F3

Fakulta elektrotechnická
Katedra kybernetiky

Bakalářská práce

Multimodální detektor polohy a rychlosti vozidel v provozu

Vojtěch Tilhon

Květen 2023
Vedoucí práce: Mgr. Martin Pecka, Ph.D.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Tilhon** Jméno: **Vojtěch** Osobní číslo: **499121**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra kybernetiky**

Studijní program: **Otevřená informatika**

Specializace: **Základy umělé inteligence a počítačových věd**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Multimodální detektor polohy a rychlosti vozidel v provozu

Název bakalářské práce anglicky:

Multimodal Detector of Pose and Velocity of Vehicles in General Traffic

Pokyny pro vypracování:

Cílem práce je navržení, implementace a natrénování detektoru vozidel z kamerových a lidarových dat. Výstupem detektoru má být 6D bounding box a odhad vektoru rychlosti/zrychlení každého vozidla ve scéně.
Specifikem projektu je práce s daty ze senzorů středně velkých mobilních robotů, tj. s daty snímanými z mnohem nižší výšky než je obvyklé v automotive datasetech.
Jedním z výstupů projektu by také měl být kvalitní trénovací dataset pro obě modality nasbíraný na robotech katedry kybernetiky.
Úkolem studenta je také prozkoumat existující implementace unimodálních detektorů (ať už z barevných kamerových obrázků nebo 3D lidarových skenů), popsat jejich slabiny vzhledem k zadané úloze a případné výhody oproti multimodálnímu řešení. Součástí práce by mělo být i kvantitativní porovnání výsledků vyvinutého multimodálního detektoru a již existujících unimodálních řešení (natrénovaných na vytvořeném datasetu).

Seznam doporučené literatury:

[1] Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 12697-12705).
[2] A Rahim, Hasliza & Sheikh, Usman Ullah & Ahmad, R.Badlishah & Md Zain, Aini Syuhada & Firuz, Suryani. (2010). Vehicle speed detection using frame differencing for smart surveillance system. 10th International Conference on Information Sciences, Signal Processing and their Applications, ISSPA 2010. 630-633. 10.1109/ISSPA.2010.5605422.
[3] Simonelli, A., Bulo, S. R., Porzi, L., Ricci, E., & Kontschieder, P. (2020, August). Towards generalization across depth for monocular 3d object detection. In European Conference on Computer Vision (pp. 767-782). Springer, Cham.
[4] Wang, T., Zhu, X., Pang, J., & Lin, D. (2021). Fcos3d: Fully convolutional one-stage monocular 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 913-922).

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Mgr. Martin Pecka, Ph.D. vidění pro roboty a autonomní systémy FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **09.01.2023** Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Mgr. Martin Pecka, Ph.D.	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. _____
Datum převzetí zadání

_____
Podpis studenta

Poděkování / Prohlášení

Děkuji svému vedoucímu Mgr. Martinu Peckovi, Ph.D., který mi při psaní závěrečné práce byl vždy k dispozici a nápomocný. Dále děkuji svým rodičům, že mě během studia podporovali a živili. V neposlední řadě děkuji svým kamarádům a přátelům, již mi vždy dokázali zvednout náladu, když to bylo zapotřebí.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne 26. 5. 2023

.....

Abstrakt / Abstract

Detekce vozidel je rychle se rozvíjející disciplínou v oboru počítačového vidění s uplatněním nejen v oblasti samořiditelných dopravních prostředků, ale také autonomních pohyblivých robotů. K rozpoznávání se v praxi často využívá dvou modalit, jmenovitě mračen bodů obdržených z lidarových měřičů a RGB hodnot kamer umístěných na vozidle či robotu. Tato práce si klade za cíl navrhnout, implementovat a natrénovat takový detektor pro středně velké roboty katedry kybernetiky. Dále je třeba vytvořit vlastní trénovací multimodální dataset, neboť volně přístupná trénovací data jsou pořizována z vozidel, která jsou mnohem větší než roboty, a nejsou tak pro naše účely reprezentativní.

Klíčová slova: RGB, lidar, multimodální detektce vozidel, odhad rychlosti vozidel, PointPillars

Vehicle detection is a rapidly developing discipline of the field of computer vision which has applications not only in the area of self-driving vehicles, but also robot locomotion. Two common modalities for object detection are point clouds obtained from lidar scanners and RGB values from cameras situated on the vehicle or robot. This work aims to propose, implement and train such a detector for robots of the Department of Cybernetics. In addition, it is of importance to collect and annotate a multimodal dataset since freely available datasets are recorded from vehicles much larger than our robots and are therefore not suitable for our use case.

Keywords: RGB, lidar, multimodal vehicle detection, vehicle velocity estimation, PointPillars

Title translation: Multimodal detector of pose and velocity of vehicles in traffic

/ Obsah

Tabulky / Obrázky

Kapitola 1
Úvod

Detekce 3D objektů se v posledních letech stala atraktivním bodem výzkumu v oblasti 3D počítačového vidění. Jejím úkolem je z předložných dat z různých senzorů, jako mohou být například kamery, termokamery, hloubkové kamery, lidary či radary, určit 3D informace o objektech nacházejících se ve scéně. Předmětem zájmu mohou být například pozice a orientace objektů v prostoru, jejich tvar nebo rychlost a zrychlení. Roboty, na kterých bude algoritmus detekce spouštěn, jsou vybaveny několika barevnými kamerami, jejichž kombinovaný úhel pohledu je plný úhel, a jedním lidarem. V této práci se budeme zejména zaměřovat na lokalizaci a odhad rychlosti automobilů v prostoru za použití lidarových snímků a RGB dat z kamer získaných z robotů katedry kybernetiky.

Formálněji řečeno cílem úlohy detekce 3D objektů je pro naše účely nalezení funkce, jejímž vstupem je RGB obrázek (případně množina RGB obrázků) a množina bodů ve 3D prostoru a výstupem je množina bounding boxů a jejich tříd, kde každý bounding box má umístění v prostoru, rozměry a úhel, který určuje rotaci podle vertikální osy. Jiné rotace objektů podobně jako další literatura[1][2][3] pro jednoduchost neuvažujeme, protože vozovka je zpravidla rovina a není obvyklé, aby vozidla svírala nenulový úhel s povrchem vozovky.

Detekce a lokalizace z RGB a lidarových dat vozidel na vozovce je tématem mnoha prací, které tuto problematiku řeší v kontextu autonomních motorových vozidel. Trénovací množiny, které jsou zde běžně vytvářeny a používány, jsou proto většinou pořizovány z osobních automobilů jedoucích na vozovce. Senzory bývají namontovány na střechách automobilů. Naším cílem ale je implementovat detektor, který bude nasazen na středně velkých mobilních robotech katedry kybernetiky, které jsou značně menší než automobily, za účelem bezpečného přejíždění silnic. Data z našich robotů a ze samořiditelných vozidel jsou rozdílná a proto není vhodné již existující metody aplikovat bez modifikací. Z tohoto důvodu jsme také v rámci této práce nasbírali a ručně anotovali náš vlastní dataset pomocí robotů katedry kybernetiky.

V následujícím textu shrneme metody, které řeší problém 3D, ale i 2D detekce objektů a které souvisejí s touto prací. Pak popíšeme implementaci a architekturu našeho řešení a experimentálně porovnáme s již existujícími metodami. Dále představíme trénovací množiny, které jsou běžně ve výzkumu 3D detekce vozidel používány, a náš vlastní dataset. V závěru práce vyhodnotíme náš detekční přístup a porovnáme ho s jinými metodami a naznačíme další možná vylepšení.

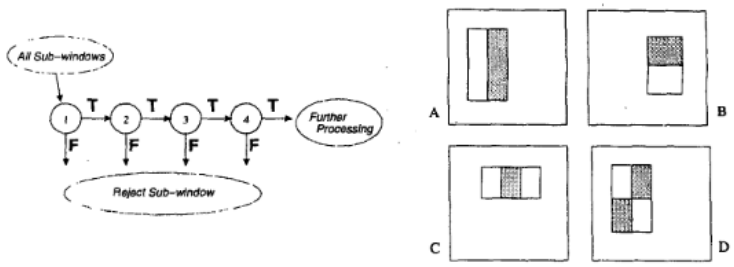
Kapitola 2
Související práce

V této kapitole uvedeme krátký přehled současných metod strojové detekce objektů. Nejprve v krátkosti uvedeme přístupy pro detekci 2D bounding boxů, protože tyto přístupy úzce souvisí s 3D detekcí a mnohé metody 3D detekce jsou adaptacemi 2D metod, a navážeme stručným přehledem metod 3D detekce.

2.1 2D detekce objektů

2D detekce objektů je v porovnání s 3D detekcí značně jednodušší problém, neboť je potřeba provést regresi jen 4 reálných čísel (x-ová a y-ová souřadnice bounding boxu a jeho šířka a výška), namísto 6 v případě 3D bounding boxů.

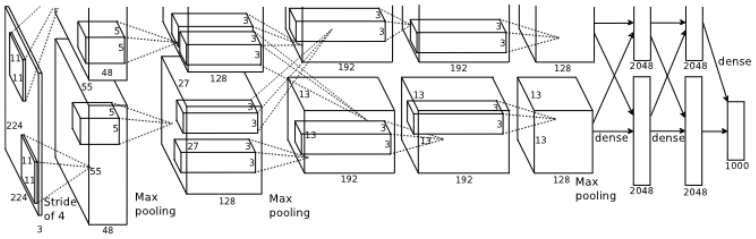
Již v roce 2001 navrhli Paul Viola a Micheal Jones algoritmus pro detekci obličejů v barevných obrázcích, který byl schopen běžet v reálném čase[4]. Hlavní myšlenkou algoritmu je uvažovat všechny možné bounding boxy v obrázku a na tyto aplikovat klasifikační algoritmus *AdaBoost*[5], kde příznakový prostor je tvořen různými součty a rozdíly hodnot pixelů v uvažovaném bounding boxu. Nad tímto prostorem je pro každý příznak vytvořen slabý binární klasifikátor s číselným prahem, který přiřazuje bounding boxu na jedné straně prahu pozitivní třídu (přitomnost obličeje) a na druhé straně negativní třídu (absence obličeje). Následně AdaBoost ze slabých klasifikátorů vytvoří silný binární klasifikátor. Rychlost tohoto algoritmu je zejména zajištěna efektivním výpočtem příznaků pomocí technik dynamického programování a možností sekvenční aplikace jednotlivých slabých klasifikátorů, která způsobí, že velké množství kandidátů je vyřazeno v rané části výpočtu.



Obrázek 2.1. Vlevo je znázorněno sekvenční rozhodování detektoru, který postupně uvažuje sofistikovanější a sofistikovanější příznaky uvažovaného bounding boxu. To mu umožňuje zamítnout jako negativní velké množství kandidátů s nízkým výpočetním časem. Vpravo jsou vizualizovány příklady některých použitých příznaků, kde součet hodnot pixelů světle vyznačených oblastí je odečten od součtu tmavých oblastí. Oba obrázky jsou převzaty z [4].

V posledních letech dominují výzkumu metody využívající konvoluční neuronové sítě. Konvoluční vrstvy se ukázaly být v problému klasifikace objektů v obrázcích velmi užitečnými, protože dobře postihují invariant posunu objektu v obrázku. Jako

vstup do neuronové sítě většinou slouží samotné normalizované hodnoty pixelů obrázku. Na obrázek se totiž dá hledět jako na 3D tensor s rozměry $H \times W \times C$, kde H je výška, W šířka obrázku a C je počet kanálů ($C = 3$ v případě RGB). Modely využívající konvoluční neuronové sítě a algoritmus zpětného šíření chyby[1], který je používán pro jejich trénování, se v praxi ukázaly již na konci 20. století[6][7], ale větší pozornosti se dočkaly v roce 2012, kdy model pojmenovaný *AlexNet*[8] snížil v soutěži *ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC 2012)*[9] chybu klasifikace na datasetu[10] z 25,7 % na 17,0 %. I v dalších letech se objevovala vylepšení tohoto klasifikačního přístupu[11][12][13], ve kterých byly zkoumány jednotlivé prvky architektury sítě, jako jsou například počty filtrů konvolučních vrstev a počet vrstev samotných.



Obrázek 2.2. Na obrázku jsou vykresleny jednotlivé vrstvy sítě *AlexNet* a jejich rozměry. Vlevo je vstupní obrázek, následuje pět konvolučních vrstev a dále tři lineární vrstvy. Výstup sítě o velikosti 1000 slouží jako vstup do *softmaxu*, jehož výstupem jsou samotná pravděpodobnostní skóre jednotlivých tříd. Obrázek je převzat z [8].

Tyto sítě neslouží k detekci a lokalizaci objektů v obrázku, nýbrž ke klasifikaci obrázků samotných, ale i tak je jejich zmínka v kontextu detekčních systémů důležitá, neboť jsou základem metod, které jsou detekce schopné.

2.1.1 Dvoufázové detektory

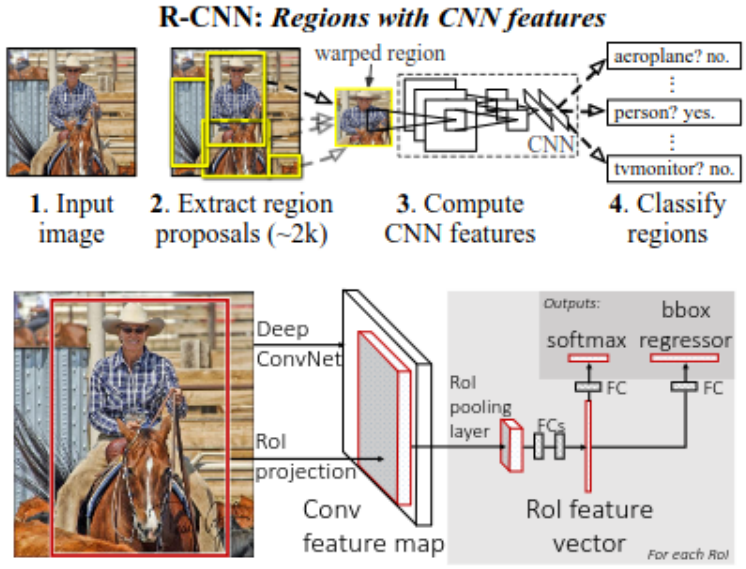
Prvním úspěšným pokusem o detekci 2D objektů v obrázcích pomocí konvolučních neuronových sítí byla architektura R-CNN[14]. R-CNN nejprve vygeneruje pro vstupní obrázek 2000 návrhů bounding boxů, které využije jako vstup do konvoluční neuronové sítě, z níž jako výstup obdrží vektor příznaků fixní velikosti. Každý navržený bounding box klasifikuje pomocí lineárního klasifikátoru metodou podpůrných vektorů[2]. Na této metodě dále staví metoda *fast R-CNN*[15], která přinesla zejména zrychlení výpočtu příznakových vektorů navržených bounding boxů v příznakové mapě obrázku. Další vylepšení poskytla architektura *faster R-CNN*[16], která představila nový a výpočetně mnohem efektivnější způsob navrhování bounding boxů pomocí hluboké konvoluční neuronové sítě nazvané *region proposal network (RPN)* a zkombinavala jej s detekční metodou *fast R-CNN*.

Tyto architektury jsou nazývány dvoufázové, protože se dají vnímat jako kombinace dvou částí: 1) navržení velkého množství bounding boxů, 2) transformace bounding boxů v obrázku do příznakového prostoru a klasifikace.

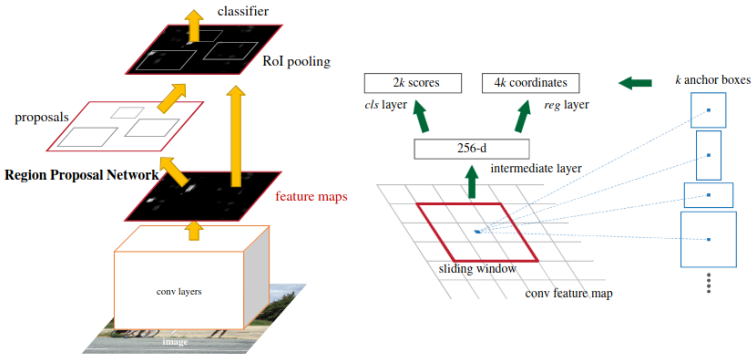
2.1.2 Jednofázové detektory

You Only Look Once (YOLO)[17] představil v roce 2016 nový způsob detekce objektů v obrázcích. Narozdíl od dvoufázových detektorů se *YOLO* skládá pouze z jedné

[1] V anglické literatuře označovaný *error backpropagation*.
[2] Také označovaná *support vector machine (SVM)*.



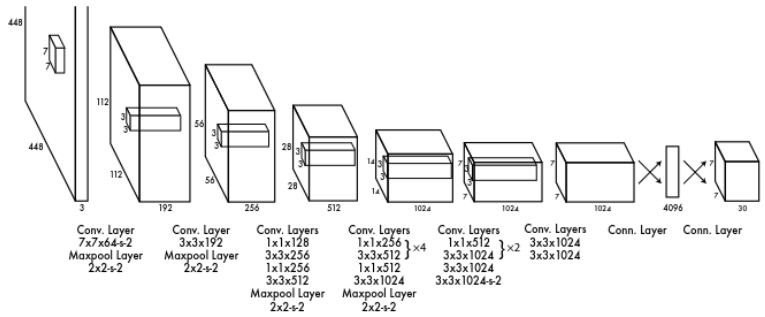
Obrázek 2.3. Nahoře je diagram architektury *R-CNN* a dole *fast R-CNN*. Všimněme si, že *R-CNN* počítá příznaky každého navrženého bounding boxu zvlášť, kdežto *fast R-CNN* nejdříve aplikuje neuronovou síť na celý obrázek, čímž značně zefektivňuje výpočet. Dále *R--CNN* využívá pro klasifikaci bounding boxů metodu podpůrných vektorů, ale *fast R-CNN* jednovrstvý perceptron pro regresi posunů bounding boxů a jiný jednovrstvý perceptron následovaný softmaxem pro klasifikaci. Obrázky jsou převzaty z [14] a [15].



Obrázek 2.4. *Faster R-CNN* nově přinesl síť *RPN* pro navrhování kandidátů, kterou spojil s původní sítí *fast R-CNN*, čímž dvě fáze detekce unifikoval. Vlevo je znázorněna celá síť *faster R-CNN* včetně *RPN*. Vpravo je detailněji zobrazena síť *RPN*, která využívá aprioních anchor boxů, k nimž v každém uvažovaném místě příznakové mapy generuje relativní souřadnice a pravděpodobnost, že vygenerovaný anchor box je objektem či nikoli. Obrázky jsou převzaty z [16].

konvoluční neuronové sítě, která má za úkol provést regresi bounding boxů objektů a pravděpodobností příslušnosti k jednotlivým třídám. Vstupní obrázek je rozdělen do pravidelné mřížky $S \times S$ bloků, kde S je hyperparametr sítě určující jemnost mřížky ($S = 7$ v původní publikaci). Síť pro každý blok v mřížce provede regresi B bounding boxů ($B = 2$), kde každý bounding box je jednoznačně určen čtyřmi čísly - souřadnicemi středu a rozměry relativně k velikosti a pozici bloku, a jejich jistot a C podmíněných pravděpodobností, kde C je počet uvažovaných skrytých tříd (v původní publikaci $C = 20$), říkajících, jaká je pravděpodobnost příslušnosti k dané třídě za podmínky, že v tom kterém bloku je objekt. Tedy celkem je výstupem sítě tensor o

velikosti $S \times S \times (5B + C)$, pro konkrétní hodnoty parametrů $7 \times 7 \times 30$. Tento přístup je schopný detekce několikrát rychleji než předešlé dvoufázové přístupy a utrpěl jen malé zhoršení kvality lokalizace detekovaných objektů.



Obrázek 2.5. Neuronová síť detektoru *YOLO* se skládá z 24 konvolučních vrstev následovaných 2 lineárními. Výstupem je přímo třídimensionální tensor obsahující souřadnice bounding boxů a podmíněné pravděpodobnosti pro každý blok obrázku. Obrázek je převzat z [17].

Metoda *YOLO* byla dále iterativně vylepšována. *YOLOv2*[18] mimo jiné přinesl do návrhu apriorní anchor boxy, které byly inspirovány jejich použitím v síti *RPN*. Síť tedy neprovádí regresi samotných složek bounding boxu, nýbrž posuny souřadnic relativně k apriornímu anchor boxu.

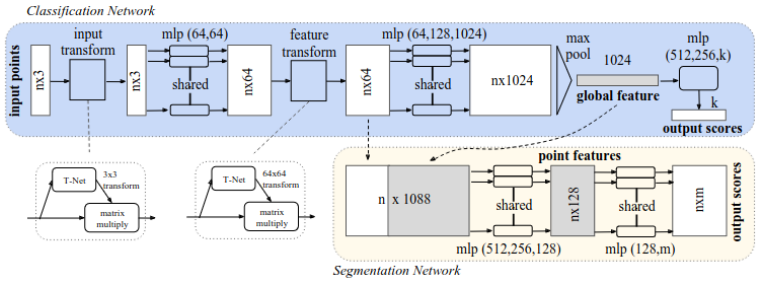
2.2 3D detekce objektů

Podobně jako u 2D detektorů jsou i u 3D verze této úlohy populárním prostředkem řešení neuronové sítě. Nejčastěji jako vstup do detekčního systému slouží mračna bodů, tedy množina bodů skládajících se ze souřadnic ve 3D prostoru a případných dalších složek jako mohou být reflektivita, intensita, atd.

2.2.1 Metoda PointNet

PointNet[19] je klasifikační a segmentační algoritmus, který operuje na mračnech 3D bodů. Tato metoda v jednoduchosti nejprve aplikuje na každý bod zvlášť stejnou transformaci (získanou regresní neuronovou sítí) následovanou dvouvrstvým perceptronem, který příznakový prostor $\mathbb{R}^3$ promítne do $\mathbb{R}^{64}$, a další transformací. Transformace bodů jsou realizovány přenásobením maticemi transformací, jejichž hodnoty jsou cílem regrese zmíněných regresních sítí. Na každý z těchto bodů z $\mathbb{R}^{64}$ je dále aplikován třívrstvý perceptron, který každý bod promítne do $\mathbb{R}^{1024}$. Metoda využívá vlastnosti množiny vstupních bodů, že nezáleží na jejich pořadí, a z těchto bodů vytvoří jeden vektor globálních příznaků z $\mathbb{R}^{1024}$ pomocí symetrické operace maximum. Konečně na globální příznaky použije třívrstvý perceptron, jehož výstupem je k-dimensionální vektor klasifikace množiny bodů, kde k je počet skrytých tříd.

Tuto metodu zmiňujeme, ačkoli se nejedná o detekční algoritmus, protože myšlenka vytvoření globálních příznaků z množiny bodů je hojně využívaná i v dalších detekčních přístupech, jmenovitě *VoxelNet*[1], *SECOND*[2] a *PointPillars*[3].



Obrázek 2.6. Na obrázku jsou znázorněny jednotlivé vrstvy sítě *PointNet*. V horní části je vidět proces vytváření vektoru globálních příznaků z množiny n bodů a následná klasifikační třívrstvá lineární síť. Obrázek je převzat z [19].

2.3 Metody využívající lidarové snímky

Zhruba řečeno se dají současné přístupy rozřadit do tří ne nutně disjunktních kategorií: 1) *voxelizace prostoru*, 2) *nahlížení na scénu z ptačího pohledu* a 3) *konstrukce hloubkového obrazu z pohledu autonomního agenta*.

VoxelNet[1] je metoda, která prostor rozdělí do 3D mřížky voxelů a z množiny bodů v každém voxelu vytvoří vektor příznaků podobným způsobem jako bylo uvedeno v sekci 2.2.1. Tím vzniká 4D tensor, na který lze aplikovat 3D konvoluce a získat tak lokální informace na úrovni voxelů. Tyto jsou dále poslány do sítě *Region Proposal Network (RPN)*, která provádí regresi bounding boxů a jejich klasifikaci.

Sparsely Embedded Convolutional Detection (SECOND)[2] je algoritmus stavějící na *VoxelNet*[1]. *SECOND*[2] využívá četnosti prázdných voxelů, čímž zlepšuje rychlost sítě. Další změnou je ztrátová funkce a regresní vrstva: místo regrese celého úhlu rotace bounding boxu podle vertikální osy je prováděna regrese úhlu v intervalu $[-\frac{\pi}{2}, +\frac{\pi}{2})$ a regrese binárního příznaku určujícího směr objektu. Dále není prováděna regrese rozměrů bounding boxu, nýbrž jsou místo nich použity předem spočtené průměrné rozměry pro danou třídu.

PointPillars[3] se dívá na scénu z ptačího pohledu a prostor rozdělí do 2D mřížky nekonečně dlouhých sloupců. Dále vytvoří pro fixní počet neprázdných sloupců vektor příznaků (využívá přitom četnost prázdných sloupců podobně jako *SECOND*[2]), čímž vzniká 3D tensor, na který lze pohlížet jako na pseudo-obrázek. Na tento tensor je možné aplikovat 2D konvoluční vrstvy a vytvořit tak příznaky ve třech různých rozlišeních. Výstupy konvolučních vrstev jsou spojeny a poslány do detekční hlavice *Single Shot Detector (SSD)*, která provádí regresi pozice bounding boxu, jeho rozměrů, úhlu v intervalu $[-\frac{\pi}{2}, +\frac{\pi}{2})$ a binárního příznaku určujícího směr objektu podobně jako metoda *SECOND*[2].

Zmiňme také metodu *YOLO3D*[20], která vytváří dvě obrázkové reprezentace vstupního mračna bodů: první je dána výškou nejvyššího bodu v jednotlivých buňkách mřížky při pohledu shora, druhá odpovídá hustotě bodů v dané buňce. Tyto reprezentace lze interpretovat jako dvoukanálový obrázek, na němž následně lze aplikovat výše popsanou metodu *YOLO*.

Kapitola 3
Architektura

Náš detektor je rozšířením již zmíněného algoritmu *PointPillars*. Před spuštěním *PointPillars* se nejdříve pokoušíme o synchronizaci obrázkových dat a příslušných mračen bodů za pomoci interpolační neuronové sítě *Real-time Intermediate Flow Estimation (RIFE)*[21]. Na interpolovaných snímcích dále používáme detekci 2D objektů sítí *YOLOv5*. Tyto detekce zakomponujeme do vstupního kanálu sítě *PointPillars* a na její výstupy dále aplikujeme metody asociace detekcí v po sobě jdoucích snímcích pro odhad vektorů rychlostí vozidel ve scéně.

3.1 Zpracování vstupních dat

Jak již bylo zmíněno v předešlém textu, naše senzory nejsou vůči sobě nijak časově synchronizovány. Roboty, které používáme, pracují v systému *Robot Operating System (ROS)*, který poskytuje vývojářsky pohodlné prostředí pro ovládání robotů a zpracování dat. Základním principem je rozdělení programové logiky do jednotlivých modulů (*node*), které spolu komunikují pomocí zpráv. Stejně tak i data ze senzorů jsou posílána zprávami a *node*, který data zpracovává, tyto zprávy čte. Každá zpráva v systému je zároveň opatřena časovou značkou, kdy byla odeslána, ovšem v našich robotech nemají kamery synchronizované hodiny se zbytkem systému, a proto se časové značky dat nedají brát jako skutečný čas pořízení snímku. To vyvolává potřebu provést alespoň základní synchronizaci a zpracování dat obdržených z kamer a lidaru.

Navrhujeme nasadit interpolační síť *RIFE*, která dovede vygenerovat libovolnou interpolaci mezi dvěma snímky video sekvence. Modely *optical flow* nelze pro interpolaci přímo použít, protože k aproximaci *optical flow* mezi interpolovaným a vstupními snímky by bylo potřeba znát i interpolovaný snímek, který zřejmě z podstaty problému není k dispozici[21]. Abychom věděli, jaký časový okamžik podle hodin lidarového snímače odpovídá hodinám dané kamery a jaké dva video snímky máme použít pro získání interpolace ve správném časovém okamžiku, musíme nejdříve přibližně identifikovat časový posun hodin kamer od zbytku systému. Toho jsme docílili zkoušením různých posunů a interpolací dle těchto posunů. Následně jsme vždy promítli několik anotovaných 3D bounding boxů do interpolovaných snímků, dokud jsme nenašli posun, za jehož použití byla interpolace vizuálně správná. Pro přední kameru je tento posun přibližně $0,47$ sekundy. S tímto posunem jsme interpolovali snímky z kamer, abychom získali přibližné obrázky odpovídající časovým značkám všech lidarových snímků v datasetu. Příklad takové interpretace, včetně původních snímků, ukazují obrázky 3.1, 3.2 a 3.3.

Na takto interpolované snímky dále aplikujeme detektor *YOLOv5* a získáváme tak 2D bounding boxy objektů, které lze snadněji použít jako další vstupní kanál v 3D detektoru. Síť *YOLOv5* jsme předem natrénovali na našich anotovaných 2D datech.

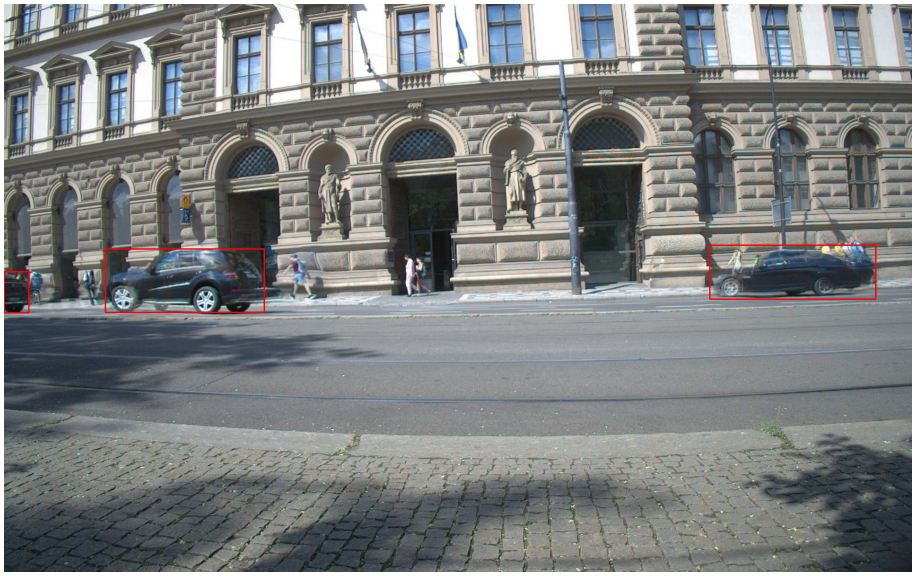
3.2 Algoritmus 3D detekce

Obrázek 3.1. Nejbližší snímek před potřebnou časovou značkou interpolace.

Obrázek 3.2. Nejbližší snímek po potřebné časové značce interpolace.

Obrázek 3.3. Ukázka interpolovaného snímku pomocí sítě *RIFE*. Jednotlivé snímky z kamery přichází s periodou přibližně 200 milisekund. Potřebná časová značka je v tomto případě přibližně v 52 % cesty mezi obrázky. Obrázek podléhá drobnému zkreslení.

Obrázek 3.4. Ukázka 2D detekce YOLOv5 na interpolovaném snímku pomocí sítě *RIFE*.



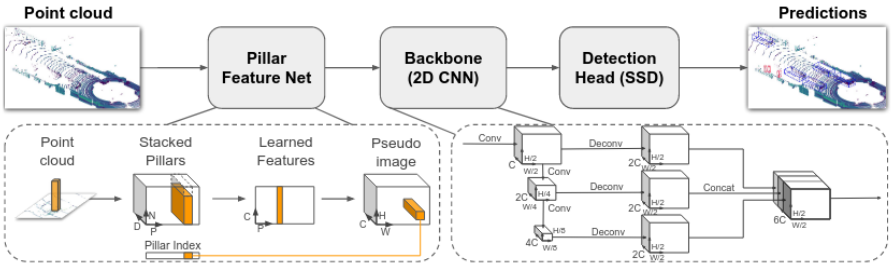
Obrázek 3.5. Ukázka 2D detekce YOLOv5 na interpolovaném snímku pomocí sítě *RIFE* obsahujícím zkreslení objektů. Kvalita detekce není zkreslením výrazně ovlivněna.

Pro 3D detekci využíváme již zmíněného algoritmu *PointPillars*. 2D detekce získané v minulém kroku používáme jako rozšíření dimenze bodů o jednu složku v mračnech bodů: kladné číslo odpovídající *confidence* 2D detektoru detekované třídy. Toto řešení se nabízí přirozeně, protože i původní metoda *PointPillars* využívá jako vstup rozšířená mračna bodů o složku reflektivity. Přesněji řečeno z původní množiny bodů

$$P_{in} \subseteq \mathbb{R}^{3+c},$$

kde v našem případě $c = 1$ a odpovídá složce reflektivity, vytvoříme množinu rozšířených bodů

$$P_{out} = \{\phi(p) \mid p \in P_{in}\},$$
$$\phi(p) = (p_1, \ldots, p_{3+c}, a)^T \in \mathbb{R}^{4+c},$$



Obrázek 3.6. Diagram architektury metody *PointPillars*, která je stěžejním článkem našeho detektoru. Vstupní data sítě tvoří mračna bodů obohacená o další složku odpovídající 2D detekcím. Tato data jsou rozdělena do mřížky sloupců a z nich je vytvořen 3D tensor (pseudo-obrázek). Na pseudo-obrázek jsou dále aplikovány konvoluční operace a vzniká tak příznaková mapa, kterou detekční hlavice *SSD* používá společně s apriorními anchor boxy k regresi bounding boxů. Obrázek je převzat z [3].

kde a je součet všech hodnot *confidence* bounding boxů 2D detekcí detekované třídy, do kterých se promítne bod p. V původní síti tedy měníme první vrstvu, která má za úkol vytvořit vektorovou reprezentaci každého sloupce bodů, tak, aby mohla přijmout výše popsaná rozšířená mračna bodů.

Další změnou jsou anchor boxy, které v původní metodě odpovídají datasetu *KITTI* a dobře na něm fungují. Vzhledem k odlišným velikostem bounding boxů našich dat jsme přepočítali velikosti anchor boxů, aby odpovídaly průměrnému rozměru bounding boxů v našem datasetu. Stejně jako v originální implementaci bereme tento anchor box ve dvou rotacích: 0 a $\frac{\pi}{2}$ radiánů.

Dále ze sítě odstraňujeme třídy *cyklistů* a *chodců*, protože v našem datasetu se *cyklisté* nenachází a detekce *chodců* není v kontextu řešeného problému relevantní.

3.3 Zpracování výstupů detektoru

Pro odhad rychlostí detekovaných vozidel při nasazení na testovacích datech využíváme metodu asociace objektů v po sobě jdoucích snímcích na základě jejich blízkosti. Předpokládejme, že v čase t máme množinu m detekovaných bounding boxů

$$B_t = \{b_t^1, \ldots, b_t^m\}$$

a v čase $t+1$ máme množinu n detekovaných bounding boxů

$$B_{t+1} = \{b_{t+1}^1, \ldots, b_{t+1}^n\}$$

a dále máme funkci

$$f_t \colon B_t \times B_{t+1} \to \mathbb{R},$$

která dvěma bounding boxům přiřadí tím větší číslo, čím „pravděpodobnější“ je, že patří stejnému skutečnému objektu. Pak můžeme uvažovat následující matici

$$C = \begin{pmatrix} f_t(b_t^1, b_{t+1}^1) & \cdots & f_t(b_t^1, b_{t+1}^n) \\ \vdots & \ddots & \vdots \\ f_t(b_t^m, b_{t+1}^1) & \cdots & f_t(b_t^m, b_{t+1}^n) \end{pmatrix},$$

kde prvek v i-tém řádku a j-tém sloupci $C_{i,j}$ říká, jak dobře odpovídá box b_t^i boxu b_{t+1}^j, a problém nalezení asociace bounding boxů ztotožnit s následujícím lineárním

programem

$$X^* = \arg\max_X \sum_{i=1}^{m}\sum_{j=1}^{n} C_{i,j}X_{i,j}$$

za podmínek

$$\forall j: \sum_{i=1}^{m} X_{i,j} \leq 1,$$
$$\forall i: \sum_{j=1}^{n} X_{i,j} \leq 1,$$
$$\sum_{i=1}^{m}\sum_{j=1}^{n} X_{i,j} = \min(m,n),$$
$$\forall i,j: X_{i,j} \in \{0,1\}.$$

Tento program je dobře známý a řešitelný v polynomiálním čase *Munkres-Kuhnovým* algoritmem[22].

Konkrétně jako funkci f_t volíme metriku *intersection over union (IoU)* 3D bounding boxů. Dále *Munkres-Kuhnovým* algoritmem najdeme optimální přiřazení X^* a v případě, že některým přiřazeným bounding boxům odpovídá IoU menší než zvolený práh, pak uvažujeme tyto bounding boxy jako nepřiřazené.

Když máme nalezené přiřazení bounding boxů v po sobě jdoucích snímcích, je odhad vektorů rychlostí již jednoduchý. Vycházíme ze vztahu

$$\vec{v} = \frac{\overrightarrow{\Delta p}}{\Delta t},$$

kde $\overrightarrow{\Delta p}$ je posun středu bounding boxu mezi snímky a Δt je doba uplynulá mezi snímky.

Kapitola 4
Použité datasety

Referenčním datasetem pro naše experimenty je často využívaný dataset *KITTI*[23], jelikož i autoři *PointPillars*[3] ho svých experimentech použili. Dále jsme pořídili vlastní dataset za pomoci robotů katedry kybernetiky, který jsme také anotovali.

4.1 Dataset KITTI

Srovnávací sada pro vidění autonomních agentů *KITTI*[23][24] byla zveřejněna v roce 2012 Technologickým institutem v Karlsruhe a Toyota technologickým institutem v Chicagu pro účely srovnání metod 3D detekce objektů a jejich lokalizace a lepší možnosti aplikace těchto metod v reálném světě. Natáčecí vozidlo snímalo okolí při průjezdu středně velkým městem Karlsruhe. Sensory použité při pořizování sady *KITTI* jsou uvedeny v tabulce 4.1.

počet	typ	název modelu	vlastnosti
1	inerciální navigační systém	OXTS RT 3003	–
1	lidarový snímač	Velodyne HDL-64E	10 Hz, 64 paprsků
2	kamera ve stupních šedi	Point Grey Flea 2 (FL2-14S3M-C)	10 Hz, 1392 × 512 pixelů
2	barevná kamera	Point Grey Flea 2 (FL2-14S3C-C)	10 Hz, 1392 × 512 pixelů
4	varifokální čočka	Edmund Optics NT59-917	–

Tabulka 4.1. Seznam senzorů a příslušenství použitých při pořizování sady *KITTI*

Předpřipravený anotovaný dataset pro 3D detekci obsahuje 7481 trénovacích snímků barevné kamery a jim příslušná mračna bodů, které se při trénování detektoru dále dělí na trénovací a validační data. Přítomné třídy a jejich celkové zastoupení v datové sadě ukazuje tabulka 4.2. Adresářovou strukturu trénovací a testovací množiny dat ukazuje obrázek 4.1. Testovací adresář neobsahuje soubory anotací objektů, proto ho během trénování detektoru, ani jeho kvantitativním vyhodnocení nelze použít. Objekty jsou označeny, pouze pokud jsou v zorném poli kamery.

4.1.1 Formát souboru s anotacemi objektů

Ke každému snímku trénovacího datasetu patří jeden soubor v textovém formátu obsahující anotace všech objektů v tomto snímku. Každý řádek odpovídá jednomu

```
training/                    testing/
├── calib                    ├── calib
│   ├── 000000.txt           │   ├── 000000.txt
│   ├── ...                  │   ├── ...
│   └── 007480.txt           │   └── 007517.txt
├── image_2                  ├── image_2
│   ├── 000000.png           │   ├── 000000.png
│   ├── ...                  │   ├── ...
│   └── 007480.png           │   └── 007517.png
├── label_2                  └── velodyne
│   ├── 000000.txt               ├── 000000.bin
│   ├── ...                      ├── ...
│   └── 007480.txt               └── 007517.bin
└── velodyne
    ├── 000000.bin
    ├── ...
    └── 007480.bin
```

Obrázek 4.1. Struktura testovacího a trénovacího adresáře datasetu *KITTI*.

název třídy	počet výskytů	průměrná výška [m]	průměrná šířka [m]	průměrná délka [m]
Car	**28742**	**1,53**	**1,63**	**3,88**
Cyclist	1627	1,74	0,6	1,76
Pedestrian	4487	1,76	0,66	0,84
Truck	1094	3,25	2,59	10,11
Van	2914	2,21	1,9	5,08
Tram	511	3,53	2,54	16,09
Misc	973	1,91	1,51	3,58
Person_sitting	222	1,27	0,59	0,8

Tabulka 4.2. Seznam tříd a průměrné rozměry bounding boxů v trénovací sadě *KITTI* pro 3D detekci objektů. Zvýrazněné jsou třídy, které náš detektor uvažuje.

objektu a obsahuje 15, případně 16 polí oddělených mezerou. Tabulka 4.3 blíže popisuje význam jednotlivých polí.

4.2 Náš dataset

Důvodů, proč je výhodné využít nově vytvořeného datasetu, a ne některého již existujícího, je několik. Kamery, resp. lidary, které byly použity při snímání datové sady *KITTI*, byly umístěny ve výšce $1,65$, resp. $1,73$ metrů nad zemí, což je o dost více než u robotů katedry kybernetiky, na které je naše aplikace směřována. Navíc dataset *KITTI* používá lidar *Velodyne*, kdežto naše roboty disponují lidarem *Ouster OS0*, který má jiné vlastnosti. Dále musíme brát v úvahu, že v našem prostředí se vyskytují jiná vozidla a objekty s jinými apriorními pravděpodobnostmi, takže náš rozhodovací systém musí být jiný než pro jiné datasety. Proto jsme nasbírali a anotovali náš vlastní dataset, který bude reprezentativnější pro naši aplikaci.

Pořizování proběhlo na Karlově náměstí v Praze na dvou místech poblíž frekventované křižovatky za denního světla a dobré viditelnosti 2.8.2022 a byly použity dva roboty, které se během natáčení téměř nehýbaly. Celkem se tak podařilo zachytit přibližně osm relativně podobných scén s mnoha automobily. Data nejsou kompletně anotována, ale

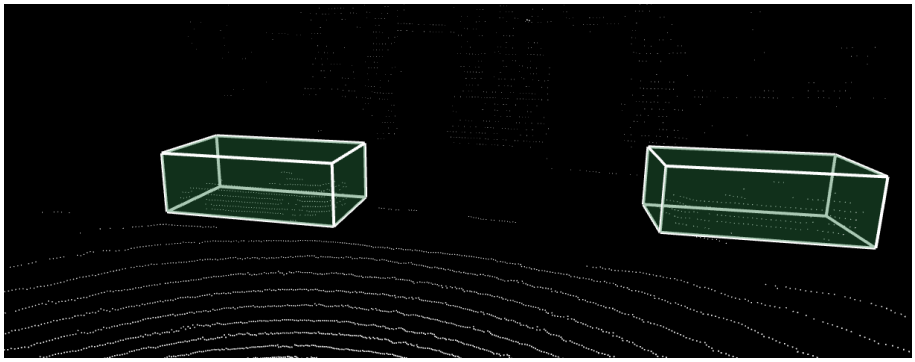
název pole	popis	nabývané hodnoty
třída	název třídy objektu	textový řetězec označující třídu
useknutost	desetinné číslo udávající, jak velká část objektu je mimo kameru	$[0, 1)$
zakrytí	celé číslo označující, v jaké míře je objekt zakryt	$\{0, 1, 2, 3\}$
alfa	úhel mezi středem objektu a osou z v souřadnicovém systému kamery	$[-\pi, \pi)$
left	pixelová souřadnice x levé hrany 2D bounding boxu	$\mathbb{R}$
top	pixelová souřadnice y horní hrany 2D bounding boxu	$\mathbb{R}$
right	pixelová souřadnice x pravé hrany 2D bounding boxu	$\mathbb{R}$
bottom	pixelová souřadnice y spodní hrany 2D bounding boxu	$\mathbb{R}$
výška	výška 3D bounding boxu	$\mathbb{R}$
šířka	šířka 3D bounding boxu	$\mathbb{R}$
délka	délka 3D bounding boxu	$\mathbb{R}$
x	x-ová souřadnice středu 3D bounding boxu v souřadnicovém systému kamery	$\mathbb{R}$
y	y-ová souřadnice středu 3D bounding boxu v souřadnicovém systému kamery	$\mathbb{R}$
z	z-ová souřadnice spodní stěny 3D bounding boxu v souřadnicovém systému kamery	$\mathbb{R}$
rotace	úhel svíraný mezi přední osou 3D bounding boxu objektu a osou y v souřadnicovém systému kamery	$[-\pi, \pi)$
skóre	nepovinné pole obsahující jistotu detekovaného bounding boxu	$(0, 1]$

Tabulka 4.3. Seznam polí v každém souboru anotací v trénovací sadě.

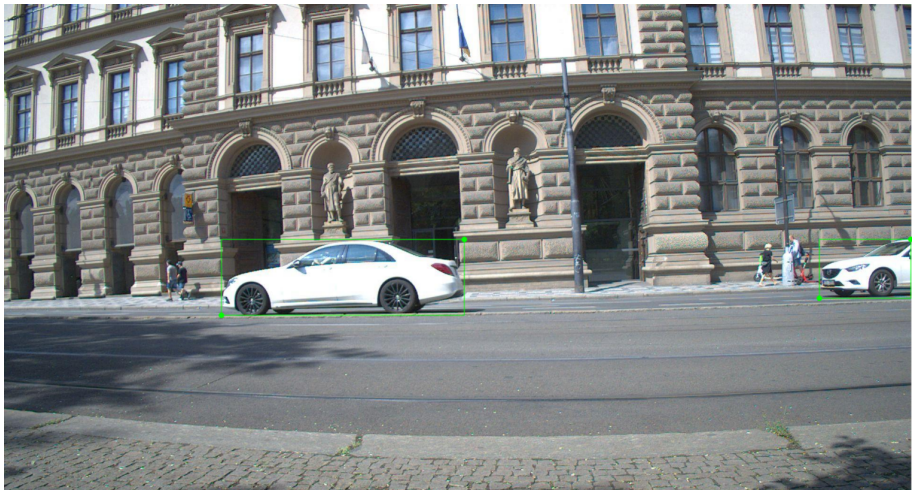
současné statistické údaje anotací 2D bounding boxů v obrázcích, resp. 3D bounding boxů v mračnech bodů obsahují tabulky 4.4, resp. 4.5. Pro anotaci 2D bounding boxů, na které se také podílela studentka Wing Yan Winnie So, jsme použili anotační nástroj *labelme* [25]. Pro anotaci 3D bounding boxů jsme využili webový anotační nástroj *Computer Vision Annotation Tool (CVAT)* [26].

Anotace následně převádíme do formátu kompatibilního s *KITTI*, což obnáší vytvoření podobné adresářové struktury, jakou ukazuje 4.1. Obsah textových souborů s anotacemi je strukturálně stejný jako v *KITTI*, jak popisuje tabulka 4.3.

Nejen že narozdíl od sady *KITTI* nejsou naše kamery a lidar přesně časově synchronizovány, ale také snímají s jinou frekvencí, a proto jsme anotaci obrázků a



Obrázek 4.2. Obrázek vizualizuje lidarový snímek z našeho datasetu obsahující dva anotované bounding boxy aut.



Obrázek 4.3. Obrázek vizualizuje dva anotované 2D bounding boxy v obrázku zachyceném přední kamerou robota zobrazující stejná vozidla jako ukázka 4.2.

3D bodů provedli zvlášť. Z tohoto důvodu také není triviální na našich datech spustit již navržené 3D multimodální detektory, protože ty většinou předpokládají přesnou korespondenci RGB dat a 3D bodů.

název třídy	počet výskytů	průměrná výška [px]	průměrná šířka [px]
Car	6717	185,13	96,44

Tabulka 4.4. Seznam tříd a průměrné rozměry 2D bounding boxů v naší trénovací sadě.

název třídy	počet výskytů	průměrná výška [m]	průměrná šířka [m]	průměrná délka [m]
Car	1148	1,89	2,43	5,05

Tabulka 4.5. Seznam tříd a průměrné rozměry 3D bounding boxů v naší trénovací sadě.

Kapitola 5
Experimenty

Naše evaluační experimenty byly uskutečněny na srovnávací sadě *KITTI*[23], jak byla popsána v sekci 4.1, a na naší sadě. Pro účely srovnání naší modifikace s původní metodou *PointPillars*[3] jsme nejdříve vyhodnotili její volně přístupnou implementaci[27]. Poté jsme natrénovali modifikovanou multimodální síť, která z ní vychází, a spustili evaluaci na ní.

Modifikovaný detektor jsme natrénovali na stejných datových sadách, ale navíc jsme použili 2D detektor na příslušných obrázcích pro rozšíření vstupních množin bodů podle sekce 3.

5.1 Evaluační metrika

Pro evaluaci 3D detekcí objektů existuje několik metrik. My budeme zejména používat metriku 3D AP (3D bounding box average precision), která je definována jako obsah pod PR (precision-recall) křivkou.

Precision definujeme jako podíl počtu pravdivých pozitivních detekcí a všech pozitivních detekcí:

$$p = \frac{TP}{TP + FP}$$

a *recall* jako podíl počtu pravdivých pozitivních detekcí a součtu počtů pravdivých pozitivních a falešných negativních detekcí:

$$r = \frac{TP}{TP + FN}.$$

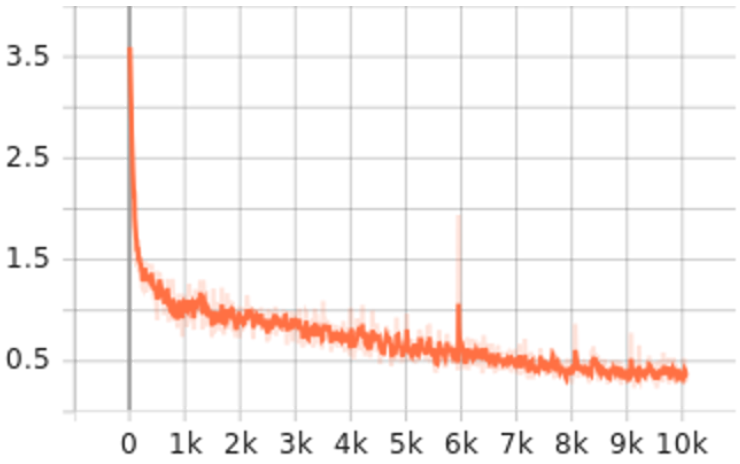
Precision tedy říká, kolik detekcí je pravdivých ze všech detekcí detektoru, a *recall*, kolik pravdivých detekcí detektor učinil ze všech skutečných příkladů. IoU dvou bounding boxů definujeme jako podíl objemu jejich průniku a jejich sjednocení. Detekované bounding boxy jsou označeny jako pravdivé, pokud podíl IoU 3D bounding boxů s anotací přesahuje práh $0,5$. V případě evaluace 2D detektoru nahradíme *3D IoU* metrikou *2D IoU*, kterou definujeme analogicky jako podíl obsahu průniku a sjednocení dvou bounding boxů.

PR (precision-recall) křivka vzniká vynesením hodnot *precision* a *recall* pro všechny hodnoty prahu skóre detekce do grafu, kde osa x představuje *recall* a osa y *precision*. Nízké hodnoty prahu, které způsobí větší množství výstupních bounding boxů, pak odpovídají bodům s vysokou hodnotou *recall* a nízkou hodnotou *precision* a vice versa. Metrika *3D bounding box average precision (3D AP)* je dána obsahem pod *PR* křivkou. Obdobně je definována *2D bounding box average precision (2D AP)*, pouze místo *3D IoU* používá *2D IoU*.

5.2 Trénování

Trénování sítě *YOLOv5* a upravené sítě *PointPillars* probíhá postupně a zvlášť. *YOLOv5* trénujeme po dobu 200 epoch na našem 2D datasetu rozděleném náhodně na trénovací a validační příklady. Ostatní parametry a hyperparametry sítě necháváme nepozměněné. Na validační sadě dosahuje plně natrénovaná síť *YOLOv5* *AP* 87,5 %. Obrázky 3.4 a 3.5 vizualizují 2D detekce na interpolovaných snímcích.

Modifikovanou síť *PointPillars* trénujeme na našich datech po dobu 160 epoch iterativní optimalizační metodou *AdamW*, kde účelovou funkci uvažujeme stejnou jako [3]. Obrázek 5.1 ukazuje vývoj ztrátové funkce modifikované metody *PointPillars*.



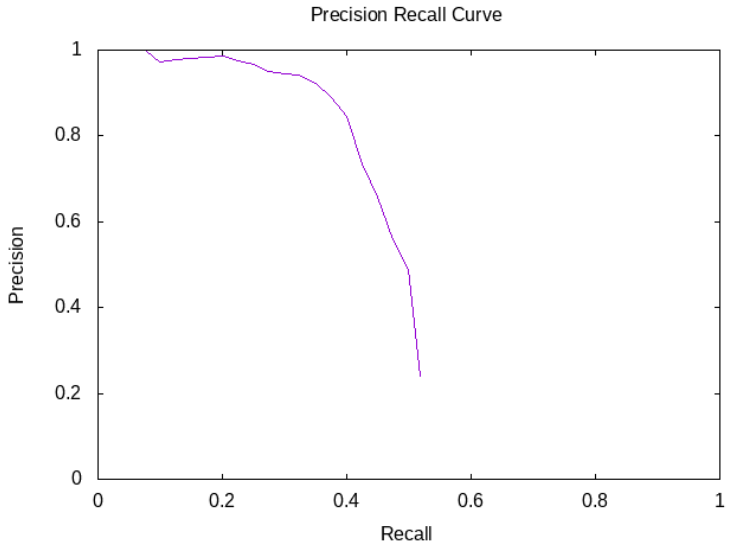
Obrázek 5.1. Vývoj trénovací ztrátové funkce v závislosti na počtu kroků optimalizátoru modifikované multimodální metody.

5.3 Evaluace

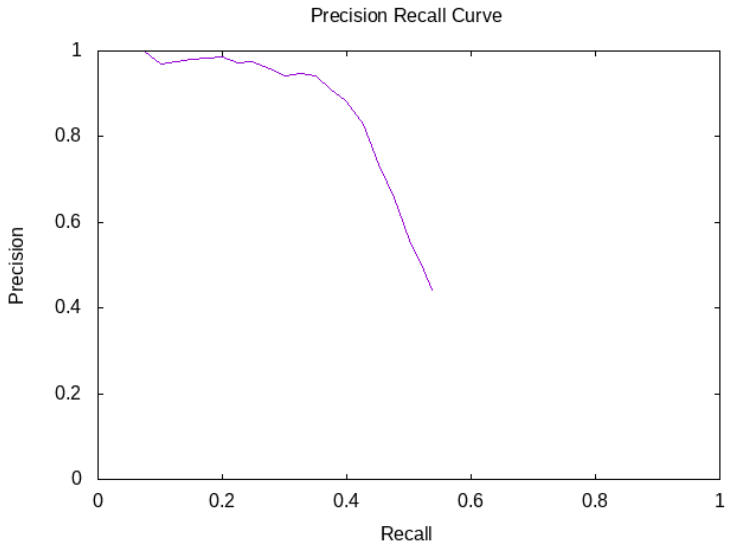
Kvantitativní výsledky evaluace výchozí a naší navržené metody shrnuje tabulka 5.1. Všimněme si, že kvalita detektoru při zvýšení minimálního počtu bodů v bounding boxech potřebného pro jejich uvažování významně nestoupla. Z toho lze soudit, že i z malého množství bodů je možné celkem spolehlivě detekovat vozidlo. Náš multimodální detektor dosahuje na našem datasetu o téměř 30 % lepší výsledek *3D AP* než původní metoda *PointPillars* samotná. *PR* křivky jednotlivých vyhodnocení vykreslují obrázky 5.2, 5.3, 5.4 a 5.5.

		3D AP [%]	
model	evaluační sada	počet bodů ≥ 1	počet bodů ≥ 10
PointPillars	Naše	47,7	48,7
Multimodální	Naše	**77,1**	**78,6**

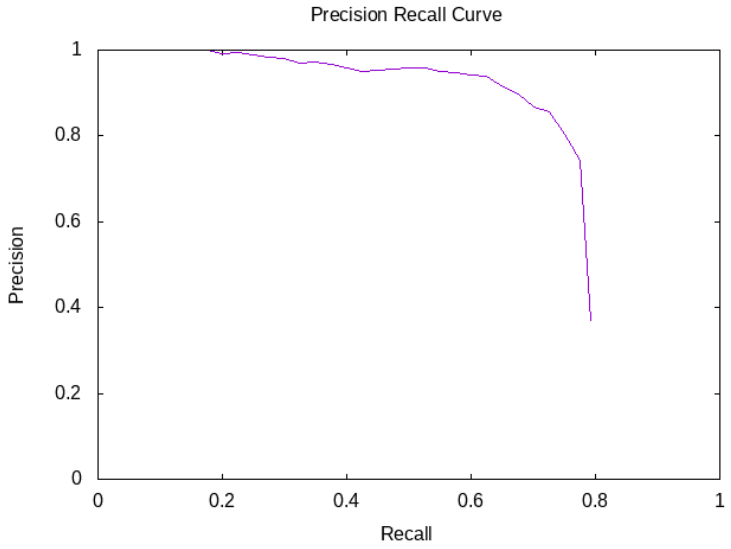
Tabulka 5.1. Výsledky evaluace metriky 3D AP obou detektorů natrénovaných na datasetech. Počet bodů znamená minimální počet bodů v skutečném bounding boxu, aby byl uvažován při evaluaci. Tedy čím nižší je minimální počet bodů, tím těžší příklady detekce jsou od detektoru vyžadovány.



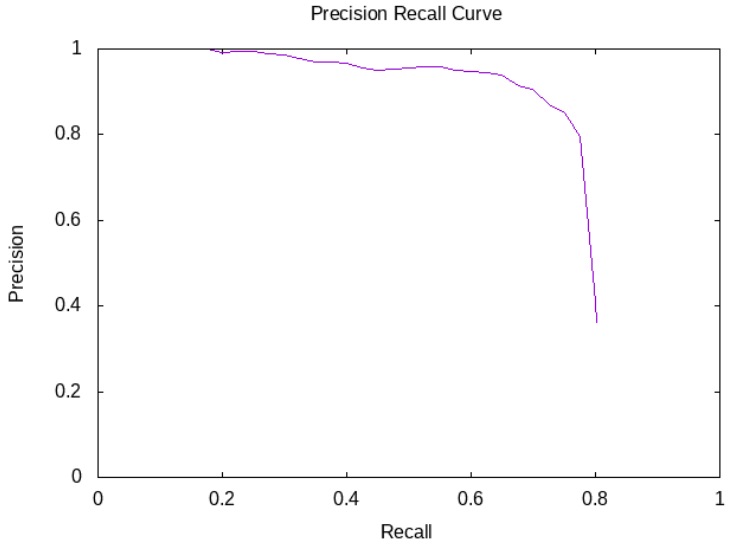
Obrázek 5.2. *PR* křivka metody *PointPillars* na našich datech s minimálním počtem bodů v bounding boxech 1.



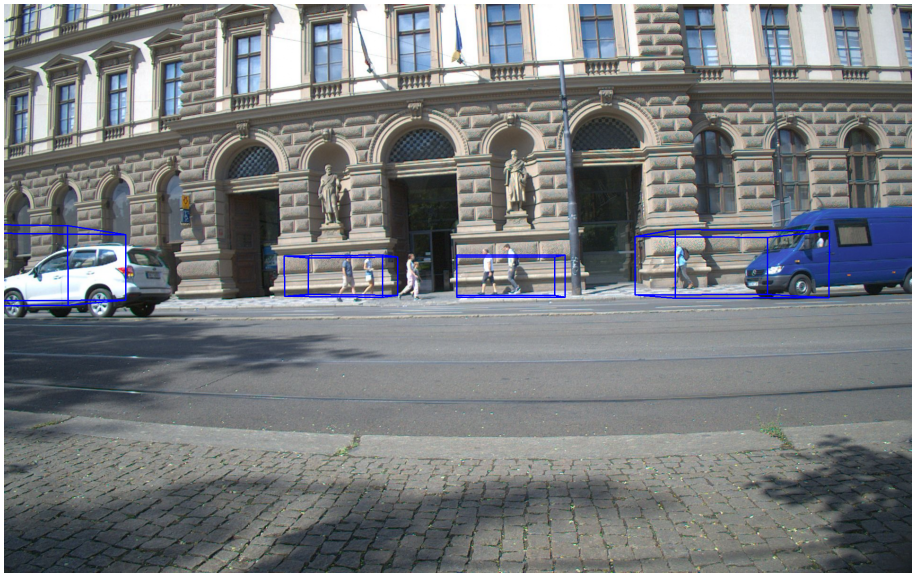
Obrázek 5.3. *PR* křivka metody *PointPillars* na našich datech s minimálním počtem bodů v bounding boxech 10.



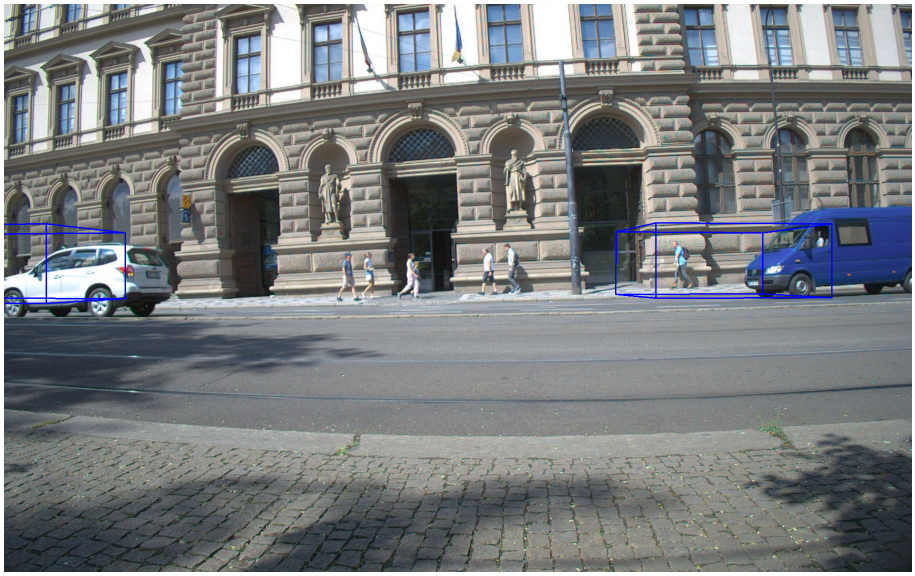
Obrázek 5.4. *PR* křivka naší metody na našich datech s minimálním počtem bodů v bounding boxech 1.



Obrázek 5.5. *PR* křivka naší metody na našich datech s minimálním počtem bodů v bounding boxech 10.



Obrázek 5.6. Detekce původní metody *PointPillars*. Na výstupu se vyskytují i falešně pozitivní detekce, ve kterých detektor označuje chodce za vozidla.



Obrázek 5.7. Detekce našeho detektoru na stejném snímku. Falešně pozitivní detekce se zde nevyskytují. Posuny bounding boxů u obou snímků jsou způsobeny tím, že lidar snímá sloupce bodů postupně a s časovým posunem, tedy obrázky jsou synchronizovány s mračnem bodů nejlépe uprostřed obrázku.

Kapitola 6
Závěr

6.1 Shrnutí

V této práci jsme nejdříve uvedli některé existující přístupy jak 2D, tak 3D detekce objektů z lidarových a kamerových dat a přiblížili, proč nejsou vhodné pro naši aplikaci na mobilních robotech bez příslušných modifikací. Představili jsme také hojně využívaný dataset *KITTI* pro srovnání metod 3D detekce a vysvětlili nutnost anotace vlastního datasetu z našeho prostředí. Dále jsme navrhli multimodální metodu pro detekci 3D objektů stavějící na algoritmu *PointPillars* a provedli experimentální srovnání, které ukázalo, že náš detektor na našich datech dosahuje lepších výsledků.

6.2 Další možnosti

V navržené metodě se nabízí několik míst pro zlepšení nebo další zkoumání, která jsou mimo rozsah této práce. Jedním z nich je rozdělení anotovaných tříd *car* na *car* a *truck*. Rozdělení vozidel do různých tříd podle velikosti, případně jiných vlastností, by potenciálně mohlo zlepšit výsledky detekce. Další možností pro zlepšení odhadu rychlostí a také lokalizace samotné je použití některých metod *trackingu* objektů jako jsou filtry a využití apriorní znalosti fyzikálního modelu scény. V podobném duchu uvádíme možnost integrace výsledků detekce předchozího snímku do vstupů detektoru při inferenci na následujícím snímku. V neposlední řadě by se při detekci dalo využít znalosti okolního prostředí mobilního robota v nasazení. Například by se detektor mohl více zaměřovat na místa, kde je silnice, a přikládat detekcím podle jejich lokality různou váhu. Další práce by také mohla zahrnovat rozšíření našeho datasetu o další anotace.

Literatura

[1] Yin Zhou a Oncel Tuzel. *VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection*. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018. 4490-4499.

[2] Yan Yan, Yuxing Mao a Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*. 2018, 18 (10). DOI 10.3390/s18103337.

[3] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang a Oscar Beijbom. *PointPillars: Fast Encoders for Object Detection From Point Clouds*. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019. 12689-12697.

[4] P. Viola a M. Jones. *Rapid object detection using a boosted cascade of simple features*. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001. I-I.

[5] Yoav Freund a Robert E Schapire. A Decision-Theoretic Generalization of On--Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*. 1997, 55 (1), 119-139. DOI https://doi.org/10.1006/jcss.1997.1504.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard a L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*. 1989, 1 (4), 541-551. DOI 10.1162/neco.1989.1.4.541.

[7] Y. Lecun, L. Bottou, Y. Bengio a P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998, 86 (11), 2278-2324. DOI 10.1109/5.726791.

[8] Alex Krizhevsky, Ilya Sutskever a Geoffrey Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*. 2012, 25 DOI 10.1145/3065386.

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg a Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*. 2015, 115 (3), 211-252. DOI 10.1007/s11263-015-0816-y.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li a Li Fei-Fei. *ImageNet: A large-scale hierarchical image database*. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009. 248-255.

[11] Matthew D Zeiler a Rob Fergus. *Visualizing and Understanding Convolutional Networks*. 2013.

[12] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus a Yann LeCun. *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks*. 2014.

[13] Karen Simonyan a Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell a Jitendra Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014.

[15] Ross Girshick. *Fast R-CNN*. 2015.

[16] Shaoqing Ren, Kaiming He, Ross Girshick a Jian Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016.

[17] Joseph Redmon, Santosh Divvala, Ross Girshick a Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. 2016.

[18] Joseph Redmon a Ali Farhadi. *YOLO9000: Better, Faster, Stronger*. 2016.

[19] R. Qi Charles, Hao Su, Mo Kaichun a Leonidas J. Guibas. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. 77-85.

[20] Waleed Ali, Sherif Abdelkarim, Mohamed Zahran, Mahmoud Zidan a Ahmad El Sallab. *YOLO3D: End-to-end real-time 3D Oriented Object Bounding Box Detection from LiDAR Point Cloud*. 2018.

[21] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi a Shuchang Zhou. *Real--Time Intermediate Flow Estimation for Video Frame Interpolation*. 2022.

[22] Harold W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*. 1955, 2 83–97.

[23] Andreas Geiger, Philip Lenz a Raquel Urtasun. *Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite*. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[24] A Geiger, P Lenz, C Stiller a R Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*. 2013, 32 (11), 1231–1237. DOI 10.1177/0278364913491297.

[25] Kentaro Wada. *labelme: Image Polygonal Annotation with Python*. `https://github.com/wkentaro/labelme`. 2018.

[26] Boris Sekachev, Nikita Manovich, Maxim Zhiltsov, Andrey Zhavoronkov, Dmitry Kalinin, Ben Hoff, TOsmanov, Dmitry Kruchinin, Artyom Zankevich, DmitriySidnev, Maksim Markelov, Johannes222, Mathis Chenuet, a-andre, telenachos, Aleksandr Melnikov, Jijoong Kim, Liron Ilouz, Nikita Glazov, Priya4607, Rush Tehrani, Seungwon Jeong, Vladimir Skubriev, Sebastian Yonekura, vugia truong, zliang7, lizhming a Tritin Truong. *opencv/cvat: v1.1.0*. 2020.
`https://doi.org/10.5281/zenodo.4009388`.

[27] zhulf0804. *PointPillars*.
`https://github.com/zhulf0804/PointPillars`. 2022.

Příloha A
Kód implementace algoritmu

Kód hlavní sítě, který byl z většiny převzat z repozitáře [27] a upraven, je k dispozici na adrese `https://gitlab.fel.cvut.cz/tilhovoj/pointpillarsyolo`. Náš použitý dataset včetně anotací je přístupný na odkazu `https://drive.google.com/drive/folders/19z4QehbngfwZyOokxtmB53alMVyGCCLq?usp=share_link`.