príznakov (SIFT). V dnešnej dobe sú však čoraz viac populárne metódy hlbokého učenia vďaka svojej presnosti a rýchlosti. Najčastejšie využívané algoritmy v detekcii objektov sú sieť založená na návrhu regiónov (R-CNN) a jej 3 verzie, „You Only Look Once“ (YOLO) spolu s jeho štyrmi variantami, „Single Shot Multibox Detector“ (SSD) alebo R-FCN, ktoré je založené na návrhu regiónov pomocou plne prepojených vrstiev [15].

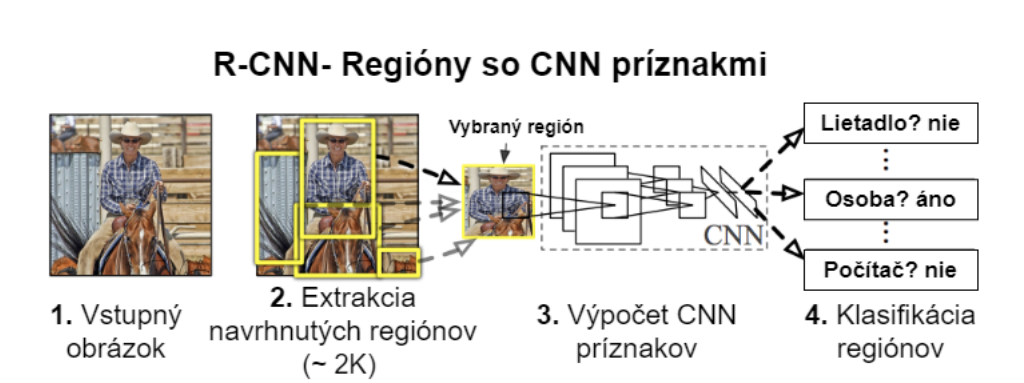
3.1 Konvolučná neurónová sieť založená na návrhu regiónov

Problém so selekciou veľkého množstva regiónov obišiel Ross Girshick a kolektív, ktorí navrhli metódu nazývanú „Region-based convolutional neural network (R-CNN)“, teda konvolučnú neurónovú sieť založenú na návrhu regiónov, kde pomocou selektívneho vyhľadávania extrahujú z obrazu iba 2 000 regiónov. Detekcia objektov pomocou R-CNN sa skladá z 3 modulov. Prvým z nich je kategoricky nezávislý návrh regiónov. Tieto návrhy definujú súbor kandidátov pre detekciu, ktoré má detektor k dispozícii. Selektívne vyhľadávanie začína počiatočnou segmentáciou založenou na intenzite pixelov obrazu použitím grafovej segmentačnej metódy. Algoritmus selektívneho vyhľadávania je iteratívny proces pri ktorom sú priradené návrhy regiónov k segmentovaným častiam obrazu a následne sú segmenty zlučované na základe podobnosti [16].

Druhým modulom je konvolučná neurónová sieť, ktorá z regiónov extrahuje vektory príznakov jednotnej dĺžky. Extrahované sú 4096-dimenzionálne vektory z kaž-

dého získaného návrhu regiónu. Tieto príznaky vstupujú do siete zloženej z piatich konvolučných a dvoch plne prepojených vrstiev.

V poslednom module sú extrahované príznaky klasifikované pomocou metódy podporných vektorov podľa prítomnosti objektu v navrhovanej oblasti. Cieľom metódy podporných vektorov je nájsť hyperrovinu v N-rozmernom priestore, ktorá zreteľne klasifikuje dátové body v tomto priestore. Okrem predpovede prítomnosti objektu v návrhoch regiónov, algoritmus predpovedá aj 4 ofsetové hodnoty, aby sa zvýšila presnosť rámčeka. Na obrázku 3.1 je možné vidieť chovanie R-CNN [14, 15].



Obr. 3.1: Ukážka detekčného systému R-CNN. Z obrázku je navrhnutých 2 000 regiónov, z ktorých sú extrahované príznakové vektory pomocou CNN pre výpočet príznakov (5 konvolučných a 2 plne prepojené vrstvy) a tie sú následne klasifikované metódou podporných vektorov (upravené z [15]).

Aj napriek pokrokom v oblasti detekcie objektov, ktoré sú poháňané úspešným návrhom regiónov v obrazoch a konvolučnými neurónovými sieťami založenými na navrhnutých regiónoch, R-CNN sa stretáva s radou problémov. Jedným z hlavných je časová náročnosť tréningu siete, keďže je potrebné klasifikovať 2000 návrhov regiónov na jeden obrázok. Taktiež táto metóda nemôže byť implementovaná real time, pretože na detekciu u každého obrazu z testovacej množiny je potrebných 47 sekúnd. Algoritmus selektívneho vyhľadávania je nemenný, v tomto štádiu sa neodohráva proces učenia, preto môže dôjsť k výberu nevhodných kandidátov na regióny [14, 15].

3.2 „Fast“ R-CNN

Autor článku z podkapitoly 3.1 vyriešil niektoré z nevýhod R-CNN a vytvoril algoritmus rýchlejšej detekcie objektov, ktorý nazval „Fast Region-based convolutional neural network (Fast R-CNN)“. Obrázok 3.2 ilustruje architektúru siete. Prístup je

podobný ako u R-CNN algoritmu, avšak u zrýchleného R-CNN je vstupom do siete celý obrázok a sada návrhov objektov. Obrázok je najprv spracovaný v konvolučnej neurónovej sieti s niekoľkými konvolučnými a „maxpooling“ vrstvami (v obrázku 3.2 ako hlboká konvolučná sieť), ktorej výstupom je príznaková mapa.

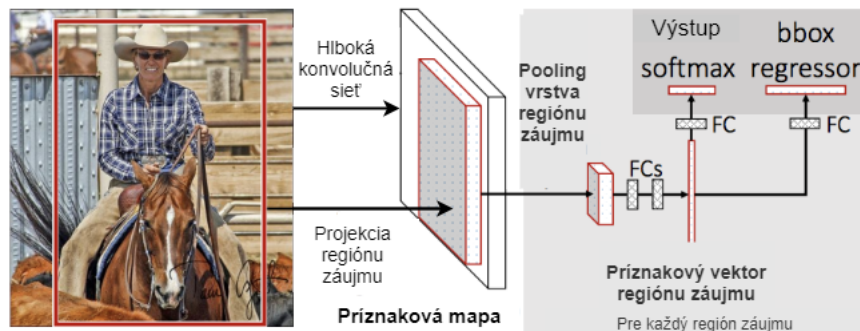
Ďalej je z návrhu regiónov pomocou podvzorkovacej vrstvy regiónu záujmu (označovaný ako „Region of Interest“ (RoI)) extrahovaný príznakový vektor jednotnej dĺžky. Región záujmu je definovaný ako obdĺžnikové okno, ktoré je v tejto vrstve rozdelené do mriežky tzv. pod-okien na ktorých hodnoty je aplikovaný „MaxPooling“.

Každý príznakový vektor vstupuje do sekvencie plne prepojených vrstiev (v obrázku 3.2 ako fc, skratka z angl. „fully connected“). Tieto vrstvy sa napokon vetvia do dvoch, z ktorej jedna produkuje „softmax“ pravdepodobnosti tried navrhnutých regiónov a druhá súradnice ohraničujúcich rámcikov („bounding boxov“), ktoré ohraničujú objekty zaradené do daných tried (v obrázku 3.2 ako „bbox regressor“).

„Softmax“ vrstva využíva „softmax“ aktivačnú funkciu, ktorá premieňa numerický výstup z predchádzajúcej vrstvy na pravdepodobnosti, ktorých súčet je jedna. Vzorec pre výpočet pravdepodobností sa nachádza v rovnici 3.1.

$$P(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}, \quad (3.1)$$

kde y_i je vstup do funkcie, $\sum_j e^{y_j}$ je súčet vstupov nachádzajúcich sa v exponente e Eulerovho čísla a $P(y_i)$ je výstupná pravdepodobnosť.



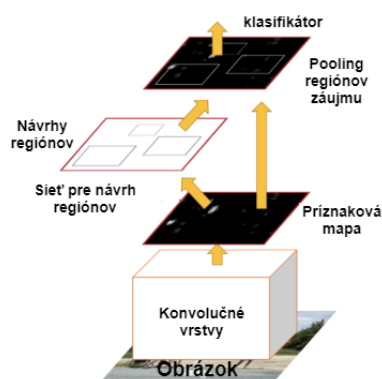
Obr. 3.2: Ukážka detekčného systému Fast R-CNN, (upravené z [17]).

„Fast“ R-CNN je rýchlejšie ako R-CNN, pretože nie je potrebné, aby do konvolučnej siete vždy vstupovalo 2000 návrhov regiónov. Namiesto toho je konvolučná operácia vykonaná len raz za jeden obrázok a z toho je generovaná príznaková mapa [14, 17].

3.3 „Faster“ R-CNN

Počiatočná výpočetná náročnosť u CNN založená na návrhu regiónov bola redukovaná vďaka zdieľaniu konvolúcie medzi návrhmi regiónov. „Fast“ R-CNN algoritmus sa blíži k detekovaniu objektov v reálnom čase, ak si odmyslíme čas potrebný k návrhu regiónov. U oboch predchádzajúcich metód sú regióny navrhované pomocou selektívneho vyhľadávania, čo je časovo náročný proces, čím je ovplyvnená výkonnosť siete. Pri porovnaní testovacích časov „Fast“ R-CNN s použitím navrhovaných regiónov, tak algoritmus sa výrazne spomalí oproti prípadu, keď návrhy regiónov použité nie sú. Práve návrhy regiónov sú výpočetnou prekážkou v najmodernejšej technike pre detekčné systémy [18].

Systém detekcie objektov, takzvaný „Faster Region-based convolutional neural network (Faster R-CNN)“ je zložený z dvoch modulov. Prvým z nich je konvolučná neurónová sieť označovaná ako „Region Proposal Network“, ktorá z obrazu vyberá jednotlivé regióny, teda ukazuje druhému modulu kam sa pozrieť a v akých častiach obrazu sa pravdepodobne nachádza objekt. Druhou časťou „Faster“ R-CNN systému je „Fast“ R-CNN detektor, ktorý využíva vybraný región [18]. Celý systém zobrazený na obrázku 3.3 je jedna zjednotená sieť slúžiaca pre detekciu objektov.



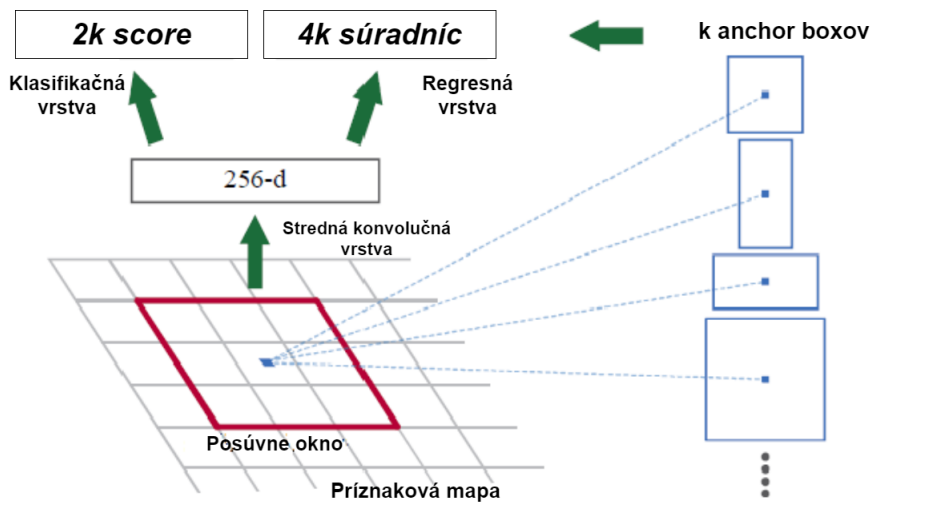
Obr. 3.3: Ukážka zjednoteného detekčného systému „Faster“ R-CNN, kde konvolučné vrstvy symbolizujú sieť pre extrakciu príznakov, nasleduje sieť pre návrh regiónov, ktorá využíva príznakovú mapu získanú v predchádzajúcom kroku. Do detekčnej siete vstupujú (znázornené šípkami) návrhy regiónov spolu s príznakovou mapou a po podvzorkovní regiónov záujmu sú regióny klasifikované, (upravené z [18]).

3.3.1 Sieť pre návrh regiónov

Na vstup siete pre návrh regiónov (RPN) je privádzaný obraz akejkoľvek veľkosti. Na výstup tejto siete je odosielaný súbor pravouhlých objektov, pričom každému

objektu prislúcha skóre, ktoré určuje vzťah medzi triedami objektov a pozadím. Na stavbu RPN sa využíva CNN a predpokladom je, že RPN zdieľa konvolučné vrstvy s „Fast“ R-CNN detektorom objektov.

Podobne ako vo „Fast“ R-CNN je obrázok najprv privádzaný na vstup konvolučnej neurónovej siete, ktorá slúži na extrakciu príznačkov z obrázka a jej výstupom je príznačková mapa. Pri navrhovaní regiónov je na výstupe z poslednej konvolučnej vrstvy prítomná malá sieť. Tá vezme ako vstup okno o veľkosti $n \times n$ z príznačkovvej mapy. Každé posuvné okno je premietnuté do príznačkov nižšej dimenzie. Tento príznačok ďalej vstupuje do dvoch príbuzných plne prepojených vrstiev, tak ako je zobrazené na obrázku 3.4. Prvou z nich je regresná vrstva, ktorej výstupom sú súradnice návrhov regiónov a druhou klasifikačná vrstva, ktorá určuje, či sa jedná o pozadie alebo je prítomný objekt.



Obr. 3.4: Ukážka siete pre návrh regiónov (RPN), (upravené z [18].)

Malá sieť pre návrh regiónov je na obrázku 3.4 zobrazená na jednej pozícii príznačkovvej mapy. Táto mini sieť sa pohybuje po príznačkovvej mape pomocou posuvného okna.

V každej pozícii posuvného okna je zároveň predpokladaných niekoľko návrhov regiónov, kde „ k “ určuje počet maximálne predpokladaných návrhov regiónov v danej pozícii. V regresnej vrstve sa nachádza $4k$ výstupov, ktoré kódujú súradnice k -rámčekov. Výstup z klasifikačnej vrstvy je $2k$ a určuje pravdepodobnosť prítomnosti objektu pre každý návrh regiónu. Počet návrhov k závisí na k -referenčných rámečkoch, ktoré sa označujú ako tzv. „anchors“ alebo kotvy. Myšlienkou „anchors“ je predefinovať tvary, ktoré následne môžeme asociovať s našimi predpokladanými návrhmi regiónov. Počet „anchors“ na príznačkovú mapu sa vypočíta ako $w \times H \times k$,

kde $w \times H$ sú rozmery príznakovkej mapy, čo znamená že k anchors rámciekov je vytvorených pre každý pixel príznakovkej mapy.

„Anchors“ rámcieky a funkcia, ktorá vypočítava návrhy regiónov pomocou týchto rámciekov sú translačne invariantné, to znamená, že ak je v obrázku posunutý objekt, tak funkcia, ktorá vypočítava návrh regiónu by mala byť schopná predikovať región v akejkoľvek pozícii. Vďaka translačne invariantnej metóde sa zároveň znižuje veľkosť modelu. Využívaná metóda klasifikuje a predpokladá ohraničujúce rámcieky na základe referenčných „anchors“ rámciekov s rôznymi pomermi strán [18].

Po výstupe návrhov regiónov z RPN, tieto návrhy vstupujú do „Fast“ R-CNN detektoru spolu s príznakovou mapou získanou extrakciou príznakov z obrázka pomocou konvolučnej neurónovej siete. Pre extrakciu môžu byť použité architektúry sietí ako ResNet, VGG16 alebo MobileNet. V detekčnej sieti sú príznaky spolu s regiónmi spracované podvzorkovacou vrstvou regiónu záujmu, plne prepojenými vrstvami a výstupnou „softmax“, a regresnou vrstvou, tak ako je popísané v podkapitole 3.2.

3.3.2 Tréning „Faster“ R-CNN

Sieť pre návrh regiónov a „Fast“ R-CNN detektor môžu byť tréňované metódou spätného šírenia chyby a ako optimalizačný algoritmus môže byť využitý stochastický gradientný zostup. Pri tréningu „Faster“ R-CNN algoritmu je nutné zohľadniť, že sa jedná o systém zložený z 2 modulov. Ak by boli tieto dve siete tréňované nezávisle, tak by boli rôzne modifikované ich konvolučné vrstvy. Cieľom je vyvinúť mechanizmus, ktorý zohľadňuje spojenie týchto dvoch častí architektúry. Autori preto uvádzajú 4–krokový striedavý tréning:

1. V prvom kroku je tréňovaná RPN pričom konvolučné vrstvy sú inicializované z predtréňovanej siete a ich parametre sú doladené.
2. Nasleduje tréning samostatnej detekčnej siete „Fast“ R-CNN, pričom sa využívajú návrhy regiónov z kroku jedna. Po tomto kroku siete medzi sebou zatiaľ nezdieľajú konvolučné vrstvy.
3. Ďalší krok predstavuje inicializáciu tréningu RPN výstupom z detektoru „Fast“ R-CNN. Týmto dochádza k zdieľaniu konvolučných vrstiev, ktoré už nie sú ďalej tréňované, ale doladujú sa vrstvy, ktoré sú využívané len RPN.
4. Nakoniec sú doladené vrstvy, ktoré využíva „Fast“ R-CNN detektor a konvolučné vrstvy ostávajú nezmenené. Po tomto kroku je natréňovaný celý detekčný systém „Faster“ R-CNN [18].

3.4 „You Only Look Once“

Doteraz boli spomínané algoritmy, ktoré sa nevnímali obrázkov ako celok, ale pozerali len na časť obrázka v ktorej je veľká pravdepodobnosť výskytu objektu. Pomocou systému „You Only Look Once (YOLO)“ jediná konvolučná sieť dokáže priamo z pixelov celého obrázku predvídať niekoľko ohraničujúcich rámcikov a zároveň pravdepodobnosti tried daných rámcikov. Tréning YOLO prebieha na celých obrázkoch a výkon detekcie je priamo optimalizovaný. Detekcia objektov je v tomto prípade považovaná za jediný regresný problém, vďaka čomu je YOLO algoritmus extrémne rýchly, čo umožňuje detekciu v reálnom čase.

Ďalšou z výhod YOLO je zachytenie celého obrázka počas tréningu aj testovacieho času, narozdiel od techník, ktoré využívajú posuvné okno alebo návrhy regiónov. To umožňuje vnímať súvisiace informácie tried nachádzajúcich sa v obraze, taktiež ich vzhlad a polohu. Tento algoritmus v procese učenia zovšeobecňuje znázornenie objektu. Ak sú tréningové dáta obrázky z prírody a testovacia množina sú umelecké obrazy, YOLO prekonáva R-CNN detekciu.

Napriek tomu, že táto technika detekcie dokáže veľmi rýchlo identifikovať objekty, problémy nastávajú pri lokalizovaní hlavne malých objektov [19].

3.4.1 Zjednotená detekcia

Komponenty, ktoré sú zapojené do detekcie objektov sú v systéme YOLO zjednotené do jednej neurónovej siete. Vďaka vnímaniu celého obrázka sa používajú všetky príznaky v obrázku na vytvorenie každého ohraničujúceho rámcika. Táto sieť predvída rámciky okolo všetkých objektov patriacich k rôznym triedam v obrázku zároveň [19].

Systém YOLO rozdeľuje vstupný obraz do mriežky rozmerov $S \times S$, ak sa stred objektu nachádza v bunke mriežky, tak táto bunka je zodpovedná za detekciu daného objektu. Každá bunka mriežky predvída niekoľko rámcikov s ich „confidence score“, ktoré je možné nazývať aj ako tzv. skóre spoľahlivosti. Toto skóre odráža aká je pravdepodobnosť, že rámcik obsahuje objekt a aký presný je daný rámcik, teda ako dobre vymedzuje daný objekt. „Confidence score“ definujeme ako:

$$Pr(\text{Objekt}) * IOU_{\text{Predikovaný}}^{\text{Pravdivý}} \quad (3.2)$$

Ak sa v danom boxe nenachádza objekt, tak je skóre spoľahlivosti rovné nule, pretože $Pr(\text{Objekt})$, teda pravdepodobnosť výskytu objektu je nulová. V inom prípade je skóre spoľahlivosti rovné spoločnému prieniku (z angl. „Intersection over Union (IoU)“) medzi predvídaným rámcikom a pravdivým rámcikom (z angl. „ground truth box“). Každý ohraničujúci rámcik sa skladá z piatich predikcií: x, y, w, h

a skóre spoľahlivosti, kde súradnice x, y reprezentujú stred ohraničujúceho rámčeka vzhľadom k hraniciam bunky v mriežke a súradnice w, h reprezentujú výšku a šírku vzhľadom k rozmerom celého obrázku.

Každá bunka taktiež predvída podmienené pravdepodobnosti tried, ktoré sú definované ako:

$$Pr(Class_i | Object) \quad (3.3)$$

Podmienie pravdepodobnosti tried udávajú pravdepodobnosť, že objekt vyskytujúci sa v bunke patrí k danej triede. Na jednu bunku pripadá vždy len jedna trieda aj napriek tomu, že cez danú bunku môže prechádzať aj viac rámčekov obklopujúcich rozdielne triedy.

V testovacom čase sú vynásobené podmienené pravdepodobnosti tried so skóre spoľahlivosti a vo výsledku sú získané skóre spoľahlivosti špecifické pre danú triedu v každom rámčeku, ktoré je definované ako:

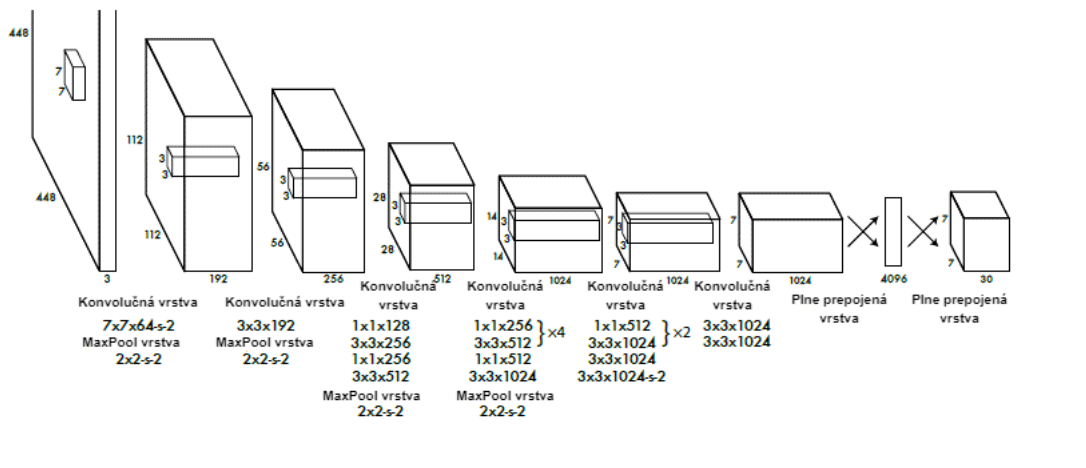
$$Pr(Class_i | Object) * Pr(Object) * IOU_{Pred}^{Truth} = Pr(Class_i) * IOU_{Pred}^{Truth} \quad (3.4)$$

Toto skóre určuje nielen pravdepodobnosť triedy objavujúcej sa v rámčeku, ale aj ako dobre rámček ohraničuje objekt [19].

3.4.2 Dizajn siete

Model siete navrhnutý v článku [19] je tvorený počiatočnými konvolučnými vrstvami, ktorých úlohou je výber príznakov z obrázka a plne prepojenými vrstvami ktoré predikujú výstupné pravdepodobnosti a súradnice. Architektúra siete je inšpirovaná GoogleNet [20] modelom pre klasifikáciu obrazov. Obsahuje 24 konvolučných vrstiev ktoré sú nasledované 2 plne prepojenými vrstvami. Používajú sa 1×1 redukčné vrstvy, ktoré redukujú príznaky z predošlých vrstiev a 3×3 plne prepojené vrstvy. Celá sieť je zobrazená na obrázku 3.5.

Trénujú sa taktiež aj rýchle verzie YOLO, tzv. „Fast“ YOLO, ktoré obsahuje menej konvolučných vrstiev, konkrétne 9 a menej filtrov v týchto vrstvách. Ostatné parametre medzi YOLO a „Fast“ YOLO sú rovnaké. Finálny výstup tejto siete je tenzor predikcií o veľkosti $7 \times 7 \times 30$ [19].



Obr. 3.5: Dizajn siete YOLO. Táto sieť obsahuje 24 konvolučných vrstiev, pričom sa striedajú filtre s veľkosťou 3×3 a 1×1 , prítomné sú aj „MaxPooling“ vrstvy a 2 plne prepojené vrstvy, (upravené z [19]).

3.4.3 Limity YOLO

YOLO predstavuje silné priestorové obmedzenie pri predikciách ohraničujúcich rámečkov, keďže každá bunka predvída len 2 boxy a môže obsahovať len jednu triedu, tým je limitovaný aj počet blízkych objektov, ktoré môžu byť predikované. Tento model má problém s detekciou malých objektov, ktoré sa nachádzajú pokope, napr. kídľ vtákov. Chybová funkcia pomocou ktorej je aproximovaný výkon detekcie rovnako zaobchádza s chybami v malých rámečkoch ako s chybami vo veľkých rámečkoch a je samozrejmé, že malá chyba v malom rámečku má väčší vplyv na IoU [19].

4 Experimentálne dáta

Pre detekciu bol využitý „Chaos challenge“ [21] dataset voľne dostupný na portáli Grand Challenge [22], ktorý obsahuje abdominálne snímky z výpočtovej tomografie (CT) od 20 rôznych pacientov, čo spolu tvorí 2 450 rezov v DICOM formáte (skratka z angl. „Digital Imaging and Communications in Medicine“). Ako detekovaný objekt bola zvolená pečeň. Všetci pacienti od ktorých boli získané snímky sú potenciálnymi darcami pečene, čo znamená že sa na nej nenachádzajú žiadne nádory, lézie alebo iné ochorenia.

CT obrazy sa získali z oblasti hornej časti brucha pacientov v portálnej venóznej fáze po injekcii kontrastnej látky. Portálna venózna fáza nastáva 70-80 sekúnd po injekcii kontrastného činidla. V tejto fáze sa parenchým pečene zväčšuje maximálne prostredníctvom prívodu krvi portálnou žilou. Portálne žily sú dobre zosilnené a pozoruje sa aj určité zlepšenie viditeľnosti pečeňových žíl. Preto je táto fáza široko používaná pre detekciu a segmentáciu pečene a ciev pred operáciou [1].

Orientácia pacienta je rovnaká pre všetky dáta. Každá sada rezov získaná od rôzneho pacienta obsahuje 16-bitové obrázky v DICOM formáte s rozlíšením 512×512 pixelov, so vzdialenosťami medzi jednotlivými rezmi 3- 3.2 mm.

Druhou využitou databázou boli snímky z magnetickej rezonančnej tomografie (MRI), ktoré obsahovali snímky od 20 pacientov z dvoch rôznych MRI sekvencií a to z T1 váženej vstupnej duálnej fázy a z T2 váženej SPIR fázy (rýchlosť nárastu longitudálnej zložky magnetizácie je charakterizovaná T1 relaxačným časom a rýchlosť úbytku transverzálnej zložky magnetizácie je daná T2 relaxačným časom). Každá z týchto fáz používa k získaniu snímok iný rádiový pulz a odlišný gradient. Táto databáza opäť neobsahuje žiadne nádory a ani iné lézie. Jedná sa o 12 bitové obrázky v DICOM formáte s rozlíšením 256×256 pixelov a so vzdialenosťou medzi rezmi, ktorá sa pohybuje v rozsahu od 5,5- 9 mm (v prieme 7,84 mm), pričom každá sada od rôznych pacientov obsahuje 26- 50 rezov (v prieme 36), čo dokopy dáva 1 064 MRI rezov [1].

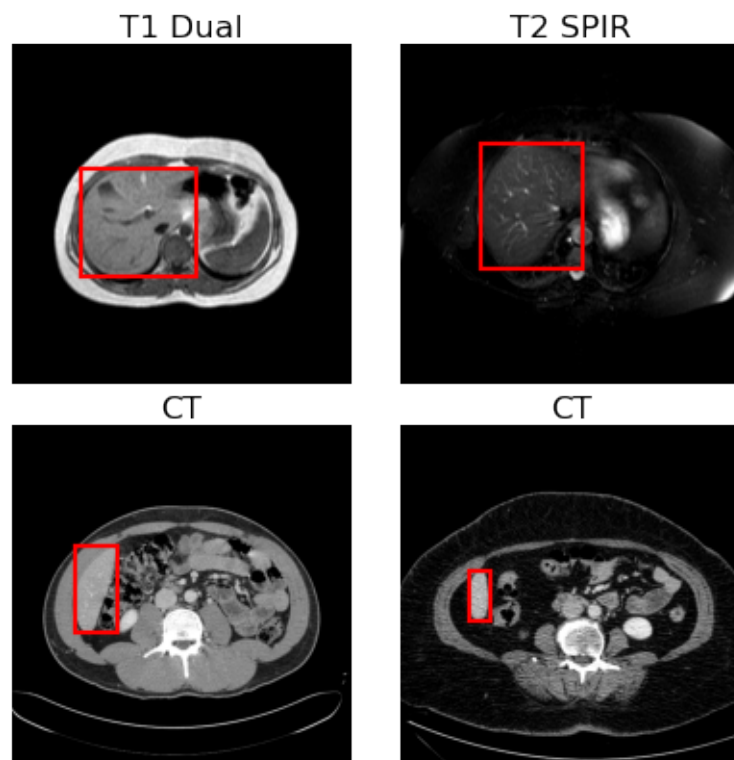
SPIR (skratka z angl. „Spectral Pre-Saturation Inversion Recovery“) je hybridná sekvencia, ktorá využíva T2 váhovaný kontrastný mechanizmus. Táto sekvencia vyžaduje citlivé nastavenie kalibrácie a homogénne magnetické pole. SPIR umožňuje selektívne potlačenie tukov, vďaka čomu je to preferovaná sekvencia k štúdiu pečene, pretože potláča tuk parenchymu pečene v obraze. Vďaka tejto výhode, sú cievy v porovnaní s inými sekvenciami viac viditeľné [1].

T1-Duálna sekvencia potláča tukové tkanivo a používa rozdiel T1 časov tukových a vodných protónov. Má dve fázy, vstupnú a výstupnú, pre detekciu boli použité len snímky zo vstupnej fázy. Táto sekvencia je veľmi užitočná pre zistenie obsahu tuku v léziách. Je vhodná pre identifikáciu tkanív bohatých na bielkoviny a tiež krvi [1].

Spolu so sadami CT a MRI snímok boli získané anotácie pre segmentáciu vo forme binárneho obrazu, ktoré boli pre detekčnú úlohu modifikované do formátu ohraničujúceho rámčeka. Každý riadok anotácie následne obsahoval:

- názov obrázka
- triedu zvolenú pre detekciu
- xmin- x-ová súradnica ľavého dolného rohu ohraničujúceho rámčeka
- ymin- y-ová súradnica ľavého dolného rohu ohraničujúceho rámčeka
- xmax- x-ová súradnica pravého horného rohu ohraničujúceho rámčeka
- ymax- y-ová súradnica pravého horného rohu ohraničujúceho rámčeka

Príklad snímok spolu s anotovanými ohraničujúcimi rámčkami je vidieť na obrázku 4.1.

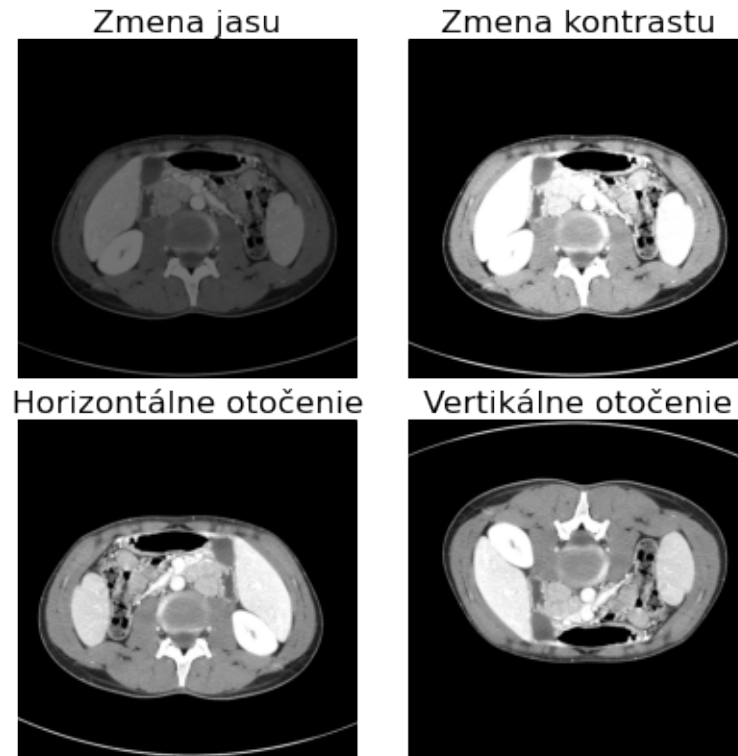


Obr. 4.1: Ukážka CT a MRI rezov s anotovanými rámčkami

Abdominálne rezy na ktorých sa nenachádzala pečeň neboli pre detekciu využité, aby nezťažovali využívanú CNN a aby tréning prebiehal rýchlejšie. Po odstránení týchto redundantných dát bolo prítomných 800 MRI snímok a 2 000 CT snímok. Pre vyrovnanie zastúpenia snímok z oboch modalít, bolo náhodne vybraných 800 CT rezov, čo spolu dáva 1 600 obrázkov pre tréning a pre validáciu.

Augmentácia sady dát prebiehala počas behu programu a bola zložená celkom zo štyroch druhov transformácií a to zmena jasnosti, zmena kontrastu, horizontálne otočenie a vertikálne otočenie. Na obrázok vstupujúci do jedného z vybraných algoritmov

bola vždy jedna z týchto transformácií aplikovaná náhodne a to s pravdepodobnosťou 0,2. V prípade geometrických transformácií boli spolu s rezom transformované aj súradnice anotovaného rámčeka. Na obrázku 4.2 je možné vidieť aplikáciu transformácií v intenzite a taktiež geometrickú augmentáciu.



Obr. 4.2: Ukážka kontrastných a geometrických transformácií v procese augmentácie

Hounsfieldova jednotka

V nasledujúcich kapitolách bude pri odvolávaní sa na CT rezy opomínaná Hounsfieldova jednotka, ktorú je vhodné pre lepšie pochopenie reprezentácie CT obrazu bližšie popísať.

CT obrazy sú reprezentované vo forme lineárneho súčiniteľa útlmu Röntgenového zväzku žiarenia. Bežne sa používa vyjadrenie v tzv. CT číslach, ktorých jednotkou je Hounsfieldova jednotka, pre porovnanie medzi systémami. Jedná sa o vzťahnutie aktuálneho lineárneho súčiniteľa útlmu k lineárnemu súčiniteľu útlmu vody. Vzorec pre výpočet CT čísla sa nachádza v rovnici 4.1.

$$CT[H.U] = k \frac{\mu_A - \mu_{voda}}{\mu_{voda}}, \quad (4.1)$$

kde μ_A je aktuálny lineárny súčiniteľ, μ_{voda} je lineárny súčiniteľ vody, $K=1\ 000$ je dohodnutá jednotka. CT číslo vody je rovné nule, vzduchu $-1\ 000$ a kosti $1\ 000$ [23].

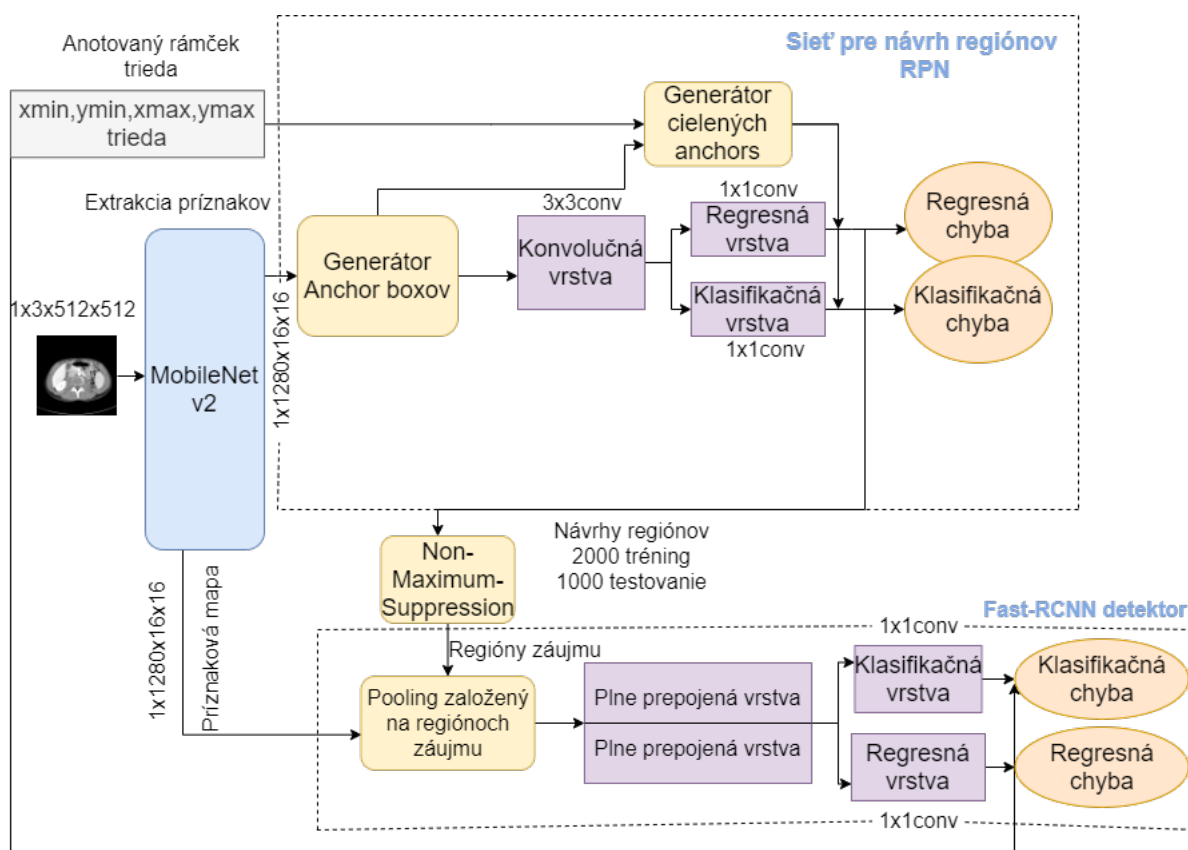
5 Implementácia algoritmov

Dva vybrané algoritmy „Faster“ R-CNN a YOLO boli implementované v programovacom jazyku Python [24]. Jeho hlavná výhoda spočíva v jeho verejnej dostupnosti, vďaka čomu je na internete prístupných množstvo knižníc a je taktiež veľmi často používaný v oblasti umelej inteligencie. Konkrétna implementácia prebiehala vo verzii Python 3.7.4.

Pre tvorbu konvolučnej neurónovej siete bola využitá knižnica Pytorch [25], ktorá bola navrhnutá, aby bola rýchla, flexibilná a ľahko použiteľná. Je vhodná pre začiatníkov, pretože kód je dobre čitateľný a dokumentácia tejto knižnice je vecne a užitočne spracovaná. Pytorch je balík s otvoreným zdrojovým kódom určeným pre strojové učenie s podporou konvolučných a rekurentných neurónových sietí, ktorý sa používa hlavne na aplikácie v oblasti počítačového videnia a spracovania prírodného jazyka. Jeho hlavnou výhodou je výpočet tenzorov so silným zrýchlením prostredníctvom grafickej procesorovej jednotky (GPU).

5.1 Realizácia „Faster“ R-CNN

Pre detekciu pečene z abdominálnych CT a MRI snímok bol využitý algoritmus „Faster“ R-CNN, ktorý sa skladá, ako bolo spomínané v kapitole číslo 3, z dvoch modulov a to zo siete pre extrakciu príznakov, siete pre návrh regiónov a „Fast“ R-CNN detektoru. Konkrétna realizácia programu pre detekciu objektov je znázornená na obrázku 5.1 vo forme blokovej schémy. „Faster“ R-CNN algoritmus bol trénovaný aj testovaný na obrázkoch jednotnej veľkosti a to 512×512 pixelov. Veľkosť obrázka vstupujúceho do modelu je udávaná vo forme $B \times C \times H \times W$, kde písmeno B symbolizuje takzvanú „batch size“, ktorá udáva počet obrázkov vstupujúcich do siete v jednej iterácii, písmeno C počet kanálov obrázku, v tomto prípade sa jedná o farebný obraz (RGB z angl. „red“, „green“, „blue“, to znamená, že každý obrázok je zložený kombináciou týchto troch farieb, pričom každá je reprezentovaná hodnotou od 0-255), H výšku a W šírku, pri tréningu aj testovaní bol rozmer obrázka vstupujúceho do siete $1 \times 3 \times 512 \times 512$.

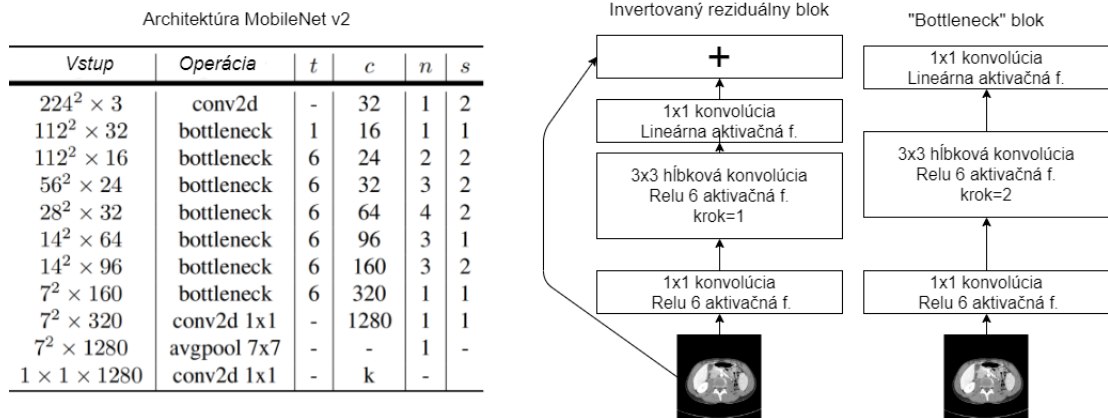


Obr. 5.1: Model „Faster“ R-CNN znázornený vo forme blokovej schémy.

5.1.1 Extrakcia príznakov

Pred tým ako obrázky vstupujú do siete pre návrh regiónov a do detekčnej siete je nutné z nich extrahovať príznaky. Pre túto úlohu bola zvolená architektúra siete MobileNet v2. Táto architektúra využíva invertovaný reziduálny blok, ktorý umožňuje preskakovanie vrstiev, avšak od bežného reziduálneho bloku sa líši tým, že spája vrstvy s menšou hĺbkou (zhládiska počtu kánalov) pričom sú preskočené vrstvy s väčšou hĺbkou. V tomto bloku sa nachádzajú tri vrstvy, ktoré tvoria tzv. „bottleneck“ reziduálneho bloku. Prvou vrstvou je kovučná vrstva s 1×1 konvulúčným filtrom. Táto vrstva využíva ReLU 6 aktivačnú funkciu. MobileNet v2 využíva hĺbkovo separabilnú konvulúciu (z angl. „depthwise separable convolution“), ktorá rozdeľuje konvulúčnú operáciu do dvoch vrstiev a to na hĺbkovú konvulúciu a bodovú konvulúciu. 3×3 hĺbková konvulúcia (z angl. „depthwise convolution“) sa od bežnej konvulúcie líši tým, že na každý vstupný kanál obrazu je aplikovaný samostatný konvulúčný filter. Využitou aktivačnou funkciou tejto vrstvy je opäť ReLU 6. 1×1 bodová konvulúcia (z angl. „pointwise convolution“) je zodpovedná za vytváranie nových príznakov prostredníctvom výpočtu lineárnej kombinácie vstupných kánalov,

čo znamená, že sa jedná o vrstvu s lineárnou aktivačnou funkciou [26]. Architektúra siete spolu s invertovanými reziduálnymi blokmi a „bottleneck“ vrstvami je zobrazená na obrázku 5.2.



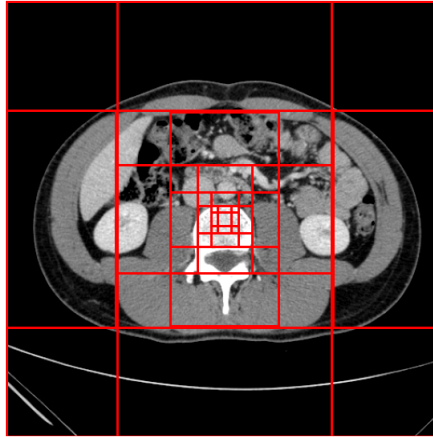
Obr. 5.2: MobileNet v2, vľavo predstavená architektúra siete, vpravo ukážka invertovaného reziduálneho bloku a „bottleneck“ bloku, (upravené z [26]).

Každý riadok architektúry na obrázku 5.2 vľavo reprezentuje sekvenciu vrstiev, ktoré sa opakujú n krát, kde c je počet výstupných kanálov, t značí pomer počtu kanálov vstupujúcich do „bottleneck“ k počtu kanálov vo vnútornej vrstve „bottleneck“ a je označovaný ako tzv. expanzný pomer. Písmeno s reprezentuje krok („stride“) s ktorým sa konvolučný filter posúva po príznakovej mape a je daný pre prvú vrstvu v sekvencii, všetky ostatné vrstvy majú s rovný jednej. Práve blok s krokom 2 sa v architektúre vyskytuje päťkrát, vďaka čomu je veľkosť príznakovej mapy vystupujúcej z tejto architektúry $1 \times 1 \times 280 \times 16 \times 16$ kde 1 280 je počet výstupných kanálov tejto siete [26].

5.1.2 Sieť pre návrh regiónov (RPN)

Ďalej boli generované „anchors“ rámčeky, ktoré indikujú možnosť výskytu objektu na danom mieste a predefinujú tvary. Tieto rámčeky boli vytvorené v 5 rôznych škálach a to 32, 64, 128, 256, 512 a v troch rôznych pomeroch 1:2, 1:1, 2:1, čo znamená, že pre každý pixel bolo vytvorených spolu 5x3 „anchors“ rámčekov. Rámčeky, ktoré prekračujú hranice mapy, sú označené indexom -1 a pri tréningu sú ignorované, pretože by významne prispievali ku chybe [18]. Príklad vytvorenia „anchors“ na jednom pixely je zobrazený na obrázku 5.3.

Tieto „anchors“ rámčeky ďalej vstupujú do siete pre návrh regiónov, ktorá je zložená z 3×3 konvolučnej vrstvy, ktorá je nasledovaná dvomi 1×1 konvolučnými vrstvami, pričom jedna slúži pre klasifikáciu objektu a druhá pre predpoveď ohraničujúceho rámčeka ako je zobrazené v schéme 5.1. Výstup zo siete pre návrh regiónov



Obr. 5.3: Znáznornenie „anchors“ rámcčkov pre „Faster“ R-CNN.

sú teda pravdepodobnosti prítomnosti vs. neprítomnosti objektu v „anchors“ rámcčku a súradnice týchto rámcčkov.

Na výstup z RPN je aplikovaná operácia „Non Maximum Suppression“, ktorá umožňuje odstrániť prekrývajúce sa prebytočné rámcčky. Po zoradení boxov zostupne podľa ich skóre výskytu objektu, sú odstránené tie, ktorých prekryv označovaný aj ako IoU je väčší ako 0,7 a zároveň je určený maximálny počet rámcčkov, ktoré budú vstupovať do detekčnej siete počas tréningu a počas testovania. Takto sú zo skupiny rámcčkov vybrané len tie s najväčším skóre. Maximálny počet boxov vstupujúcich do „Fast“ R-CNN detektoru počas tréningu bol určený na 2 000 a počas testovania na 1 000.

5.1.3 Detekčná sieť

Na vstup detekčnej siete prichádza príznaková mapa z MobileNet v2, ale aj návrhy regiónov z RPN ako je znázornené šípkami v obrázku 5.1. Návrhy regiónov sú použité na podvzorkovanie regiónov záujmu (v obrázku ako „pooling“ založený na regiónoch záujmu) z príznakovkej mapy. Výstup z podvzorkovacej vrstvy má veľkosť $N \times 7 \times 512 \times 512$, kde N je počet návrhov z RPN. Následne príznaky prechádzajú dvomi plne prepojenými vrstvami, po čom nasleduje klasifikačná („softmax“) vrstva, ktorá predpovedá N tried objektov vrátane pozadia spolu s pravdepodobnosťou výskytu a regresná vrstva predpovedajúca súradnice rámcčkov pre jednotlivé triedy, z ktorej výstup teda bude $N \times 4$. Pre tréning klasifikačnej vrstvy bolo zvolené minimálne IoU medzi návrhmi regiónov a anotovanou oblasťou na hodnotu 0,5, aby tieto návrhy boli považované za pozitívne. Zároveň je zvolená minimálna hodnota pravdepodobnosti výskytu objektu v rámcčku na 0,05, všetky rámcčky s nižšou hodnotou pravdepodobnosti sú odstránené, čo nám výrazne znižuje počet nesprávnych

detekcií.

Pri evaluácii bola na detekčný systém aplikovaná operácia „Non Maximum Suppression“, pričom boli odstránené rámčeky prekrývajúce sa nad 50 % a rámčeky so skóre klasifikačnej vrstvy menším ako 0,05. Výstup z Faster R-CNN detektoru sú súradnice ohraničujúceho rámčeka ($x_{min}, y_{min}, x_{max}, y_{max}$), číselne označené triedy (0-pozadie, 1-pečeň) a hodnota pravdepodobnosti výskytu danej triedy v rámčeku („softmax“ hodnota).

5.1.4 Výpočet chybovej funkcie

Pri tréningu siete pre návrh regiónov je priradená binárna hodnota každému z „anchors“ podľa prítomnosti alebo neprítomnosti objektu (v blokovej schéme 5.1 je blok reprezentujúci túto operáciu pomenovaný ako generátor cielených „anchors“). Pozitívnu hodnotu priradzujeme v dvoch prípadoch. „Anchors“ prekrývajúci sa s anotovanou oblasťou s najvyššou hodnotou IoU alebo „anchors“, ktorý ma hodnotu IoU väčšiu ako 0,7 s akýmkoľvek pravdivým rámčekom. Jeden pravdivý box môže priradiť pravdivú hodnotu niekoľkým „anchors“. Zvyčajne druhá podmienka determinuje pozitívnu hodnotu, avšak používa sa aj prvá, pretože môžu nastať prípady, kedy IoU nie je vyššie ako 0,7. Negatívnu hodnotu priradzujeme „anchors“, ktorých IoU s anotovaným rámčekom je menšie ako 0,3 [18].

„Faster“ R-CNN počíta celkovo 4 chybové funkcie:

- Chybová funkcia pre regresiu rámčeka v RPN
- Chybová funkcia pre klasifikáciu do dvoch tried v RPN
- Chybová funkcia pre regresiu rámčeka vo „Fast“ R-CNN detektore
- Chybová funkcia pre klasifikáciu do N tried vo „Fast“ R-CNN detektore

Chybová funkcia pre pre regresiu ohraničujúceho rámčeka označovaná aj „Smooth L1 Loss“ je definovaná ako:

$$L(\{x_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*), \quad (5.1)$$

kde i je index „anchor“, p_i je predpokladaná pravdepodobnosť daného „anchor“, p_i^* je označenie pravdivého rámčeka, ktoré je 1 ak je „anchor“ pozitívny a 0 ak je negatívny, t_i je vektor predvídajúci 4 parametrizované súradnice predvídaného rámčeka, t_i^* je pravdivý box asociovaný s pozitívnym „anchor“, L_{log} je logaritmus chybovej funkcie L vypočítaný pre dve triedy: prítomnosť vs. neprítomnosť objektu, L_{reg} je regresná chyba vypočítaná ako: $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$. N_{cls} je veľkosť množiny N-tried, N_{reg} je počet anchors a λ je balancujúci faktor [18].

Pre regresiu ohraničujúcich rámčekov sú používané 4 parametrizované súradnice a to:

$$t_x = (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \quad (5.2)$$

$$t_w = \log(w/w_a) \quad t_h = \log(h/h_a), \quad (5.3)$$

$$t_x^* = (x^* - x_a)/w_a \quad t_y^* = (y^* - y_a)/h_a, \quad (5.4)$$

$$t_w^* = \log(w^*/w_a) \quad t_h^* = \log(h^*/h_a), \quad (5.5)$$

kde x, y sú súradnice určujúce stred rámčeka, w, h sú súradnice určujúce výšku a šírku rámčeka a x, x_a, x^* (tiež platí pre y, w, h) sú súradnice pre ohraničujúci rámček, „anchor“ rámček a anotovaný rámček.

Pre hodnotenie presnosti klasifikácie v sieti pre návrh regiónov sa používa binárna krížová entropia nazývaná aj ako logaritmickej chyba, pretože sa jedna o binárnu klasifikáciu výskytu objektu v rámčeku. Jej výpočet je daný ako:

$$J(w) = -\frac{1}{N} \sum_{n=1}^N [y_n \log(\hat{y}_n) + (1 - y_n) \log(1 - \hat{y}_n)], \quad (5.6)$$

kde N je počet tried y_n je pravdivá trieda a \hat{y}_n predikovaná pravdepodobnosť výskytu danej triedy.

Keďže v detekčnej sieti modelu sa vo väčšine prípadov jedná o klasifikáciu do viacerých skupín, využívaná chybová funkcia je krížová entropia pre viacero tried definovaná ako:

$$J = -\frac{1}{N} \left(\sum_{n=1}^N y_n \log(\hat{y}_n) \right), \quad (5.7)$$

kde y_n je pravdivé označenie objektu a \hat{y}_n predikovaná pravdepodobnosť výskytu danej triedy v rámčeku [18, 27].

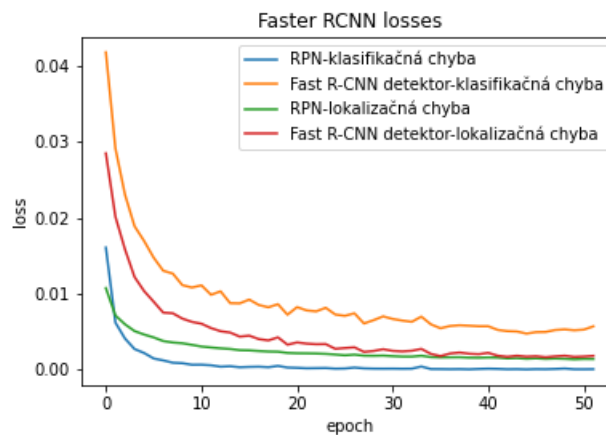
5.1.5 Tréning „Faster“ R-CNN

Na vytvorenej učebnej sade dat popísanej v časti 4 bol trénovaný „Faster“ R-CNN algoritmus. Na začiatku boli nastavené váhy modelu ako náhodné hodnoty pomocou normálneho rozloženia so smerodatnou odchýlkou 0,01. Pre optimalizáciu parametrov modelu bol využitý „Adam“, ktorý je rozšírením stochastického gradientného zostupu s krokom učenia nastaveným na hodnotu 0,001. Model bol trénovaný na 50 epochách, pričom po každej epoche boli nové váhy uložené.

Počas tréningu boli počítané 2 chybové funkcie pre oba moduly, ktoré tvoria „Faster“ R-CNN detekčný systém. Ako miera rozdielu medzi „anchors“, ktoré obsahujú anotované rámčeky a „anchors“ predikovanými sieťou pre návrh regiónov bola využitá takzvaná „Smooth L1“ chybová funkcia. Je možné vypočítať túto chybovú funkciu pre všetky „anchors“, ale výsledok by bol posunutý k negatívnym „anchors“, pretože prevažujú. Aby mohla byť vypočítaná chybová funkcia bolo náhodne navzorkovaných 256 „anchors“ v obrázku, kde pomer pozitívnych k negatívnym „anchors“

je najviac 1:1. Ak sa v tejto množine vzorkov nachádzalo menej ako 128 pozitívnych „anchors“ rámečkov, tak bola vyplnená negatívnymi „anchors“ rámečkmi podľa vzoru [18].

Na výstup z klasifikačnej vrstvy RPN, ktorý predstavuje binárnu hodnotu pravdepodobnosti výskytu objektu v danom „anchor“ rámečku bola aplikovaná binárna krížová entropia. „Smooth L1“ chybová funkcia bola aplikovaná aj na porovnanie výstupu z regresnej vrstvy „Fast“ R-CNN detektoru so súradnicami anotovaných rámečkov. Pre klasifikačnú úlohu detekčnej siete bol znovu využitý výpočet krížovej entropie. Chybové funkcie tréningového a validačného datasetu sú počas učenia sledované a je využitá funkcia automatického predčasného zastavenia učenia ak chyba vo validačnom datasete vzrastie alebo stagnuje v 3 epochách po sebe. Priebeh chybových funkcií je zobrazený na obrázku 5.4



Obr. 5.4: Ukážka priebehu chybových funkcií „Faster“ R-CNN, kde sú vyznačené klasifikačné a regresné chybové funkcie RPN a „Fast“ R-CNN detektoru.

Sieť bola trénovaná pomocou 4- krokového striedavého tréningu spomínaného v časti 3.3.2. Najprv bola trénovaná sieť pre návrh regiónov, neskôr detekčná sieť a až v ďalšom kroku medzi sebou začali zdieľať váhy.

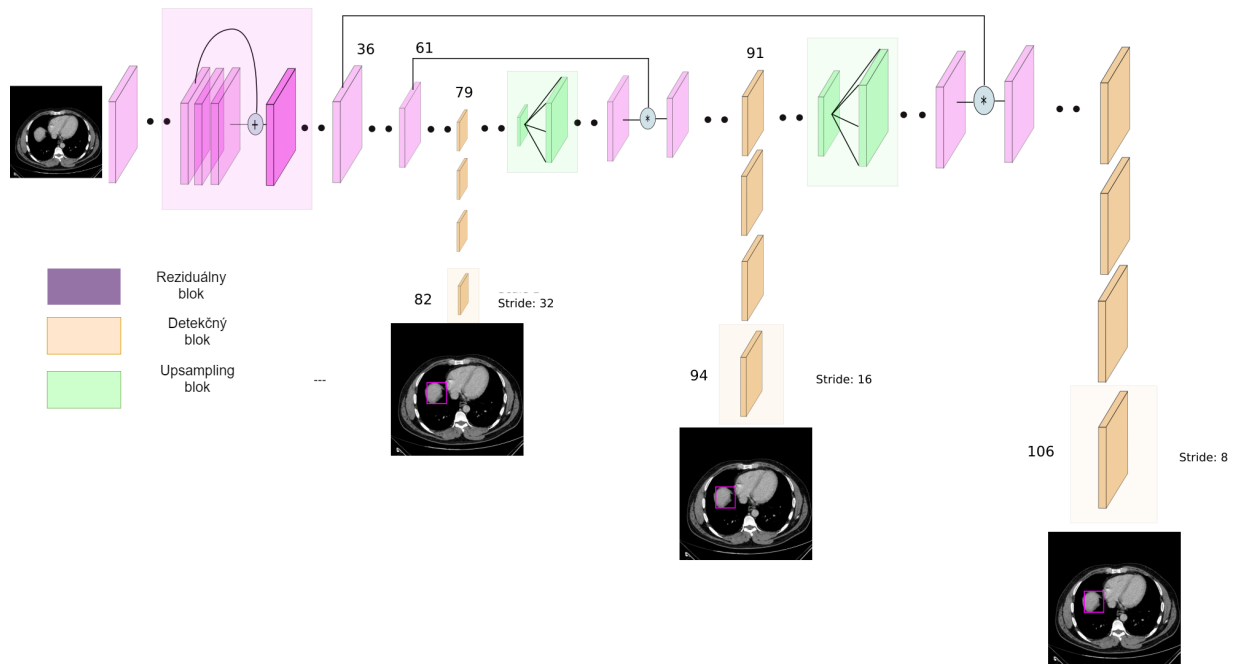
5.2 Realizácia YOLO v3

Druhý algoritmus využitý pre detekciu pečene z abdominálnych CT a MRI snímok bol YOLO, konkrétne jeho tretia verzia. YOLO v3 používa variantu Darknet- 53, ktorá má originálne 53 vrstiev tréovaných na Imagenet [28], avšak pre detekciu bolo k týmto vrstvám pripojených ďalších 53 vrstiev, čo vytvára 106 vrstvovú architektúru siete YOLO v3. V tejto najnovšej architektúre YOLO využívanej pre detekciu je umožnené preskakovanie niektorých spojení vrstiev za čo sú zodpovedné reziduálne bloky a taktiež tzv. „upsampling“ blok, ktorý umožňuje zväčšenie príznakovej mapy. Sieť pre extrakciu príznakov Darknet-53 je možné vidieť na obrázku 5.5.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	128 × 128
	Convolutional	64	3 × 3	
	Residual			
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	64 × 64
	Convolutional	128	3 × 3	
	Residual			
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	32 × 32
	Convolutional	256	3 × 3	
	Residual			
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	16 × 16
	Convolutional	512	3 × 3	
	Residual			
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	8 × 8
	Convolutional	1024	3 × 3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

Obr. 5.5: Ukážka architektúry DarkNet-53, kde „Type“ reprezentuje druh vrstvy, „Filters“ počet filtrov, „Size“ veľkosť filtra v konvolučných vrstvách a „Output“ veľkosť výstupnej príznakovej mapy, (prevzaté z [29]).

Všetky obrázky boli pred vstupom do siete transformované na veľkosť 416×416 pixelov. Rozmer obrázka, ktorý do algoritmu vstupoval mal opäť štyri dimenzie o veľkosti $1 \times 3 \times 416 \times 416$. Model YOLO počíta so súradnicami anotovaných ohraničujúcich rámcikov, ktoré sú normalizované výškou a šírkou obrázka. V tomto prípade boli súradnice prepočítané na x-ovú a y-ovú súradnicu stredu rámcika, šírku rámcika a výšku rámcika, pričom každá z týchto súradníc sa nachádza v intervale (0,1). Bloky architektúry YOLO v3 sú zobrazené na obrázku 5.6.



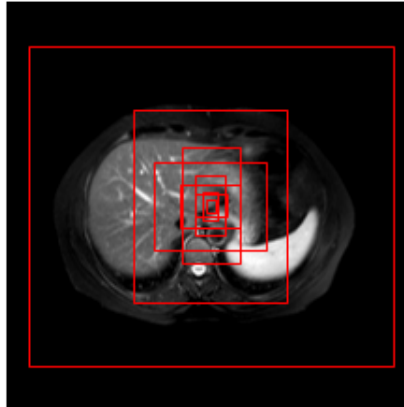
Obr. 5.6: Architektúra YOLO v3. V obrázku sú vyznačené tri druhy blokov: Rezi-duálny, detekčný (tzv. YOLO blok) a „upsampling“ blok. Taktiež je zvýraznená detekcia v troch vrstvách (82, 94, 106), (upravené z [30]).

5.2.1 Detekcia v troch vrstvách

Detekcia prebieha v troch rôznych vrstvách siete, použitím detekčného jadra („kernelu“) veľkosti 1×1 , na príznakové mapy 3 rôznych veľkostí. Veľkosť detekčného jadra je daná ako $1 \times 1 \times (B \times (5 + C))$, kde B je počet ohraničujúcich rámečkov, ktoré bunka v príznakovej mape dokáže predikovať, číslo 5 určuje počet súradníc ohraničujúceho rámečka plus skóre spoľahlivosti, a C označuje počet tried. V tomto prípade je veľkosť detekčného jadra rovná $1 \times 1 \times 18$. Prvá detekcia prebieha v 82. vrstve, kde sa jadro posúva po príznakovej mape s krokom 32, čo znamená že ak je vstupný obrázok veľkosti 416×416 , tak výstupná príznaková mapa z tejto vrstvy bude mať v prípade troch ohraničujúcich rámečkov a jedného detekovaného objektu veľkosť $13 \times 13 \times 18$. Ďalšia detekcia prebieha v 94. vrstve, kde sa jadro posúva po obrázku s krokom 16, z ktorej výstup je príznaková mapa o veľkosti $26 \times 26 \times 18$ a posledná finálna detekcia prebieha v poslednej 106. vrstve s krokom posúvania jadra 8, čím sa získa príznaková mapa o veľkosti $52 \times 52 \times 18$ ako je zobrazené v schéme číslo 5.6. Prvá detekčná vrstva je zodpovedná za detekciu veľkých objektov, druhá vrstva za detekciu objektov strednej veľkosti a tretia za detekciu malých objektov [29, 30].

YOLO v3 používa celkovo 9 „anchors“ rámečkov, ktoré boli rovnomerne rozložené pre každú škálu, zoradené v zostupnom poradí. Tieto anchors boli vytvorené

zhlukovacou metódou k-priemerov (z angl. „k-means“) na COCO datasete [34] na základe výšky a šírky anotovaných oblastí a boli pridané na každý pixel obrazu. Tieto rámčeky dobre všeobecne predefinujú rozmery objektov. Výšky a šírky „anchors“ sú dané na tieto hodnotky: ((373,326), (156,198), (116,90), (59,119), (62,45), (30,61), (32,23), (16,30), (10,13)). Ich rozloženie na jednom pixely je zobrazené na obrázku č. 5.7 [29].



Obr. 5.7: Ukážka anchors rámčekov pre YOLO

Pri evaluácii je na výstup z modelu YOLO aplikovaná operácia „Non Maximum Suppression“, vďaka čomu sú odstránené všetky predikcie so skóre nižším ako 0,05 a zároveň rámčeky prekrývajúce sa na viac ako 50 %. Výstup z funkcie „Non Maximum Suppression“ je potom v tvare súradníc boxu (x-min,y-min,x-max,y-max), skóre spoľahlivosti, pravdepodobnosť danej triedy a predikovaná trieda.

5.2.2 Chybová funkcia a tréning YOLO v3

V architektúre YOLO v3 sú predvídané pravdepodobnosti tried a súradnice rámčekov na troch škálach. V každej tejto škále je zároveň vypočítaná aj chybová funkcia, ktorá je následne sčítaná, aby mohli byť pri tréningu optimalizované jednotlivé vrstvy. Šírka a výška rámčeku sú normalizované šírkou a výškou obrázku, takže klesnú do intervalu 0 a 1. Súradnice rámčeku x a y sú parametrizované tak, aby kompenzovali pozíciu danej bunky mriežky, takže taktiež spadnú do intervalu 0 a 1.

YOLO v3 celkovo počíta tri chybové funkcie a to:

- chybová funkcia lokalizácie rámčeku- stredná kvadratická odchýlka
- chybová funkcia skóre výskytu objektu- krížová entropia
- chybová funkcia pre klasifikáciu objektu- krížová entropia

Na optimalizáciu lokalizácie rámčekov sa používa stredná kvadratická odchýlka (MSE). MSE rovnako váhuje chybu v malých a veľkých rámčekoch. Malé odchýlky

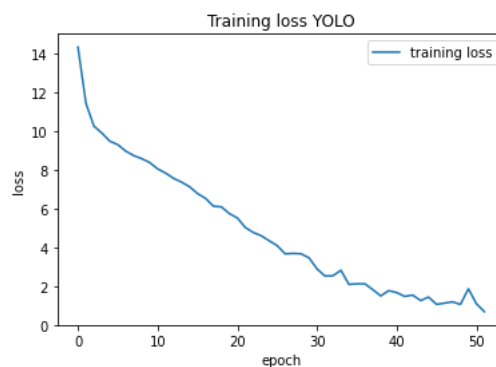
vo veľkých rámčekoch nie sú také závažné ako malé odchýlky v malých rámčekoch. Aby sa to čiastočne vyriešilo predpovedá sa druhá odmocnina výšky a šírky ohraničujúceho rámčeka namiesto toho, aby sa predpovedala priamo výška a šírka [29].

YOLO predvída niekoľko ohraničujúcich boxov na bunku, my však chceme, aby bol za každý objekt zodpovedný len jeden rámček. Len ten rámček predvídajúci objekt bude pri tréningu použitý ako výsledný, ktorý ma najväčší prienik s anotovanou oblasťou.

Narozdiel od prvej verzie YOLO je v jeho tretej verzii výpočet chyby skóre výskytu objektu a predikcie triedy realizovaný pomocou krížovej entropie. V každom obrázku sa nachádza mnoho buniek, ktoré neobsahujú žiaden objekt. Skóre výskytu objektu týchto buniek sa potom blíži nule. Tieto bunky často prevyšujú gradient buniek, ktoré obsahujú objekty, čo vedie k nestabilite modelu a divergovaniu od počiatku tréningu.

Pre vyriešenie tohto problému bola počítaná chyba skóre výskytu objektu len z boxov s najvyšším prekryvom s pravdivým rámčekom a zároveň tých, ktorých prekryv bol väčší ako 0,5. Ak je u nejakého rámčeka prekryv vyšší ako 0,5, ale nie je to rámček s najvyšším prekryvom, tak táto predikcia sa nebude počítat ku chybe [29].

Pri tréningu architektúry YOLO v3 bol využitý optimalizačný algoritmus „Adam“ s krokom učenia 0,001 a výsledný súčet chýb z troch škál bol propagovaný späť, aby boli aktualizované váhy siete. Opäť bola sledovaná chyba validačného datasetu, rovnako ako pri „Faster“ R-CNN, aby nedošlo k preučeniu modelu. Priebeh súčtu chybových funkcií je možné vidieť na obrázku č. 5.8.



Obr. 5.8: Priebeh YOLO chybovej funkcie učebnej sady dát daný ako súčet chýb z 3 škál.

6 Hodnotenie a výsledky detekcie objektov

Pre overenie schopnosti vybraných algoritmov detekovať pečeň z CT a MRI snímok bola na model aplikovaná opakovaná validácia náhodného výberu (z angl. „Repeated random subsampling validation“), kde z 1 600 dostupných snímok bolo desaťkrát náhodne vybraných 300 obrázkov pre tréning a 150 pre validáciu, pričom nedošlo k prekryvaniu snímok z trénovacej a validačnej sady dát [31].

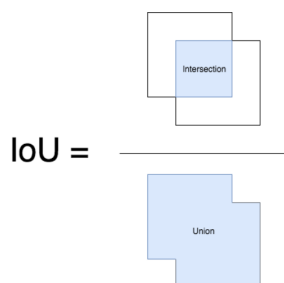
Každý z modelov bol trénovaný na 50 epochách, avšak bolo využité automatické predčasné zastavenie tréningu pri stagnácii alebo zvyšovaní chyby validačnej sady dát po viac ako 3 epochách, aby nedošlo k preučeniu modelu, čo by znamenalo, že model sa až príliš dobre adaptoval na učebné snímky a mal by slabú generalizačnú schopnosť. Jeho výsledky na nových dátach vo fáze evaluácie by neboli uspokojivé. Pre následné metriky validácie detekcie objektov, boli využité posledné získané váhy.

6.1 Metriky hodnotenia detekcie

Metriky detekcie objektov nie sú vždy jednoduché, pretože okrem klasifikácie objektov musia taktiež zahŕňať aj ich lokalizáciu v obrázku. Okrem toho sa v datasete zvyčajne nachádza viacero tried, ktorých rozloženie nie je rovnomerné. Taktiež sa musí posúdiť riziko nesprávnych klasifikácií a s tým určiť „confidence score“ pre každý nájdený objekt.

Pre hodnotenie lokalizácie daného objektu bola využitá metrika nazývaná „Intersection over Union (IoU)“ alebo spoločný prienik medzi pravdivým (vo vzorci *truth*) a predvídaným ohraničujúcim rámčekom (vo vzorci *pred*), niekedy označovaný aj ako Jaccardov index. IoU skóre nadobúda hodnoty od 0 do 1 a čím väčšia je podobnosť medzi dvomi spomínanými boxmi, tým väčšie číslo nadobúda.

$$IOU_{pred}^{truth} = \frac{truth \cap pred}{truth \cup pred} \quad (6.1)$$



Obr. 6.1: Ukážka vzorca Intersection over Union graficky, (prevzaté z [32].)

Po vypočítaní IoU sa pre každú detekciu stanoví prah, kde hodnoty IoU nad týmto prahom sa považujú za pozitívne predpovede a hodnoty nižšie sa považujú za nesprávne predpovede. Presnejšie povedané, predpovede sú klasifikované do troch skupín a to pravdivo pozitívny (TP), falošne pozitívny (FP) a falošne negatívny (FN). Pravdivo negatívne detekcie (TN) sa zväčša neuvádzajú, pretože v obrázku by sa nachádzalo veľké množstvo takýchto miest.

Jednou z využívaných metrík pre hodnotenie klasifikácie a rovnako aj lokalizácie je presnosť označovaná aj ako pozitívna prediktívna hodnota (PPV), ktorá hodnotí ako dobre si zodpovedajú pravdivé rámčeky s predikovanými a senzitivita, ktorá meria pravdepodobnosť správnej detekcie.

$$presnosť = \frac{TP}{TP + FP} \quad (6.2)$$

$$senzitivita = \frac{TP}{TP + FN} \quad (6.3)$$

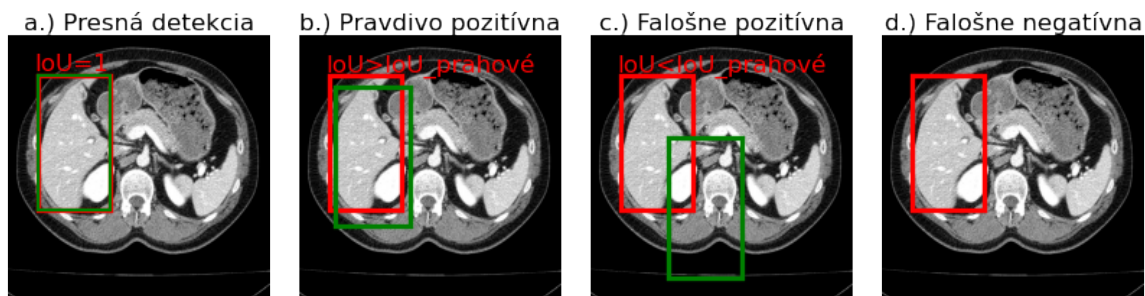
Vysoká senzitivita a nízka pozitívna prediktívna hodnota znamená, že všetky objekty boli detekované, ale väčšina z detekcií bola nesprávna. Naopak vysoká PPV a nízka senzitivita značí, že objekty sú lokalizované správne, ale väčšina objektov nebola detekovaná [33].

Skóre spoľahlivosti („confidence score“) sa určuje u každého rámčeka a hodnotí, aká je pravdepodobnosť výskytu objektu a zároveň ako dobre je daný objekt ohraničený. Za hodnotu skóre spoľahlivosti je zodpovedná klasifikačná vrstva modelu. Jeho prahová hodnota je laditeľný parameter, ktorým sa dá ovplyvniť presnosť a senzitivita a tým vylepšiť celkový výkon modelu. Zmenou prahového skóre spoľahlivosti alebo prahového IoU môžeme sledovať zmeny presnosti a senzitivity [32, 33].

Ako metrika hodnotenia presnosti detekcie objektu bola zvolená krivka závislosti presnosti na prahovej hodnote IoU. Bol zohľadnený výsledok klasifikácie a zároveň lokalizácie objektu.

Do vyššie spomínanej TP skupiny bol zaradený rámček, ktorého hodnota IoU bola väčšia ako aktuálna prahová hodnota a zároveň u ňo bola správne určená trieda. Do FP kategórie bol zaradený objekt, ktorého IoU bolo menšie ako prah alebo bol klasifikovaný ako pozadie.

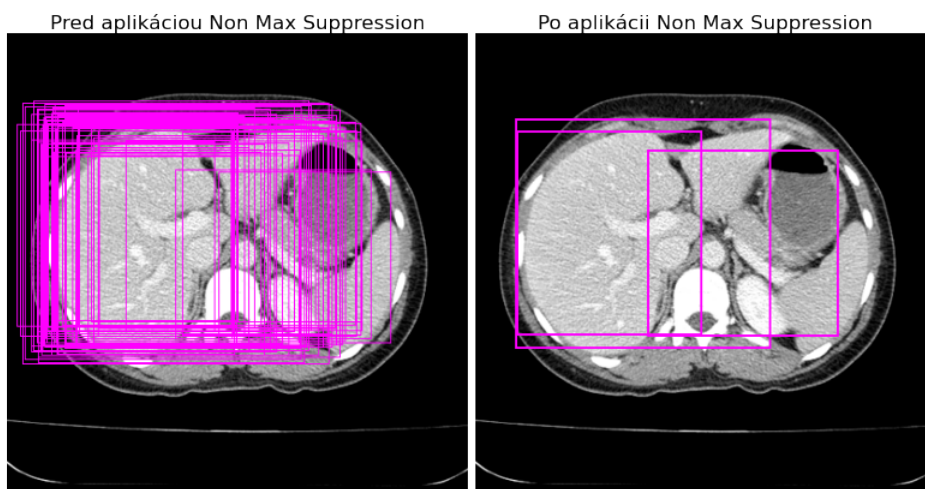
Do FN detekcií zaradzujeme tie výsledky, kedy prítomný objekt nebol nájdený alebo bol klasifikovaný ako pozadie. Tu je možné pozorovať rozdiel medzi YOLO a „Faster“ R-CNN modelom. YOLO algoritmus nie je schopný detekovať pozadie. „Faster“ R-CNN na druhej strane dokáže klasifikovať pozadie, čoho dôsledkom je, že pri evaluácii s natrénovaným modelom nemusí detekovať žiadny rámček. Na obrázku číslo 6.2 sú znázornené rôzne typy detekcií, kde červenou farbou je znázornený anotovaný rámček a zelená farba označuje predikovaný rámček.



Obr. 6.2: Ukážky zaradenia detekcií do tried. V obrázku a.) je zobrazená presná detekcia, kedy je pravdivý rámček identický s predikovaným. V príklade b.) je zobrazená pravdivo pozitívna detekcia, kedy je IoU väčšie ako zvolený prah (napr. 50 %). Falošne pozitívna detekcia je na snímke c.) kde je IoU menšie ako stanovený prah a v snímku d.) je falošne negatívna detekcia, pretože rámček nebol predikovaný.

6.2 Porovnanie dosiahnutých výsledkov

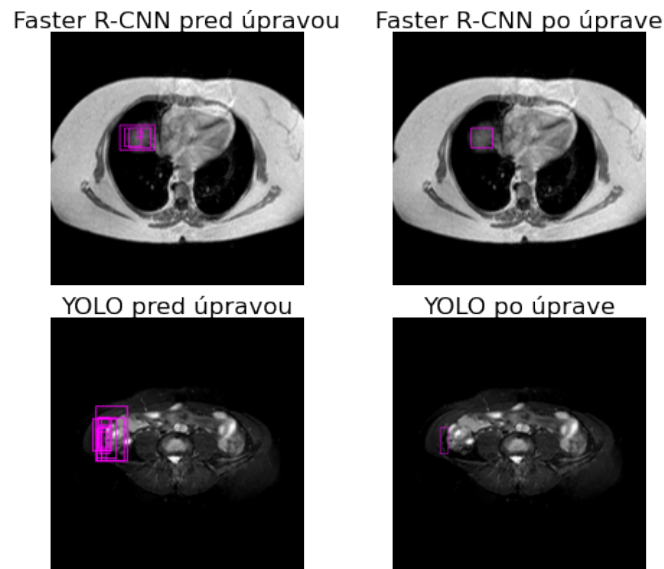
Oba aplikované modely detekujú väčší počet oblastí pre jeden prítomný objekt. Redukcia počtu boxov bola zabezpečená operáciou „Non Maximum Suppression“, kde bol zvolený prah „confidence score“, ktoré je dané klasifikačnou vrstvou na 0,05 a boli odstránené všetky oblasti prekrývajúce sa na viac ako 50 % pričom boli zachované detekcie s vyšším skóre.



Obr. 6.3: Ukážka aplikácie funkcie „Non Maximum Suppression“ na obrázok.

V niektorých prípadoch je však počet predikovaných oblastí pre jeden prítomný objekt stále väčší ako 1. V praxi je však potrebný len jeden box, ktorý by vymedzil hľadanú oblasť v obraze. Väčší počet rámečkov detekujúcich jeden objekt významne

znižuje presnosť metódy. Ako je vidieť na obrázku 6.3 vpravo, zobrazené sú 3 boxy, pričom len jeden z nich vymedzuje pečeň presne, ďalší je malý a je mimo oblasti záujmu a tretí rámček je príliš veľký. Pri validácii bol preto využitý prístup výberu jedného ohraničujúceho boxu na základe najvyššej hodnoty pravdepodobnosti výskytu triedy v rámčeku, teda najvyššej hodnoty výstupu z klasifikačnej vrstvy. Na obrázku č. 6.4 sú zobrazené príklady detekcie z oboch modelov pred a po filtrácii detekcií podľa najvyššej hodnoty ich skóre.



Obr. 6.4: Ukážka výberu rámčeku na základe „confidence score“.

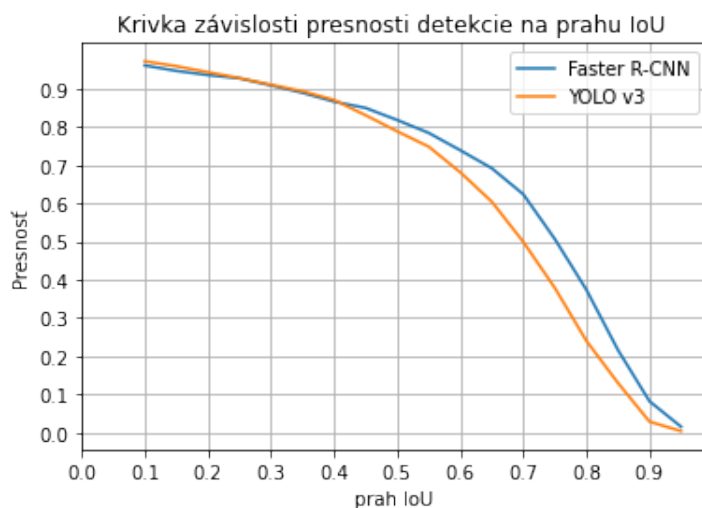
Po redukcii počtu detekcií na jeden objekt, boli vypočítané presnosti na 18 úrovniach prahu prekryvu v intervale od 0,1 do 0,95 a získané hodnoty boli vynesené do krivky presnosti pre oba algoritmy, tak ako bolo popísané v podkapitole číslo 6.1.

V tabulke číslo 6.1 sú zobrazené hodnoty presnosti na minimálnom prekryve rovnom 50 % pre algoritmus „Faster“ R-CNN, YOLO a pre 2 využité modalita ako priemery zo všetkých opakovaných validácií náhodného výberu, kde stĺpec *CT + MRI* predstavuje presnosť celého modelu, keďže bol trénovaný a validovaný na oboch modalitách zároveň.

Tab. 6.1: Priemerné presnosti detekcií z validácií náhodného výberu pri IoU 50 % .

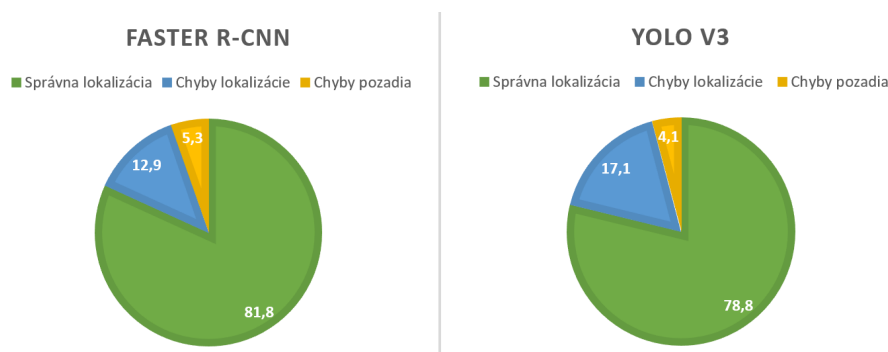
algoritmy	Faster R-CNN			YOLO v3		
	CT	MRI	CT+MRI	CT	MRI	CT+MRI
presnosť[%]	84,8	78,8	81,8	84,7	73,0	78,9

V krivke predstavenej na obrázku 6.5 sú hodnoty presnosti uvádzané ako priemery z opakovaných validácií náhodného výberu.



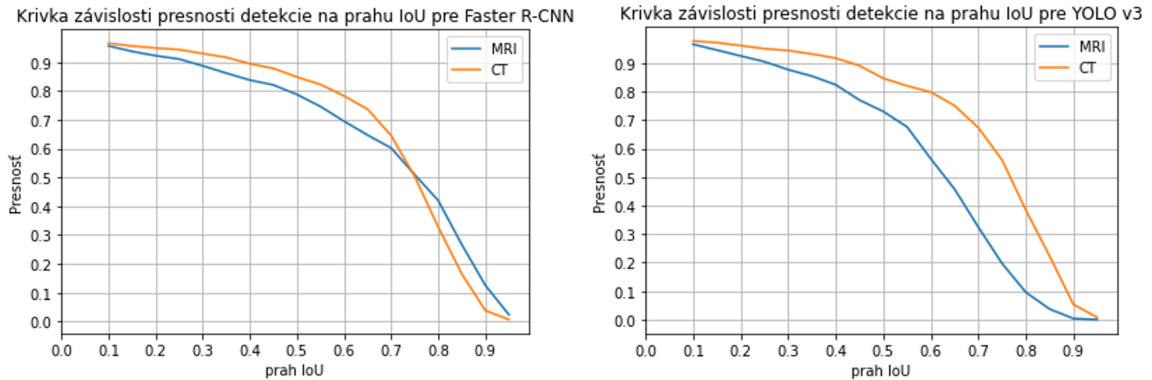
Obr. 6.5: Krivka závislosti presnosti detekcie na prahu IoU.

Z krivky je patrné, že presnosť „Faster“ R-CNN algoritmu je na väčšine úrovni IoU väčšia ako presnosť algoritmu YOLO. Na minimálnej hodnote prekryvu 50 % je presnosť celého (CT + MRI) „Faster“ R-CNN modelu 81,8 % a YOLO v3 modelu 78,9 %. Zároveň je vidieť, že do veľkosti prekryvu približne 30 %, presnosť metódy YOLO v3 prevyšuje presnosť „Faster“ R-CNN. Napriek tomu, že „Faster“ R-CNN algoritmus je presnejší, vykonáva viac detekcií pozadia ako YOLO. Ako detekcia pozadia boli určené výsledky s $IoU < 0,2$ a zároveň so správnou triedou. YOLO v3 má síce problémy so správnou lokalizáciou objektu, ale vrámci detekcie pozadia má menšiu chybovosť, čo je zobrazené aj na koláčovom grafe číslo 6.6. Je to spôsobené tým, že YOLO vníma obrázok ako celok, čo narozdiel od „Faster“ R-CNN, ktorý využíva posuvné okno umožňuje vnímať triedy a ich vzhľad v širšom kontexte.



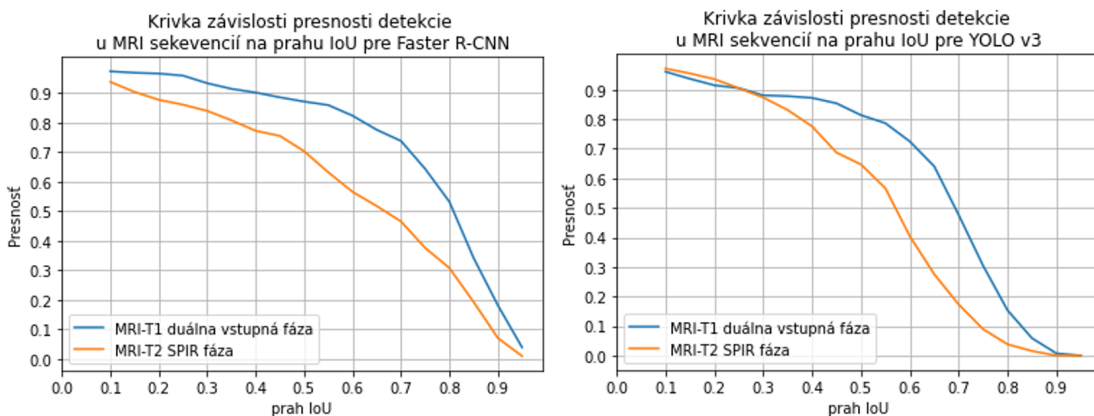
Obr. 6.6: Koláčové grafy ukazujúce percentuálne zastúpenie 3 typov detekcií pri N-objektoch. Za správne detekcie sa považujú lokalizácie s IoU väčším ako 0,5 a správnou triedou, za chyby lokalizácie sa považujú detekcie s prekryvmi v intervale $<0,2;0,5>$ a do chybných detekcií pozadia sú zaradené oblasti s IoU menším ako 0,2.

Následne bola porovnaná schopnosť predikovať oblasti s pečňou na CT a MRI snímkoch, kde hodnoty závislosti ich presnosti na minimálnom prahu IoU 0,5 sú zobrazené v tabuľke 6.1. U „Faster“ R-CNN je presnosť detekcie na CT skenoch 84,8 % a na MRI skenoch je to 78,8 %. U Modelu YOLO v3 je presnosť na CT snímkoch o niečo menšia a je to 84,7 % a na MRI snímkoch presnosť výrazne poklesla na hodnotu 73 %. Závislosť presnosti na prahu IoU u oboch modelov na 2 rôznych modalitách je predstavená na obrázku číslo 6.7.



Obr. 6.7: Grafy závislosti presností na prahu IoU pre CT a MRI modalitu a oba algoritmy.

Pri porovnávaní výsledkoch na CT a MRI snímkoch bolo zistené, že oba modely dosahujú lepšie výsledky na dátach z výpočetnej tomografie. Pri bližšom preskúmaní dát z magnetickej rezonančnej tomografie, boli patrné významné rozdiely medzi výsledkami na T1 váhovanej duálnej vstupnej fáze a na T2 váhovanej SPIR fáze, ktoré sú pre lepšiu vizualizáciu zobrazené v grafoch 6.8.



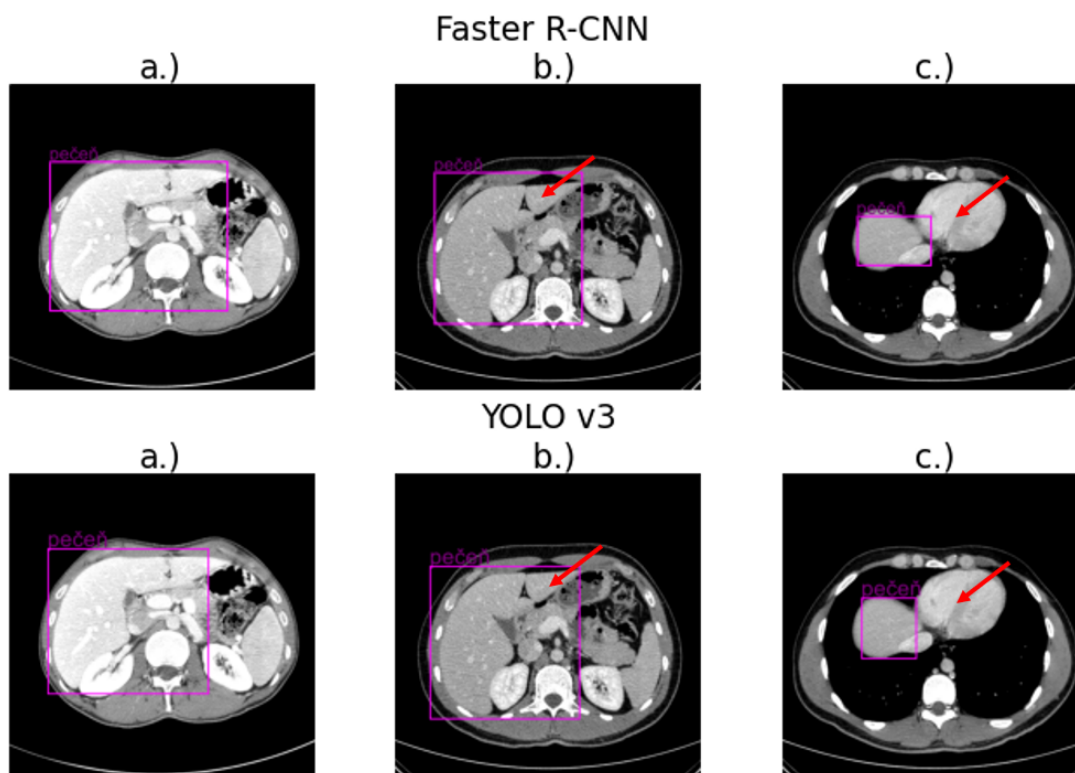
Obr. 6.8: Grafy závislosti presností na prahu IoU pre MRI sekvencie a oba algoritmy.

Presnosť na väčšine prekryvoch je u oboch metód vyššia pri T1 duálnej vstupnej fáze. Tieto výsledky môžu byť spôsobené tým, že aj napriek vysokému kontrastnému rozlíšeniu, rozloženie farieb v T2 váhovanej SPIR fáze sa veľmi líši od duálnej fázy a od CT. Rezy zo SPIR fázy sú narozdiel od ďalších dvoch druhov snímok tmavšie, čo bolo potvrdené aj zistením priemernej farby vyskytujúcej sa v pixeloch týchto snímok. Taktiež vrámci tréningových dát sú rezy z T2 SPIR váhovanej fázy o 5 % menej zastúpené ako rezy z T1 duálnej vstupnej fázy. Presnosť „Faster“ R-CNN na rezoch z duálnej T1 váhovanej vstupnej fázy pri minimálnom prahu prekryvu 50 % je 87.2 % a na rezoch z T2 váhovanej SPIR fázy je presnosť 70.4 %. Ak by sme v tomto prípade brali do úvahy len snímky z duálnej fázy, potvrdil by sa predpoklad, že detekcia na MRI rezoch bude dosahovať lepšie výsledky ako detekcia na CT rezoch, vďaka ich lepšiemu kontrastnému rozlíšeniu mäkkých tkanív. V prípade YOLO v3 je presnosť u duálnej fázy 81,4 % a u SPIR fázy 65 %.

Zhľadiska porovnania rýchlosti 2 spomínaných modelov, YOLO v3 zďaleka prevažuje rýchlosť „Faster“ R-CNN. Pri evaluácii bola meraná doba trvania jednej iterácie pri behu programu na centrálnej procesorovej jednotke (CPU) s procesorom Intel Core i5-1035G1, s pracovnou frekvenciou 1 GHz a s pamäťou RAM 8GB, pričom počas jednej iterácie do siete vstupoval jeden snímok, čo bolo sledované na N-obrázokoch. Priemerná doba trvania jednej iterácie u YOLO v3 bola 1,82 sekundy a u Faster R-CNN 6,45 sekundy. YOLO v3 je vďaka svojej rýchlosti vhodné aj na detekciu v reálnom čase.

6.3 Diskusia dosiahnutých výsledkov

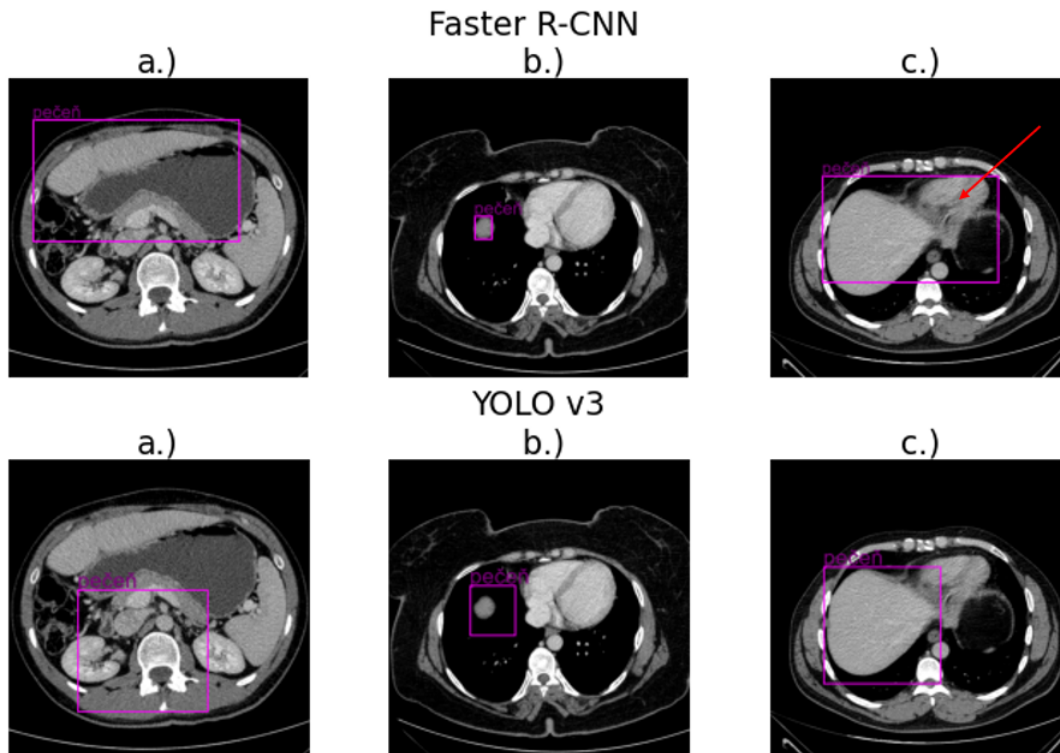
Ako už bolo spomínané, oba algoritmy dosahovali najlepšie výsledky na abdominálnych CT rezoch, príklady takýchto detekcií sú uvedené na nasledujúcich obrázkoch. Vo všetkých obrázkoch vyskytujúcich sa v tejto podkapitole sú uvedené výsledky na rovnakých rezoch z oboch algoritmov, aby bolo možné pozorovať prítomné rozdiely.



Obr. 6.9: Príklady detekcií na CT rezoch s vysokým IoU s anotovanou oblasťou. Prvý riadok predstavuje výsledky „Faster“ R-CNN modelu. Druhý riadok predstavuje výsledky YOLO v3 modelu.

Ako možné vidieť na obrázku 6.9, tvary a polohy pečene sa u rôznych pacientov a typov rezov značne odlišujú. Pre porovnanie sú zobrazené výsledky na rovnakých rezoch z oboch algoritmov. Na snímkoch označených písmenom a.) sa dá zrakom posúdiť odlišnosť CT čísla od snímkov b.) a c.). Ako zaujímavé detekcie taktiež hodnotím skeny označené ako b.), kde je malá časť pečenevého tkaniva oddelená od väčšej časti (malá časť je vyznačená červenou šípkou), no aj napriek tomu ju oba modely dokázali detekovať. Modely si taktiež dokázali poradiť aj s nejasnými hranicami u susedných orgánov (vyznačený červenou šípkou) ako je možné vidieť na reze c.).

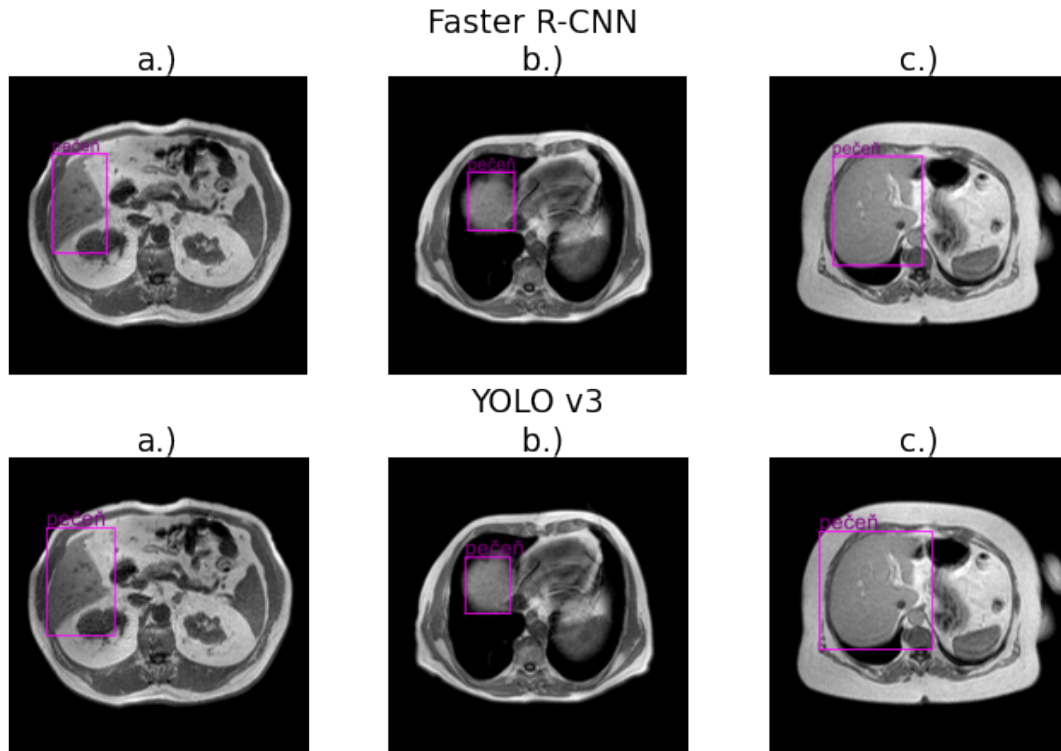
Tento dataset bol do bakalárskej práce vybraný na základe výziev, ktoré so sebou prináša. Jednou z nich je podobná Hounsfieldova jednotka susediacich orgánov, čo sa prejavilo aj na výsledkoch detekcie. Na obrázku 6.10 sú zobrazené často vyskytujúce sa detekcie v CT sade dát, kde „Faster“ R-CNN alebo YOLO v3 zlyhávali.



Obr. 6.10: Príklady detekcií s nízkym IoU medzi predvídaným a pravdivým rámčekom na CT rezoch. Prvý riadok predstavuje výsledky „Faster“ R-CNN modelu. Druhý riadok predstavuje výsledky YOLO v3 modelu.

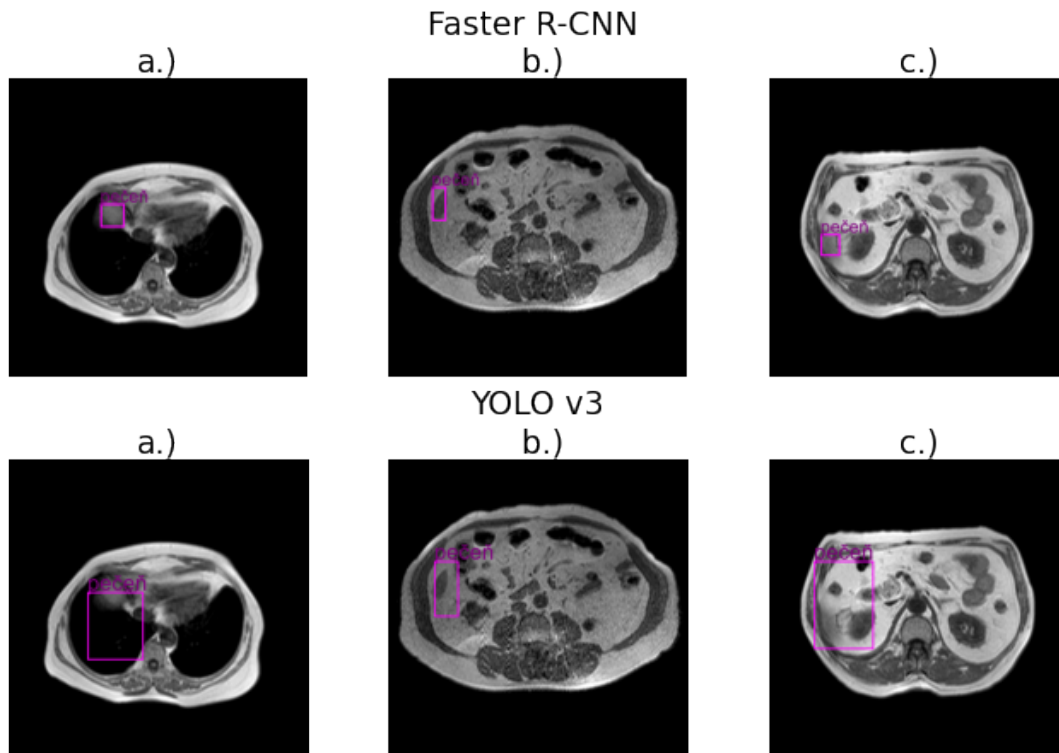
Rez označený písmenom a.) na obrázku 6.10 je najhoršou detekciou v CT datach u YOLO v3 modelu, keďže rámček je úplne mimo pečenevého tkaniva. „Faster“ R-CNN na rovnakom snímku objekt detekuje, avšak ohraničená oblasť je príliš veľká, čo spôsobuje nízke IoU s pravdivým rámčekom. Pri porovnaní rezov označených ako b.) je patrné, že YOLO v3 predikoval väčší rámček aj keď bol objekt malý. Prvá verzia YOLO modelu má problémy s nachádzaním malých objektov v obrázkoch, čo sa však verziou YOLO v3 a detekciou na 3 škálach čiastočne vyriešilo, no ako je možné vidieť, niekedy lokalizácia menších objektov zlyháva. Na skene c.) v sade obrázkov 6.10 je patrné, že do oblasti ohraničenej rámčekom je spolu s pečenuou u „Faster“ R-CNN zahrnuté aj srdce (vyznačené červenou šípkou), práve kvôli nejasej hranici a podobnému CT číslu blízkyh orgánov.

Na obrázku 6.11 sú zobrazené uspokojivé výsledky z detekcií pre T1 váhovanú duálnu vstupnú fázu. Opäť je možné pozorovať odlišné veľkosti pečene vrámci jednotlivých rezov a taktiež odlišnosti v jej tvare medzi pacientami. Nejasné hranice medzi orgánmi sú zobrazené napríklad v reze označenom písmenom b.).



Obr. 6.11: Príklady detekcií s vysokým prekryvom s anotovanou oblasťou na MRI rezoch v T1 váhovanej duálnej vstupnej fáze. Prvý riadok predstavuje výsledky „Faster“ R-CNN modelu. Druhý riadok predstavuje výsledky YOLO v3 modelu.

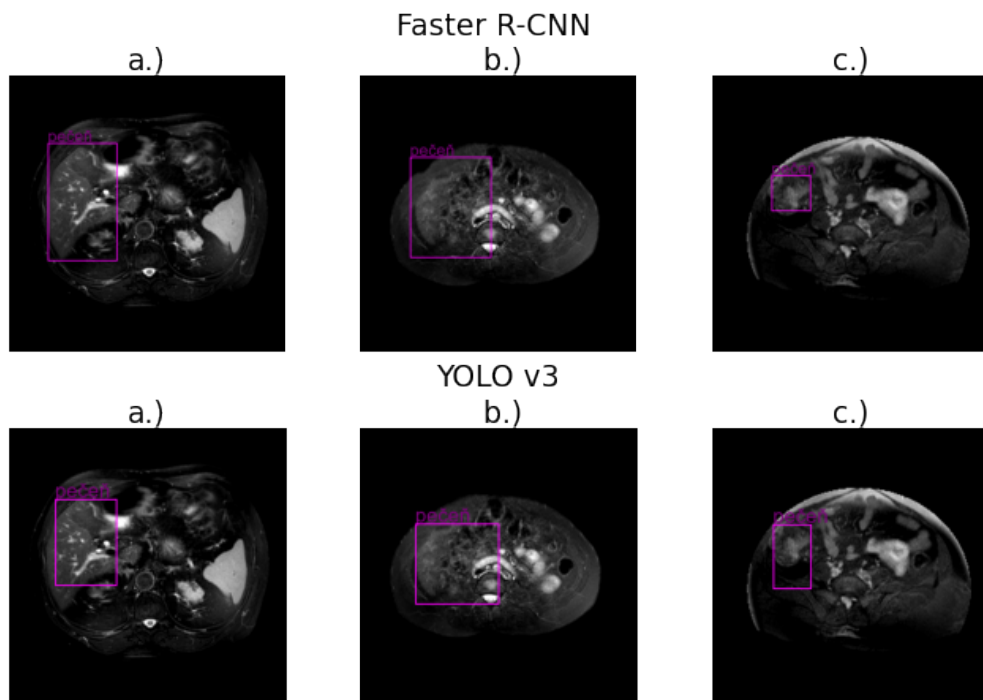
Na obrázku číslo 6.12 sú zobrazené detekcie z T1 duálnej vstupnej fázy. Detekcie na rezoch z „Faster“ R-CNN v prvom riadku majú vysokú presnosť. V druhom riadku sú pre porovnanie rovnaké rezy z YOLO v3, kde sú okrem pečene vymedzené aj susediace mäkké tkanivá kvôli neostrému rozhraniu medzi pečeňou a okolitými tkanivami.



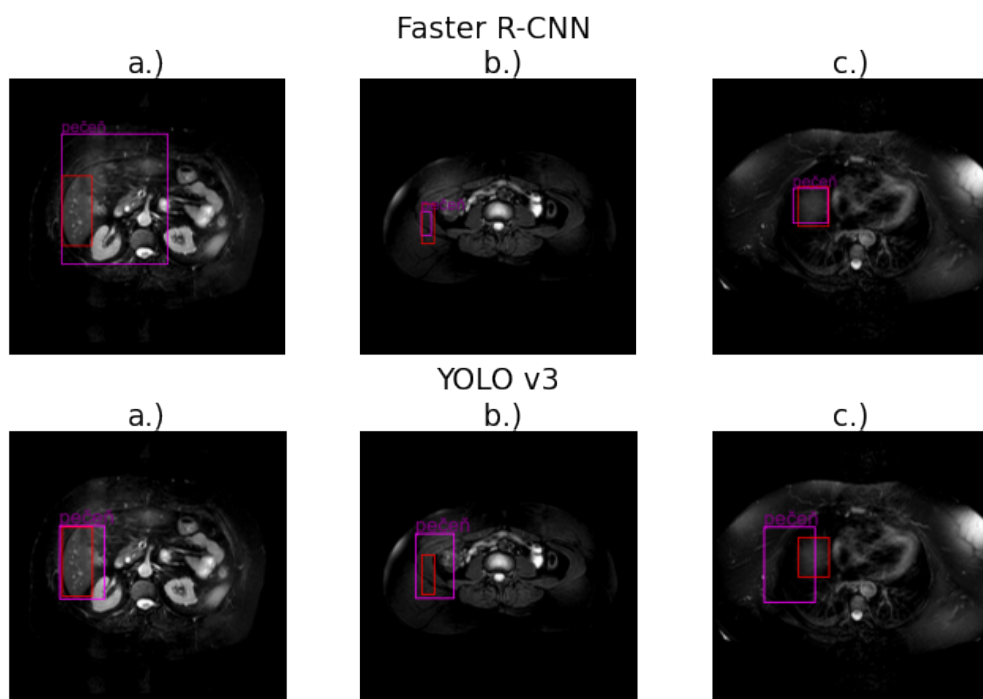
Obr. 6.12: Príklady detekcií na MRI rezoch z T1 váhovanej duálnej vstupnej fázy. Prvý riadok predstavuje výsledky „Faster“ R-CNN modelu s vysokým IoU medzi predikovanými a anotovanými rámcami. Druhý riadok predstavuje výsledky YOLO v3 modelu, kde je IoU na rovnakých rezoch nízke.

Na obrázkoch 6.13 a 6.14 sú zobrazené snímky z T2 váhovanej SPIR fázy MRI. Rezy sú viditeľne tmavšie a pečeňové žily sú intenzívne zvýraznené, čím sa data odlišujú od predchádzajúcich dvoch typov. Na obrázku 6.13 sú zobrazené rezy, kde boli dosiahnuté dobré výsledky.

Najnižšie presnosti u oboch modelov boli dosiahnuté u T2 váhovanej SPIR fázy MRI snímkov. Pečeň bola vo väčšine prípadov nájdená, ale predikovaný rámček bol príliš veľký alebo malý, čo znížilo IoU medzi anotovanou a predikovanou oblasťou, čoho dôsledkom je menšia presnosť. Príklady detekcií s menším prekryvom sú zobrazené na obrázku 6.14. Spolu s predikovanými boxmi (fialové) sú zobrazené aj anotované rámčeky (červené) z dôvodu, že u týchto snímkov je hranica medzi mäkkými tkanivami nedostatočne viditeľná.



Obr. 6.13: Príklady detekcií na MRI rezoch v T2 váhovanej SPIR fáze s vysokým IoU pre „Faster“ R-CNN v 1. riadku a pre YOLO v3 v 2. riadku.



Obr. 6.14: Príklady neúspešných detekcií na MRI rezoch v T2 váhovanej SPIR fáze. Pre „Faster“ R-CNN v 1. riadku je nepresná detekcia nachádzajúca sa v reze a.). Neúspešné detekcie pre YOLO v3 sa v 2. riadku nachádzajú na rezoch b.) a c.).

6.4 Využitie v medicíne

Detekcia orgánov brušnej dutiny je dôležitou prerekvizitou pre 3D vizualizáciu a tlač, ktoré slúžia pre hodnotenie ich polohy, tvaru, veľkosti ciev do nich vstupujúcich a taktiež vystupujúcich pred operačnými výkonmi. Detekcia pečene ako predspracovanie pre segmentáciu slúži hlavne ku klasifikovaniu adekvátnosti pečene pred transplantáciami. Pri transplantácii je pečeň chorého pacienta vyňatá z brušnej dutiny a je nahradená orgánom od darcu. Ten je umiestnený na rovnaké miesto, na ktorom sa nachádzala pôvodná pečeň a sú na seba postupne napojené všetky cievy a žľčovody, ktoré boli z pôvodnej pečene odpojené. Úspešná transplantácia pečene má vo väčšine prípadov za následok významné zlepšenie kvality života chorého.

Práve rastúci dopyt po transplantácii pečene a sprievodný nedostatok darcov s mozgovou smrťou zvýšili potrebu transplantácie od žijúcich darcov. Zabezpečenie bezpečnosti darcov a príjemcov je rozhodujúce. Predoperačná identifikácia anatómie pečene môže umožniť lepšie predoperačné chirurgické plánovanie, odvrátiť zbytočné operácie u pacientov s potenciálne nevhodnou anatómiou, a tým znížiť komplikácie transplantácie [1].

V poslednej dekáde vzrástol záujem o spracovanie biomedicínskych obrazov pomocou metód umelej inteligencie. Práve hlboké neurónové siete sú vhodné pre veľké medicínske dáta a môžu byť použité pre získavanie užitočných informácií.

Využitím detekcie anatomicky významných oblastí by mohol byť eliminovaný ľudský faktor, keďže hodnotenie lekárskeho snímkov býva často subjektívne. Pre dosiahnutie výsledkov detekcie dostačujúcich pre využitie v medicíne je však potrebné, aby učebná databáza obsahovala okrem veľkého množstva snímkov aj netypické prípady, či už tvaru, polohy orgánov alebo výskytu nezvyčajných lézií, aby model dosahoval v medicínskych aplikáciách nutnú excelentnú generalizačnú schopnosť. Vývoj zobrazovacích systémov však veľmi rýchlo napreduje, čím sa stáva zložitým zozbierať veľké množstvo abnormálnych prípadov pre nové modalities.

„Faster“ R-CNN a YOLO v3 majú vysokú presnosť pri detekcii objektov z reálneho sveta nasnímaných digitálnymi kamerami, tak ako uvádzajú články [18, 29]. Avšak pri aplikácii na medicínske dáta je samozrejme potrebné zväziť kvalitu obrazových dát, pomer signálu ku šumu, kontrastné a priestorové rozlíšenie, ktoré nie je vždy dostačujúce. Tieto moderné algoritmy umelej inteligencie však majú potenciál pomôcť lekárom a iným odborníkom s interpretáciou obsahu obrázku. Pre úlohu detekcie pečene z abdominálnych CT a MRI snímkov sa „Faster“ R-CNN vďaka vyššej presnosti javí ako vhodnejšia metóda. Rýchlosť YOLO v3 modelu v tejto aplikácii považujem za irelevantný faktor.

Záver

Cielom bakalárskej práce bolo zoznámiť sa s problematikou konvolučných neurónových sietí so zameraním na detekčné architektúry a vypracovať rešerš a teoretický rozbor dvoch algoritmov pre detekciu využívajúcich hlboké učenie. Z oblasti praktickej časti bolo cieľom vytvoriť ľubovoľnú učebnú databázu medicínskych snímok a aplikovať na ňu „Faster“ R-CNN a YOLO metódu vo vhodne zvolenom programovacom prostredí. Po aplikácii oboch algoritmov na medicínske dáta bolo nutné algoritmy porovnať, diskutovať výsledky a využitie v diagnostike.

Prvá kapitola sa zaoberá umelými neurónovými sieťami na ktoré je v nasledujúcej kapitole nadviazané konvolučnými neurónovými sieťami, hlavne ich architektúrou a druhom vrstiev. Algoritmy využívané pre detekciu objektov „Faster“ R-CNN a YOLO sú popísané v kapitole číslo 3.

Oba využívané modely boli implementované v programovacom jazyku Python s využitím knižnice Pytorch, ktorá podporuje konvolučné neurónové siete. Súčasťou štvrtej kapitoly je popísaná využitá sada dát CT a MRI abdominálnych snímok, ktorá bola získaná z verejne dostupného portálu „Grand Challenge“ a tvorili ju CT a MRI rezy od 40 pacientov. Časť MRI snímok bola získaná v T1 váhovanej duálnej vstupnej fáze a druhá časť v T2 váhovanej SPIR fáze. Ako detekovaný objekt bola zvolená pečeň.

Piata kapitola pojednáva o realizácii dvoch zvolených algoritmov a to súhrnne o vytvorených modeloch, výpočte chybových funkcií a ich následným tréňovaním.

V kapitole č. 6 sú zachytené a diskutované výsledky detekcie pečene získané pomocou opakovanej validácie náhodného výberu. Algoritmy sú porovnané na základe výpočtu presnosti pri rôznych prahoch IoU. Pri minimálnom prahu prekryvu 50 % dosahuje „Faster“ R-CNN presnosť 81,8 % a YOLO v3 78,9 %. Oba modely zároveň reagovali lepšie na CT ako na MRI data. Na zreteľ boli brané zároveň výsledky detekcie na rezoch z oboch MRI sekvencií, kde pri prekryve 50 % presnosť oboch modelov na T1 váhovanej vstupnej duálnej fáze prevýšila presnosť na T2 váhovanej SPIR fáze takmer o 17 %. Tieto menej uspokojujúce výsledky boli dosiahnuté z dôvodu menšieho počtu dostupných snímok a odlišného rozloženia farieb v snímkoch zo SPIR fázy MRI. Aj napriek priaznivejším výsledkom „Faster“ R-CNN, YOLO v3 algoritmus oproti nemu vykonáva menej chýb detekcie pozadia, to znamená detekcie s prekryvom menším ako 20 % a pri porovnaní modelov zľadiska rýchlosti zďaleka prevyšuje rýchlosť „Faster“ R-CNN.

Po zhodnotení dosiahnutých výstupov detekcie na dostupnej databáze sa využitie konvolučných neurónových sietí, konkrétne „Faster“ R-CNN a YOLO v3 modelov, javí ako sľubná technika pre detekciu objektov v medicínskych obrazoch.

Literatúra

- [1] KAVUR, A. E.; GEZER, N. S.; BARIS, M.; CONZE, P.; GROZA, V.; PHAM, D. D. CHAOS Challenge - Combined (CT-MR) Healthy Abdominal Organ Segmentation [online]. 2020, [cit. 29.05.2020]. Dostupné z URL: <<https://arxiv.org/abs/2001.06535>>.
- [2] MOUREK, J. Fyziologie: učebnice pro studenty zdravotnických oborů. Praha: Grada, 2005. ISBN 80-247-1190-7.
- [3] NIELSEN, M.A. Neural Networks and Deep Learning. In: *Determination Press* [online]. 2015, s.2–35. posledná aktualizácia 26.12.2019 [cit. 30.12.2019]. Dostupné z URL: <<http://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf>>.
- [4] Artificial Neural Networks In: *Cinnamon* [online]. [cit. 31.12.2019]. Dostupné z URL: <https://cinnamon.agency/blog/post/artificial_neural_networks>.
- [5] Neural Network Tutorial. In: *Neural Designer* [online]. [cit. 31.12.2019]. Dostupné z URL: <<https://www.neuraldesigner.com/learning/neural-networks-tutorial>>.
- [6] Everything you need to know about artificial neural network. In: *Medium: Technology, Invention, App, and More* [online]. 2015, posledná aktualizácia 28.12.2015 [cit. 30.12.2019]. Dostupné z URL: <<https://medium.com/technologyinventionand-more/everything-you-need-to-know-about-artificial-neural-57fac18245a1>>.
- [7] BUSHAEV, V. How do we ‘train’ neural networks? In: *Medium: Towards data science* [online]. 2017, posledná aktualizácia 27.11.2017 [cit. 30.12.2019]. Dostupné z URL: <<https://towardsdatascience.com/how-do-we-train-neural-networks-edd985562b73>>.
- [8] LEE, J.G.; JUN, S.; CHO, Y.W.; LEE, H; KIM, G.B.; SEO, J.B.; KIM, N. Deep Learning in Medical Imaging: General Overview. In: *Korean Journal of Radiology* [online]. 2017, [cit.30.12.2019], vol.18, no.4. ISSN 2005-8330. Dostupné z URL: <<https://doi.org/10.3348/kjr.2017.18.4.570>>.

- [9] MORNSTEIN, V. Lékařská fyzika a biofyzika. Brno: Masarykova univerzita, 2018. ISBN 978-80-210-8984-6.
- [10] BARMETTLER, J. The Full Story behind Convolutional Neural Networks and the Math Behind it. In: *Hackernoon* [online]. 2019, posledná aktualizácia 15. 10. 2019 [cit. 30. 12. 2019]. Dostupné z URL: <https://hackernoon.com/the-full-story-behind-convolutional-neural-networks-and-the-math-behind-it-2j4fk3zu2>.
- [11] Achieving 90% accuracy in Object Recognition Task on CIFAR-10 Dataset with Keras: Convolutional Neural Networks. In: *Machine Learning In Action: A perfect hands-on practice for beginners to elevate their ML skills* [online]. 2018, posledná aktualizácia 24. 3. 2018 [cit. 30. 12. 2019]. Dostupné z URL: <https://appliedmachinelearning.blog/2018/03/24/achieving-90-accuracy-in-object-recognition-task-on-cifar-10-dataset-with-keras-convolutional-neural-networks/>.
- [12] KANCHAN, S. ReLU : Not a Differentiable Function: Why used in Gradient Based Optimization? and Other Generalizations of ReLU. In: *Medium* [online]. 2018, posledná aktualizácia 31. 5. 2018 [cit. 30. 12. 2019]. Dostupné z URL: <https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec>.
- [13] Deep Learning Cage Match: Max Pooling vs Convolutions [online]. [cit. 30. 12. 2019]. Dostupné z URL: <https://mc.ai/deep-learning-cage-match-max-pooling-vs-convolutions/>.
- [14] GANDHI, R. R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms. In: *Medium: Towards data science* [online]. 2017, posledná aktualizácia 09. 07. 2018 [cit. 30. 12. 2019]. Dostupné z URL: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>.
- [15] GIRSHICK, R.; DONAHURE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *CoRR* [online]. 2013, [cit. 30. 12. 2019], vol. 1311, no. 2524. Dostupné z URL: <http://arxiv.org/abs/1311.2524>.
- [16] UIJLINGS, J.; VAN DE SANDE, K.; GEVERS, T.; SMEULDERS, A. Selective Search for Object Recognition. In: *International Journal of Computer Vision*

- [online]. 2013, [cit. 27. 05. 2010], Dostupné z URL:
<<https://doi.org/10.1007/s11263-013-0620-5>>.
- [17] GIRSHICK, R. Fast R-CNN. In: *The IEEE International Conference on Computer Vision (ICCV)* 2015, s. 1440–1448.
- [18] REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems*. 2015, s. 91–99.
- [19] REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You Only Look Once: Unified, Real-Time Object Detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. Los Alamitos, CA, USA: IEEE Computer Society, 2016, s. 779—788. ISSN 1063-6919. Dostupné z URL:
<<https://arxiv.org/abs/1506.02640>>.
- [20] SZEGEDY, C.; LIU, W.; YANG, J.; SHERMANET, P.; REED, S.; ANGLELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. Going Deeper with Convolutions [online]. 2014, [cit. 29. 05. 2020]. Dostupné z URL:
<<https://arxiv.org/abs/1409.4842>>.
- [21] KAVUR, A. E.; SELVER, M. A.; DICLE, O.; BARIS, M.; GEZER, N. S.; CHAOS - Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge Data (Version v1.03) [online]. 2019, [cit. 29. 05. 2020]. Dostupné z URL:
<<https://zenodo.org/record/3431873#XtEAajozY2w>>.
- [22] Grand Challenges in Biomedical Image Analysis. [online]. [cit. 29. 05. 2020]. Dostupné z URL:
<<https://grand-challenge.org/>>.
- [23] DENOTTER, T. D.; SCHUBERT, J. In: *StatPearls* [online]. Treasure Island (FL): StatPearls, 2020, posledná aktualizácia 11. 5. 2020 [cit. 4. 6. 2020]. Dostupné z URL:
<<https://www.ncbi.nlm.nih.gov/books/NBK547721/>>.
- [24] Welcome to Python.org [online]. [cit. 29. 05. 2020]. Dostupné z URL:
<<https://www.python.org/>>.
- [25] Pytorch.org [online]. [cit. 29. 05. 2020]. Dostupné z URL:
<<https://www.python.org/>>.
- [26] SANDLER, M.; HOWARD, A.G.; ZHU, M.; ZHMOGINOV, A.; CHEN L.C. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. In: *CoRR*) [online]. 2018, [cit. 29. 05. 2020].

- Dostupné z URL:
<<http://arxiv.org/abs/1801.04381>>.
- [27] ZHANG, Z.; SABUNCU, M.R. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In: *CoRR* [online]. 2018, [cit. 29.05.2020]. Dostupné z URL:
<<http://arxiv.org/abs/1805.07836>>.
- [28] ImageNet [online]. [cit. 29.05.2020]. Dostupné z URL:
<<http://www.image-net.org/>>.
- [29] REDMON, J.; FARHADI, A. YOLOv3: An Incremental Improvement. In: *CoRR* [online]. 2018, [cit. 29.05.2020]. Dostupné z URL:
<<https://arxiv.org/pdf/1804.02767.pdf>>.
- [30] KATHURIA, A. What's new in YOLO v3? In: *Medium: Towards data science* [online]. 2018, posledná aktualizácia 23.4.2018 [cit. 30.12.2019]. Dostupné z URL:
<<https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>>.
- [31] HALL, M.A. Correlation-based Feature Selection for Machine Learning. [online]. 1999, s. 17–18. [cit. 29.05.2020]. Dostupné z URL:
<<https://www.lri.fr/~pierres/donn%E9es/save/these/articles/lpr-queue/hall199correlationbased.pdf>>.
- [32] SHAH, T. Measuring Object Detection models – mAP – What is Mean Average Precision? In: *Medium: Towards data science* [online]. 2018, posledná aktualizácia 16.1.2018 [cit. 30.12.2019]. Dostupné z:
<<https://towardsdatascience.com/what-is-map-understanding-the-statistic-of-choice-for-comparing-object-detection-models-1ea4f67a9dbd>>.
- [33] AIDOUNI, M.E. Evaluating Object Detection Models: Guide to Performance Metrics [online]. 2019, posledná aktualizácia 5.10.2019 [cit. 30.12.2019]. Dostupné z URL:
<<https://manalelaidouni.github.io/manalelaidouni.github.io/Evaluating-Object-Detection-Models-Guide-to-Performance-Metrics.html#average-precision>>.
- [34] COCO dataset–Common Object In Context [online]. [cit. 29.05.2020]. Dostupné z URL:
<<http://cocodataset.org>>.

Zoznam príloh

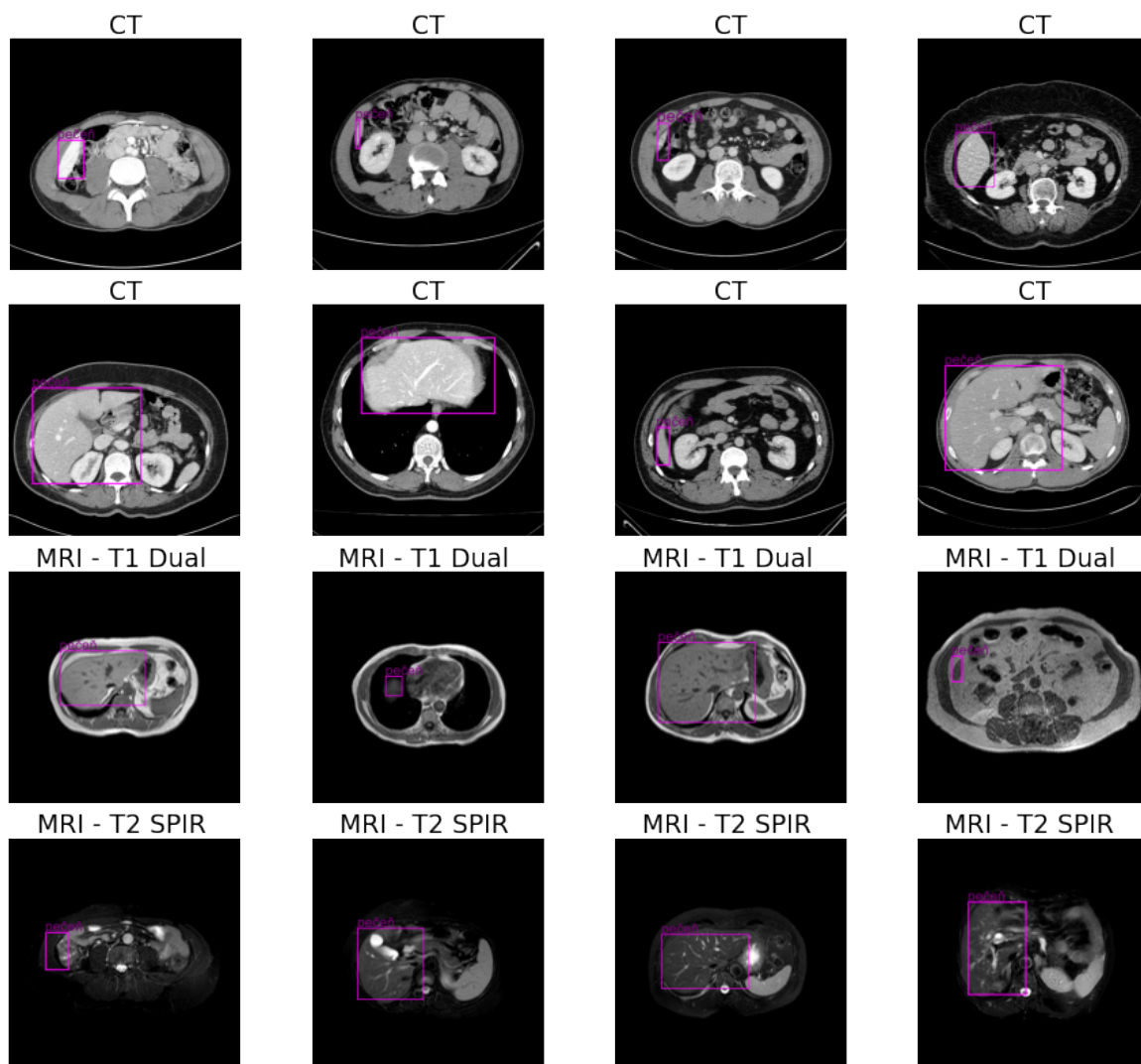
A	Tabuľka presností	63
B	Ukážky detekcií z modelu „Faster“ R-CNN	64
C	Ukážky detekcií z modelu YOLO v3	65
D	Obsah príloh v informačnom systéme VUT	66

A Tabuľka presností

Tab. A.1: Tabuľka presností pre opakované validácie náhodného výberu pri minimálnom prahu prekryvu predikovanej oblasti a anotovanou 50 %. V tabuľke sú presnosti vypočítané pre model „Faster“ R-CNN a model YOLO v3 a pre obe modalities (CT a MRI). Zároveň je uvádzaná celková presnosť daného modelu na všetkých datach z validačnej množiny (CT+MRI).

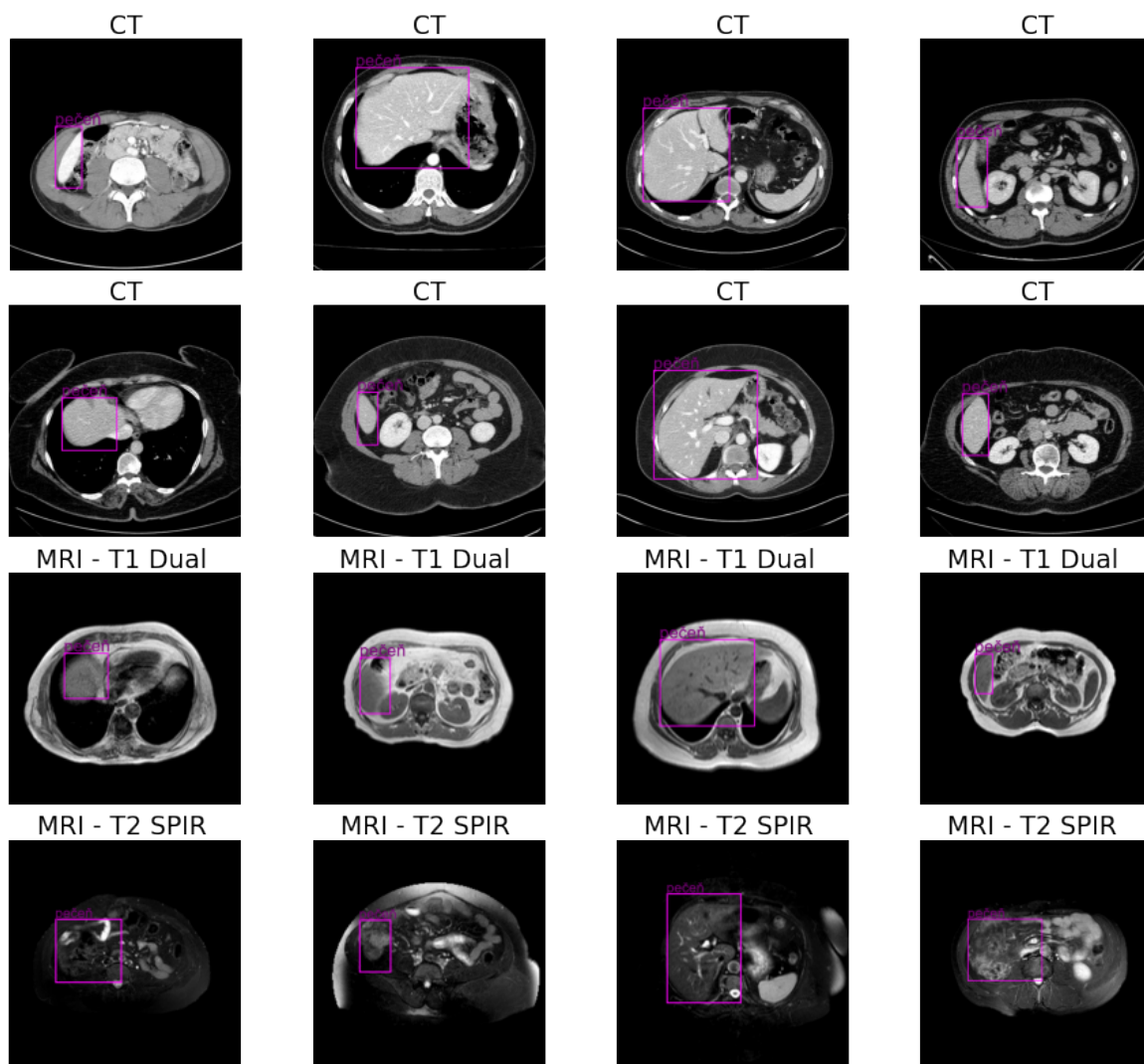
Presnosti detekcií pri Iou 50 %[%]						
Validácia	Faster R-CNN			YOLO		
modalita	CT	MRI	CT+MRI	CT	MRI	CT+MRI
1	83,3	68	75,7	78,6	80	79,3
2	88,7	86,5	87,6	83,7	68,4	76,1
3	81,3	74,8	78,1	85,3	77,9	81,6
4	82,9	85,9	84,4	83,3	67,8	75,6
5	81,7	80,1	80,9	88	72,9	80,5
6	84,9	77,7	81,3	86,7	78,7	82,7
7	81,1	76,6	78,9	89,3	71,2	80,3
8	88,6	89,3	89,0	85,7	68,3	77
9	89,8	76,8	83,3	82,3	72,3	77,3
10	85,9	72	79,0	83,6	72,7	78,2
priemer	84,8	78,8	81,8	84,7	73	78,9

B Ukážky detekcií z modelu „Faster“ R-CNN



Obr. B.1: Ukážky výsledných detekcií z „Faster“ R-CNN modelu. V prvých dvoch riadkoch sa nachádzajú ukážky z CT rezov, v treťom riadku výsledky na MRI rezoch z T1 vstupnej duálnej fázy a vo štvrtom riadku sú zobrazené skeny z MRI T2 váhovanej SPIR fázy.

C Ukážky detekcií z modelu YOLO v3



Obr. C.1: Ukážky výsledných detekcií z YOLO v3 modelu. V prvých dvoch riadkoch sa nachádzajú ukážky z CT rezov, v treťom riadku výsledky na MRI rezoch z T1 vstupnej duálnej fázy a vo štvrtom riadku sú zobrazené skeny z MRI T2 váhovanej SPIR fázy.

D Obsah príloh v informačnom systéme VUT

Súčasťou bakalárskej práce sú prílohy v informačnom systéme VUT, ktoré obsahujú tieto zložky.

```
/ ..... koreňový adresár priložených súborov
├── faster_rcnn ..... prečínok so zdrojovými kódmi pre „Faster“ R-CNN
│   ├── train_rcnn_demo.py ..... ukázkový skript pre tréning „Faster“ R-CNN
│   ├── detection_script_rcnn.py ..... skript určený pre testovanie detekcie pečene
│   ├── checkpoints ..... priečínok pre váhy natrénovaného modelu
│   ├── images ..... priečínok pre uloženie obrázkov zvolených pre detekciu
│   ├── masks .... priečínok s ukázkami masiek z ktorých sú vytvárané anotácie počas
│   │   tréningu
│   ├── names
│   │   ├── train_names.csv ..... súbor s názvami obrázkov pre tréning
│   │   └── valid_names.csv ..... súbor s názvami obrázkov pre validáciu
│   └── output_images_rcnn ..... priečínok pre uloženie výsledných detekcií
├── YOLO ..... prečínok so zdrojovými kódmi pre YOLO v3
│   ├── train_yolo_demo.py ..... ukázkový skript pre tréning YOLO v3
│   ├── detection_script_yolo.py ..... skript určený pre testovanie detekcie pečene
│   ├── models.py ..... model YOLO v3
│   ├── checkpoints ..... priečínok pre váhy natrénovaného modelu
│   ├── images ..... priečínok pre uloženie obrázkov zvolených pre detekciu
│   ├── masks .... priečínok s ukázkami masiek z ktorých sú vytvárané anotácie počas
│   │   tréningu
│   ├── names
│   │   ├── train_names.csv ..... súbor s názvami obrázkov pre tréning
│   │   └── valid_names.csv ..... súbor s názvami obrázkov pre validáciu
│   ├── output_images_yolo ..... priečínok pre uloženie výsledných detekcií
│   ├── config
│   │   └── yolov3.cfg ..... súbor obsahujúci architektúru YOLO v3
│   ├── utils
│   │   ├── parse_config.py .....
│   │   └── utils.py ..... funkcie potrebné pre chod YOLO v3
├── requirements.txt ..... požiadavky na knižnice a ich verzie
└── readme.txt ..... návod k používaniu modelov pre detekciu pečene
```