



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**DETEKCE OBJEKTŮ V LASEROVÝCH SKENECH PO-
MOCÍ KONVOLUČNÍCH NEURONOVÝCH SÍTÍ**

OBJECT DETECTION IN THE LASER SCANS USING CONVOLUTIONAL NEURAL NETWORKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL ZELENÁK

VEDOUcí PRÁCE

SUPERVISOR

Ing. MARTIN VEĽAS,

BRNO 2020

Zadání bakalářské práce



Student: **Zelenák Michal**
Program: Informační technologie
Název: **Detekce objektů v laserových skenech pomocí konvolučních neuronových sítí**
Object Detection in the Laser Scans Using Convolutional Neural Networks
Kategorie: Umělá inteligence

Zadání:

1. Zorientujte se v problematice počítačového vidění, v robotice a v základních technikách laserového skenování a senzoru Velodyne LIDAR.
2. Seznamte se s konceptem konvolučních neuronových sítí se zaměřením na detekci objektů a sémantickou segmentaci.
3. Vyberte vhodné metody a nástroje, navrhnete neuronovou síť pro detekci vybrané skupiny objektů (např. přírodní objekty, silnice, vozidla, a pod.).
4. Navržené řešení implementujte s využitím existujících frameworků pro hluboké učení.
5. Experimentujte s vaší implementací a případně navrhnete vlastní modifikace metod.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění prvních tří bodů zadání.

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

Vedoucí práce: **Veřás Martin, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 31. července 2020
Datum schválení: 1. listopadu 2019

Abstrakt

Táto práca je zameraná na segmentáciu vozovky v laserových skenoch, pomocou konvolučnej neurónovej siete. Pre riešenie uvedeného problému, ktorý nájde uplatnenia v oblastiach údržby vozovky, boli použité konvolučné neurónové siete pre ich flexibilitu a rýchlosť. Práca prináša implementáciu a modifikácie existujúcej metódy, ktorá daný problém rieši pomocou plne prepojenej konvolučnej neurónovej siete. Týmito modifikáciami sú napríklad použitie rôznych parametrov pre chybovú funkciu, použitie iného počtu tried v modeli a datasete. Vplyv modifikácií bol experimentálne overený a bola dosiahnutá presnosť 96.12%, a hodnota F-measure 95.02%.

Abstract

This work is focused on road segmentation in laser scans, using a convolutional neural network. To achieve this goal, which will find application in the field of road maintenance, convolutional neural networks have been used for their flexibility and speed. The work brings implementation and modifications of the existing method, which solves the problem by using a fully connected convolutional neural network. Used modifications include, for example using of various parameters for the loss function, the use of a different number of classes in the network model and dataset. The effect of the modification was experimentally verified and the accuracy of 96.12%, and the value for F-measure 95.02% were achieved.

Kľúčové slová

Neurónová sieť, konvolučná neurónová sieť, LIDAR, cesta, vozovka, Velodyne, KITTI, PyTorch, segmentácia

Keywords

Neural network, Convolutional neural network, LIDAR, road, roadway, Velodyne, KITTI, PyTorch, segmentation

Citácia

ZELENÁK, Michal. *Detekce objektů v laserových skenech pomocí konvolučních neuronových sítí*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Velas,

Detekce objektů v laserových skenech pomocí konvolučních neuronových sítí

Prehlásenie

Prehlasujem, že som túto bakalárskou prácu vypracoval samostatne pod vedením pána Ing. Martina Velasa. Ďalšie informácie mi poskytli Luca Caltagirone z univerzity Chalmers University of Technology. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Michal Zelenák
31. júla 2020

Podakovanie

Chcem poďakovať pánovi Ing. Martinovi Velasovi za všetky odborné konzultácie a rady poskytnuté pre riešenie práce, a taktiež jednému z autorov pôvodnej metódy Luca Caltagironemu, ktorý mi poskytol informácie a dáta ku danej metóde.

Obsah

1	Úvod	3
2	Technológia laserového mapovania	4
2.1	Technológia LiDAR	5
3	Neurónová sieť	8
3.1	Trénovanie neurónovej siete	12
3.2	Typy vrstiev v neurónových sieťach	15
4	State of the art	20
4.1	Príklad metód používajúcich kombináciu kamerových snímok a laserových skenov	20
4.2	Príklad metód používajúcej iba kamerové snímky	21
4.3	Príklad metódy používajúcej laserové skeny	21
5	Návrh systému	22
5.1	Použité dáta	22
5.2	Pretvorenie mračna bodov na tvar vhodný pre vstupnú vrstvu plne prepoje- nej konvolučnej neurónovej siete	23
5.3	Zväčšenie datasetu použitého pre neurónovú sieť	23
5.4	Použitý model neurónovej siete	28
6	Implementácia	30
6.1	Príprava a spracovanie datasetu	30
6.2	Trénovanie neurónovej siete	31
6.3	Použitie natrénovanej neurónovej siete	32
7	Experimenty	33
7.1	Rozdelenie datasetu pre tréovanie a testovanie	33
7.2	Určenie presnosti neurónovej siete	34
7.3	Vytvorenie primitívneho klasifikátora	35
7.4	Výsledok prvého behu	36
7.5	Experimenty s parametrami <code>weight</code> a <code>reduction</code> chybovej funkcie typu <code>CrossEntropyLoss</code>	36
7.6	Experimenty s počtom použitých tried	37
7.7	Najlepšie dosiahnuté výsledky tréovania neurónovej siete	38
7.8	Zhodnotenie výsledkov dosiahnutých experimentmi	43
8	Záver	45

Literatúra	46
A Obsah přiloženého paměťového média	50
B Manuál	51

Kapitola 1

Úvod

Teória strojového učenia predbehla technológiu svojej doby, a preto nebolo možné tieto poznatky aplikovať. V dnešnej dobe je konečne možné začať uplatňovať tieto znalosti, nakoľko v posledných desiatkach rokov dosiahol technologický pokrok v oblasti výpočetnej sily obrovský posun. V tejto práci je použitá časť teórie strojového učenia, a to konkrétne poznatky o konvolučných neurónových sieťach, ktoré sa v poslednej dobe stali akýmsi módnym trendom v oblasti spracovania dát.

Táto práca sa zaoberá detekciou objektov v laserových skenoch pomocou konvolučných neurónových sietí. Jednou zo základných entít v mestskom prostredí a doprave je vozovka, ktorej segmentácia nachádza široké možnosti použitia v rôznych oblastiach. Práve preto bola táto práca zameraná konkrétne na segmentáciu vozovky. Samotná segmentácia vozovky je náročná úloha, nakoľko táto činnosť vyžaduje veľmi zložitý algoritmus. Jednoduchší prístup k tejto úlohe, je použitie konvulčnej neurónovej siete, s ktorej použitím odpadá veľa problémov s vytvorením algoritmu, ktorý by dokázal rozpoznať vozovku, no na druhú stranu je potrebné vyriešiť problémy s implementáciou a natrénovaním tejto neurónovej siete.

Problém segmentácie vozovky je neustále riešený veľkým množstvom odborníkov, pričom niektoré z dostupných riešení sú popísané v kapitole 4. Použité riešenie v práci bolo inšpirované práve existujúcimi riešeniami. Práca prináša modifikácie pôvodnej metódy Caltagirone et al. [7], ako napríklad zmena parametrov učenia, zmena parametrov chybovej funkcie, či rôzne zmeny použitých tried v datasete a modeli neurónovej siete.

Použitie výsledku práce je možné nájsť pri rôznych prácach na vozovke, akými sú napríklad maľovanie dopravného značenia na vozovku, či údržba vozovky a jej okolia alebo rôzne aplikácie meraní plôch vozovky a jej okolia.

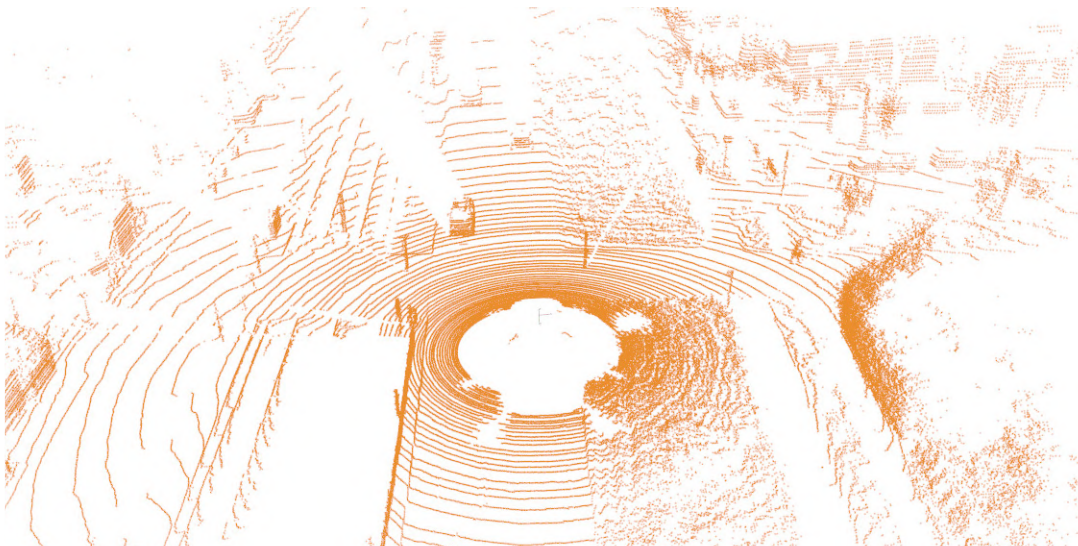
V kapitole 2 sa nachádza teoretický úvod k laserovým dátam, ktoré sú používané ako vstup konvulčnej neurónovej siete. Stručným spôsobom je tu popísaný spôsob získavania laserových dát a ich ukladania. Nasledujúca kapitola 3 popisuje teóriu potrebnú pre riešenie problému segmentácie vozovky pomocou konvulčnej neurónovej siete. Sú tu popísané vybrané parametre neurónových sietí a princípy ich tréningovania. Kapitola 5 predstavuje použité princípy pre riešenie problému. Nachádzajú sa tu informácie o použitej neurónovej sieti, ako aj teoretické poznatky na prípravu dát vhodných na tréningovanie neurónovej siete. V kapitole 6 sa nachádzajú informácie k príprave datasetu použitého pre tréningovanie neurónovej siete, ako aj informácie o samotnom prevedení tréningovania neurónovej siete. Kapitola 7 popisuje výsledky dosiahnuté pri zmenách určitých parametrov pre prípravu vstupných dát, parametrov tréningovaného modelu neurónovej siete alebo parametrov samotného tréningovania. V poslednej kapitole 8 sa nachádza zhrnutie dosiahnutých výsledkov v práci a informácie ohľadom možností, ktoré by mohli zlepšiť dosiahnuté výsledky.

Kapitola 2

Technológia laserového mapovania

Laserové mapovanie je technológia pre snímanie svojho prostredia alebo konkrétneho objektu. Táto technológia poskytuje presné a detailné informácie o snímanom prostredí. Pri mobilných systémoch LiDAR býva ich súčasťou aj GPS a INS jednotka. Časté sú taktiež systémy pre laserové mapovanie bez systémov GPS a INS, v ktorých sú mapované body popísané v lokálnych súradniciach. V posledných rokoch bol zaznamenaný zvýšený rast počtu vedeckých príspevkov s témou laserového mapovania, čo ukazuje, že laserové mapovanie sa stáva obľúbenou technológiou [31]. Obrovskou výhodou laserového mapovania oproti fotografovaniu je schopnosť presne reprezentovať priestor v troch dimenziách. Ďalšou výhodou laserového mapovania je schopnosť snímať svoje prostredie aj za zlých podmienok [21].

Výsledkom laserového mapovania je množina bodov, ktorú nazývame mračno bodov. Príklad vizualizácie naskenovaného mračna je možné vidieť na obrázku 2.1.



Obr. 2.1: Vizualizácia mračna bodov

Medzi najznámejšie triedy použitia technológie laserového mapovania patrí:

- Použitie počas letu (snímanie so systémom LiDAR pripevneným napr. k lietadlu)
- Použitie v stacionárnej pozícii

- Použitie v pohybujúcom sa vozidle alebo inom pozemnom nosiči (napr. batohové riešenie)

2.1 Technológia LiDAR

LiDAR je technológia pre meranie vzdialenosti k meranému objektu, pričom táto technológia funguje podobne ako radar. Hlavný rozdiel medzi systémom LiDAR a radarom je v používaní kratších vlnových dĺžok. Systémy LiDAR najčastejšie využívajú infračervené svetlo, kdežto radar používa rádiové vlny. V závislosti od metodiky použitej na zachytenie údajov môžu mať zachytené údaje rôznu hustotu [31].

Použitie technológie LiDAR

Technológia LiDAR je dnes bežne používaná v oblastiach ako astronómia, geológia, biológia, geodézia, archeológia [31, 27], a taktiež v systémoch ADAS (Advanced Driver Assistance Systems).

Princíp fungovania

Existujú dva druhy merania vzdialenosti medzi snímaným bodom a systémom LiDAR. Prvý prístup používa čas letu a druhý prístup používa výpočet fázového posunu. Zariadenia používajúce pre výpočet vzdialenosti čas letu dokážu zachytiť stovky až tisícky bodov za sekundu, kdežto zariadenia používajúce pre výpočet vzdialenosti fázový posun dokážu zachytiť stovky tisíc bodov za sekundu, čo je ich nesmiernou výhodou. Na druhej strane zariadenia používajúce čas letu majú oveľa väčší rozsah priestoru, ktorý dokážu snímať, čo ich robí výbornými kandidátmi na skenovanie veľmi vzdialených objektov. Oba typy zariadení narážajú na technologické limity, pre ktoré nemôžu úplne poraziť druhý typ zariadenia. Pre prvý typ zariadení používajúci čas letu, je limitom pulzová frekvencia, pretože laser musí čakať, kým sa každý lúč vráti, a až potom môže vyslať ďalší. Pre druhý typ zariadení používajúci fázový posun je technickým limitom vzdialenosť, nakoľko vysielaný signál na dlhšie vzdialenosti by musel byť oveľa silnejší, a taktiež snímané priebehy vln by boli priveľmi dlhé, čím by trpela presnosť [1].

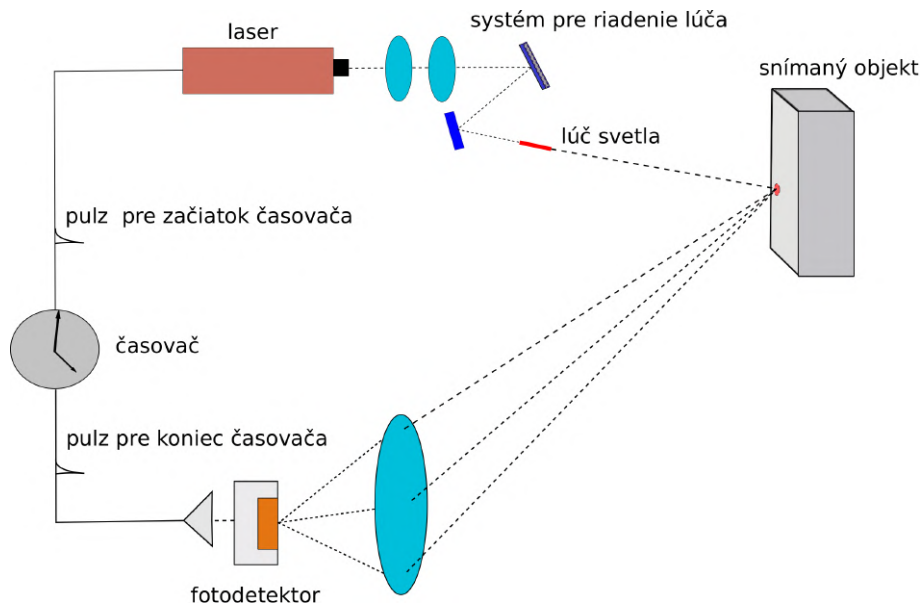
Pri metodike výpočtu času letu lúča laser vytvorí optický impulz, ktorý sa preniesie, odrazí od objektu a vráti k prijímaču. Prijímač presne odmeria čas, ktorý uplynie medzi vyslaním a prijatím impulzu. Keďže rýchlosť cestovania pulzu je rovnaká ako rýchlosť svetla v danom prostredí (táto rýchlosť je známa), je možné previesť čas cesty lúča na vzdialenosť. Na základe údajov o rozsahu lasera, uhle laserového skenu, pozícii lasera (z GPS), laserovej orientácie (z INS), kalibračných dát, je možné vypočítať presné súradnice pre každý jeden prijatý impulz [24]. Prijímač zaznamená čas návratu a intenzitu lúča, ktorý sa vrátil, a následne je vypočítaný čas letu lúča. Vzťah medzi časom, rýchlosťou a vzdialenosťou k cieľu (odrazová plocha) môže byť vyjadrený nasledovne:

$$d = \frac{t_1 - t_2}{2} * c \quad (2.1)$$

kde d je vzdialenosť k cieľu, t_1 je čas vyslania lúča, t_2 je čas návratu lúča a c je rýchlosť šírenia svetla v danom prostredí. Rozdiel medzi vyslaním a prijatím lúča (čas, kedy lúč cestoval) je ešte predelený na polovicu (lúč cestoval vzdialenosť k cieľu dvakrát) [25].

Príklad fungovania tejto metódy je možné vidieť na obrázku 2.2, kde laser pri vyslaní lúča generuje impulz pre časovač (ten si poznačí čas, kedy bol impulz prijatý). Po odrazení

lúča od objektu a jeho zachytení systémom LiDAR je opäť generovaný impulz pre časovač. Časovač môže v tomto momente vypočítať čas cesty lúča. Následne systém LiDAR dokáže dopočítať vzdialenosť podľa vyššie uvedeného vzorca.



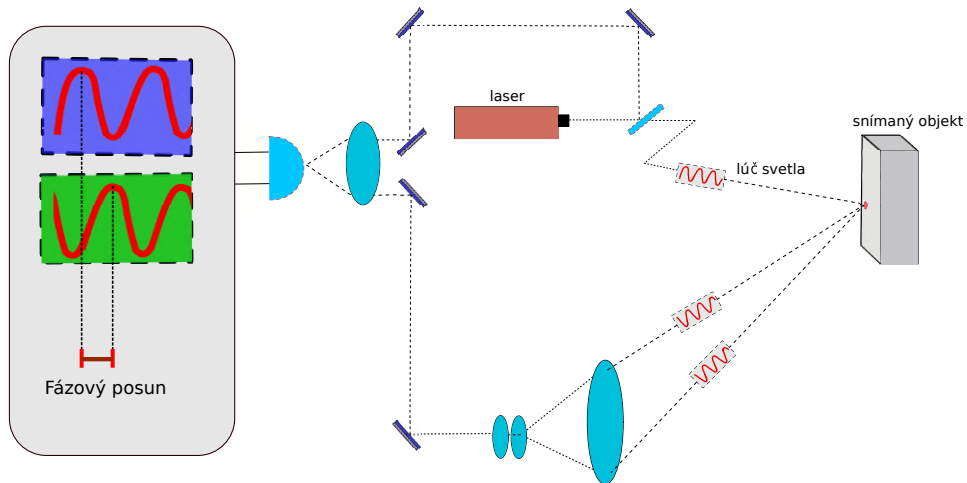
Obr. 2.2: Znáznornenie metodiky používajúcej čas letu lúča [39]

Pri tejto metodike merania vzdialenosti bodu je dôležité správne nastaviť periódu opakovania vysielania signálu, ktorú nastavuje užívateľ. Pre jej výpočet, je potrebná znalosť najvzdialenejšieho snímaného objektu, čiže najdlhšia snímaná vzdialenosť. Výpočet tejto periódy je odvodený z následného vzorca:

$$R_{max} = \frac{c * T}{2} \quad (2.2)$$

kde R_{max} je najväčšia snímaná vzdialenosť, c je rýchlosť šírenia svetla v danom prostredí a T je hodnota periódy vysielania signálu. Z toho vyplýva, že perióda vysielania signálu priamo ovplyvňuje maximálnu snímateľnú vzdialenosť. Naopak minimálna vzdialenosť pre snímanie touto metódou je obmedzená saturáciou fotodetektora, optickým zarovnaním a elektronickým oneskorením [39]. Pri týchto systémoch je dôležitý časovač daného zariadenia [27].

Druhá používaná metodika, používajúca výpočet fázového posunu (fázový posun je posun krivky, v tomto prípade sínus, po osi x), potrebuje pre výpočet vzdialenosti komplikovanejší systém. Príklad systému používajúceho túto metodiku je uvedený na obrázku 2.3. Pre túto metodiku je potrebných viac komponentov. Namiesto použitia impulzného laserového zdroja, je používaný kontinuálny zdroj, kde je výkon modulovaný konštantnou frekvenciou (na obrázku zjednodušene označené ako laser). Výstup lasera je teda funkcia $f = \sin(x)$, kde os x značí čas a os y výkon lasera (na obrázku vrchná krivka v modrej ploche). Nakoľko fotodetektory dokážu okrem prítomnosti svetla, zistiť aj jeho silu, je možné vytvoriť sínusovú krivku návratového signálu (na obrázku spodná krivka v zelenej ploche). Dopplerov jav spôsobí, že lúč, ktorý prešiel dlhšiu vzdialenosť (objekt je vzdialený viac), má väčší fázový posun frekvencie (na obrázku spodná časť šedej plochy) ako lúč, ktorý prešiel kratšiu vzdialenosť.



Obr. 2.3: Znáznornenie metodiky používajúcej fázový posun [39]

Pomocou fázového posunu, získaného z rozdielu medzi vrcholmi vln v radiánoch, je možné zistiť vzdialenosť k objektu s nasledujúcou rovnicou,

$$d = \frac{c * \Delta\phi}{2\pi * f} \quad (2.3)$$

kde d je vzdialenosť, c je rýchlosť svetla v danom prostredí, $\Delta\phi$ je fázový rozdiel a f je frekvencia s ktorou bol výkon lasera modulovaný [39].

Obe používané metodiky generujú trojrozmerné informácie o prostredí a jeho povrchu. Bežne sú ukladané informácie o polohe, výške a lesklosti povrchu bodu [12].

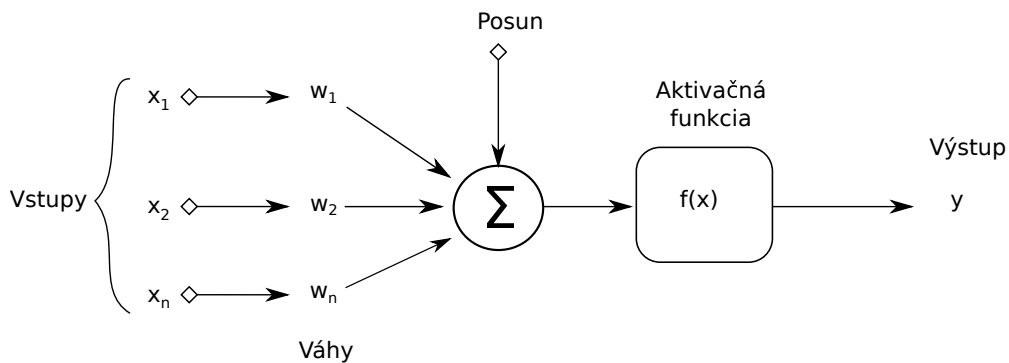
Kapitola 3

Neurónová sieť

Neurónová sieť je forma algoritmu strojového učenia. [34]. Primárna schopnosť neurónových sietí je rozpoznávať vzorce vo svete okolo nás. Neurónová sieť pozostáva z malých jednotiek, pripomínajúcich neuróny nachádzajúce sa v mozgu. Tieto jednotky sú nazývané neuróny a sú zoskupené do vrstiev. Vrstva je množina neurónov [22, 3].

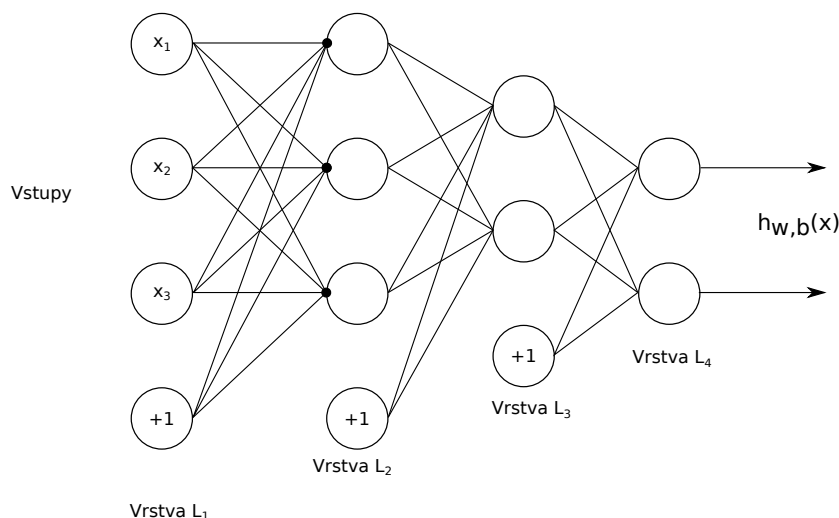
3.0.1 Umelý neurón

Umelý neurón je základný prvok neurónovej siete [4]. Skupina týchto prvkov v neurónovej sieti sa nazýva vrstva neurónovej siete. Neurón má vstupy a výstup (na obrázku 3.1 sú vstupmi neurónu x_1, x_2, x_n). Na vstupy neurónov je pripojený vstup pre danú neurónovú sieť, ak sa jedná o prvú vrstvu, inak sú na vstupy neurónov pripojené výstupy iných vrstiev (vo väčšine prípadov). Na výstup (na obrázku 3.1 je výstup neurónu y) je privedená hodnota, ktorú neurón vypočíta pre dané vstupy. Každý vstup má svoju váhu (váha spojenia, na obrázku 3.1 w_1, w_2, w_3). Neurón vezme hodnotu z pripojeného vstupu a vynásobí ju váhou pre daný vstup. Súčet všetkých pripojených vstupov vynásobených ich váhami a bias hodnoty je vstup pre aktivačnú funkciu (na obrázku 3.1 $f(x)$). Aktivačná funkcia jej vstup matematicky transformuje a výsledok priradí na výstup neurónu.



Obr. 3.1: Umelý neurón [4]

Prevádzka neurónovej siete je nasledovná, do prvej vrstvy neurónov sa načítajú vstupy, výstupy z prvej vrstvy pokračujú ako vstupy do ďalšej vrstvy neurónov, ktorá opäť produkuje výstupy. Výstup, ktorý vracia posledná vrstva, je samotný výstup neurónovej siete (napr. zviera na obrázku).



Obr. 3.2: Príklad neurónovej siete [35]

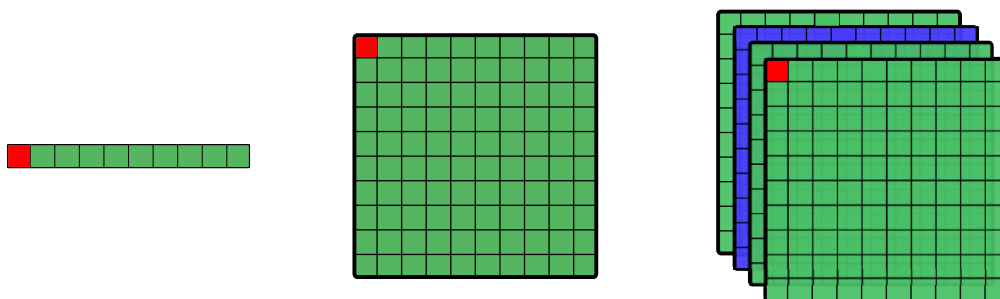
Na Obr.3.2 je napríklad vrstva L1 vstupnou vrstvou, potom nasledujú dve vnútorné vrstvy L2 a L3 (nazývané tiež skryté). Posledná vrstva L4 je vrstva, ktorá obsahuje všetky možné výstupy siete (každý neurón reprezentuje jeden možný výstup) [4]. Matematicky je možné výstup neurónu vyjadriť následne:

$$y = f\left(\sum_{n=1}^n x_n * w_n + b\right) \quad (3.1)$$

kde x_n je n-tý vstup neurónu, w_n je n-tá váha neurónu, b je posunutie (bias) a $f()$ je aktivačná funkcia.

3.0.2 Vstup neurónovej siete

Vstupy pre neurónové siete, môžu nadobúdať rôzne tvary (závisí od riešeného problému, u a tiež architektúry neurónovej siete). Najčastejšie sú vstupy znázornené na obrázku 3.3 (červená farba znázorňuje jeden element vstupu - jedna hodnota). V ľavej časti obrázka sa nachádza vstup, ktorý má tvar vektora. V strednej časti sa nachádza vstup, ktorý má tvar poľa. V pravej časti obrázka sa nachádza množina polí, pričom modrá farba značí jeden prvok tejto množiny - vrstvu vstupu (v angl. feature map). Táto množina polí sa nazýva tenzor.

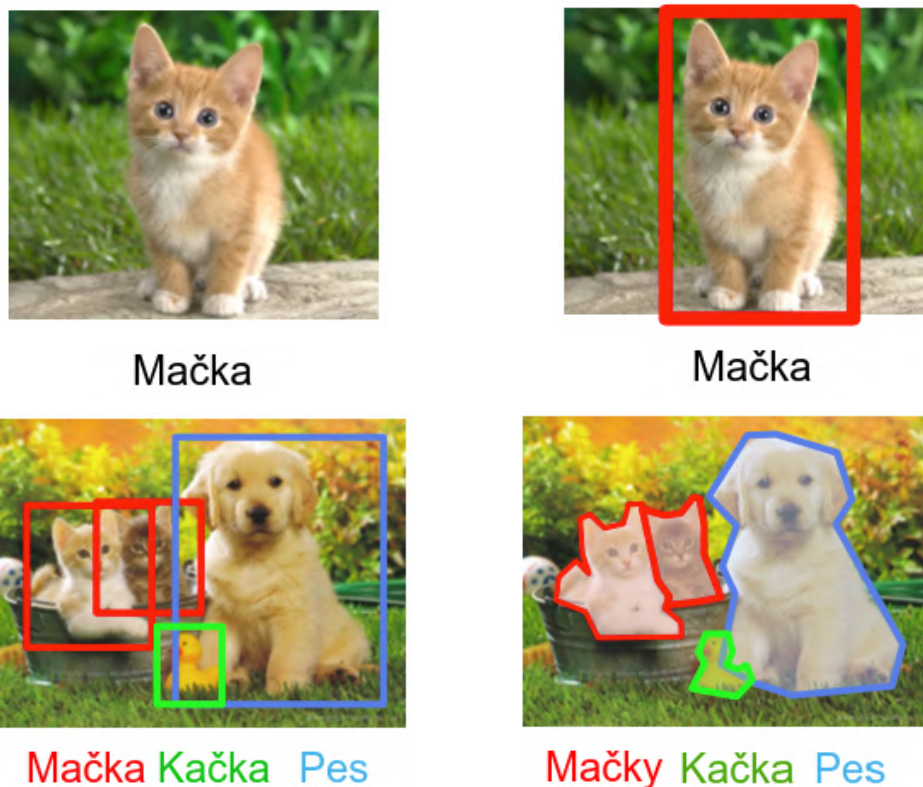


Obr. 3.3: Príklad vstupov pre neurónové siete

Rôzne vlastnosti vstupných dát sú rozdelené do *kanálov*, pričom každý kanál predstavuje jednu vrstvu vstupu (vrstva vstupu má tvar matice), napríklad rgb obrázok má 3 kanály. Každý kanál má svoju filtračnú maticu. [15].

3.0.3 Používanie neurónových sietí

Neurónové siete majú obrovské množstvo možností použitia pre veľmi široké účely. Popísané budú niektoré z prístupov, postupne podľa zložitosti úlohy.



Obr. 3.4: Typy výstupov neurónových sietí [14]

Neurónové siete, ktoré prevádzajú *klasifikáciu vstupu*, tento vstup spracujú vo svojej vnútornej štruktúre a na výstup siete sú privedené hodnoty príslušností dát zo vstupu ku klasifikovaným výstupným triedam (napr. ľavú hornú časť obrázka 3.3 by sieť zameraná na klasifikáciu, zaradila k triede obrázkov, na ktorých sa nachádza mačka). [11].

Pri zložitejšom použití neurónových sietí, kedy je vyžadovaná *lokalizácia klasifikovaného objektu*, je úlohou siete určiť príslušnosť dát k výstupnej triede, no zároveň musí tento objekt v dátach presne lokalizovať. Na obrázku 3.3 v pravej hornej časti obrázku je možné vidieť lokalizovanú a klasifikovanú mačku (popis Mačka a červený rám) [11].

Neurónové siete, kde úlohou je *detekcia viacerých objektov* vo vstupných dátach predstavujú ďalší stupeň zložitosti neurónových sietí. Tieto siete vykonávajú lokalizáciu pre viacero objektov v dátach. Na obrázku 3.3 v ľavej spodnej časti je vidieť príklad, ako by mohol vyzeráť výstup popisovanej siete.

Nasledujúcim stupňom je *segmentácia objektov*, ktorej úloha je zaradiť všetky dáta k zodpovedajúcim triedam [11]. Na obrázku 3.3 v spodnej pravej časti, je možné vidieť,

ako približne segmentácia objektov vyzerá. Používajú sa dva typy segmentácie, sémantická a inštančná. Inštančná segmentácia dokáže klasifikovať niekoľko objektov rovnakej triedy. Naopak sémantická segmentácia dokáže každý pixel zaradiť ku triede, no nerozlíši viac objektov rovnakej triedy [14]. Na obrázku 3.3 v pravej spodnej časti sa nachádza inštančná segmentácia. V prípade sémantickej segmentácie by obe mačky na obrázku 3.3 v pravej spodnej časti boli klasifikované ako jeden objekt. V práci bola implementovaná sémantická segmentácia vozovky.

3.0.4 Aktivačné funkcie neurónových sietí

Aktivačná funkcia [36], často nazývaná aj prenosová funkcia, je matematická funkcia, ktorá transformuje výstup neurónov. Používa sa v umelých neurónových sieťach, pre transformáciu výstupných signálov neurónov, ktoré sú po tejto transformácii privedené na vstupy neurónov ďalších vrstiev. V neurónových sieťach nachádzajú aktivačné funkcie miesto po spočítaní sumy váhami vynásobených vstupov a posunu, tak ako je to znázornené na obrázku 3.1. Ak by nebola použitá aktivačná funkcia, výstup siete by bola lineárna funkcia (polynóm prvého stupňa). Vďaka aktivačným funkciám, je možné získať nelineárne mapovanie vstupov na výstupy. Dôležitá vlastnosť aktivačnej funkcie je možnosť jej derivovania, aby mohla byť vypočítaná chyba s ohľadom na váhy, a tiež optimalizované váhy pomocou gradientu alebo akejkoľvek inej optimalizačnej techniky pre zníženie chyby [36]. Použitie aktivačnej funkcie umožňuje riešiť zložitejší problém pomocou menšej siete. Existuje veľa druhov aktivačných funkcií. [41].

Jednou z najpoužívanejších funkcií pri dopredných neurónových sieťach je *ReLU* (Rectified linear unit). Je častokrát prvou voľbou pri tvorbe modelu siete. Má veľmi efektívny výpočet [32]. Relu aktivačná funkcia má nasledovný tvar:

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (3.2)$$

Taktiež často používaná aktivačná funkcia *ELU* (exponential linear unit), má na rozdiel od ostatných aktivačných funkcií, parameter navyše α , ktorého hodnota by mala byť kladná. Hodnota tohto parametra je bežne medzi 0.1 a 0.3. Taktiež rovnako ako ReLU sa používa pri konvolučných neurónových sieťach.

$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x < 0 \end{cases} \quad (3.3)$$

Funkcia je kvôli výrazu e^x , výpočtovo drahšia, ako napríklad jej podobná funkcia ReLU, a taktiež na rozdiel od ReLU môže produkovať záporné hodnoty [33].

V práci bola taktiež použitá aktivačná funkcia typu *softmax*. Ide o funkciu, ktorá transformuje vektor zložený z n reálnych hodnôt na vektor zložený z n reálnych hodnôt, pričom suma tohto transformovaného vektora má hodnotu 1 a jednotlivé hodnoty sa nachádzajú v intervale $< 0; 1 >$. Toto umožňuje nahliadať na hodnoty výstupného vektora ako na hodnoty pravdepodobnosti. Vstupný vektor môže obsahovať záporné, pozitívne alebo aj vysoké čísla. Funkcia softmax malé a záporné hodnoty transformuje na malú pravdepodobnosť a vysoké hodnoty transformuje na vysokú pravdepodobnosť [38].

3.1 Trénovanie neurónovej siete

Trénovanie neurónovej siete je proces, pri ktorom sa menia parametre neurónovej siete za účelom zlepšenia presnosti výsledkov, ktoré daná sieť produkuje. Počas trénovania neurónovej siete sa používajú hlavne dva prístupy, trénovanie s učiteľom a trénovanie bez učiteľa [22].

Pri trénovaní bez učiteľa má sieť k dispozícii vstupy pre trénovanie, no nemá zodpovedajúce výstupy. Jej cieľom je odhaliť vzory vo vstupných dátach bez toho, aby jej bolo ukázané, čo má hľadať [22].

Pri trénovaní s učiteľom majú všetky trénovacie dáta dvojicu vstupných dát a požadovaných výstupov. Akonáhle je sieť schopná dostatočne presne určiť výstupy pre testovacie dáta, môže byť táto neurónová sieť použitá na reálne dáta (dáta, ktoré neboli použité pri trénovaní siete) [22]. Pri trénovaní sieť určuje výstupy pre vstupné dáta a porovnáva ich s požadovanými výstupmi. Čím je rozdiel väčší, tým viac sa menia parametre siete (konkrétne sa menia napr. váhy pre vstupy neurónov). Pre určenie váh, ktoré majú byť modifikované, sa používa algoritmus spätnej propagácie. Tento algoritmus spočíva v prejdení neurónovej siete odzadu (od výstupu k vstupu) a preskúmania každého spojenia, aby sa zistilo, ako sa bude správať výstup v závislosti od zmeny váh pre tieto spojenia [4].

Pri učení je jedným z najdôležitejších parametrov rýchlosť učenia neurónovej siete. Tento parameter určuje, ako rýchlo sa neurónová sieť učí. Určuje, ako rýchlo sa budú meniť jednotlivé váhy neurónovej siete počas učenia (aký veľký podiel každej váhy neurónovej siete, je možné zmeniť v jednom kroku trénovania). Ide o jeden z najdôležitejších parametrov pri trénovaní neurónovej siete.

3.1.1 Algoritmus spätnej propagácie chyby

Algoritmus spätnej propagácie chyby (backpropagation) spočíva v porovnaní výsledku, ktorý vznikol použitím neurónovej siete, a očakávaným výsledkom a v následnom výpočte gradientu chyby. Tento algoritmus pre svoje správne fungovanie potrebuje, aby mu predchádzali dva procesy. Prvým procesom je prechod vstupu vnútornou štruktúrou neurónovej siete a druhým procesom je výpočet hodnoty stratovej funkcie. Následne sú prevedené dva kroky algoritmu spätnej propagácie chyby - spätný prechod neurónovou sieťou a úprava jej váh.

Spätný prechod neurónovou sieťou

Spätný prechod neurónovou sieťou je činnosť, kedy sa rozhoduje, ktoré váhy najviac ovplyvnili hodnotu stratovej funkcie, a hľadá sa spôsob, ako upraviť tieto váhy tak, aby bola hodnota stratovej funkcie nižšia ako aktuálna hodnota [10]. Používa sa gradient chyby, ktorý sa vypočíta ako derivácia funkcie, ktorá je tvorená jednotlivými váhami na vstupe každého neurónu. Tento výpočet je vykonaný pre každú vrstvu siete. Gradient chyby je možné si predstaviť ako smer, ktorým sa váhy menia [29].

Úprava váh

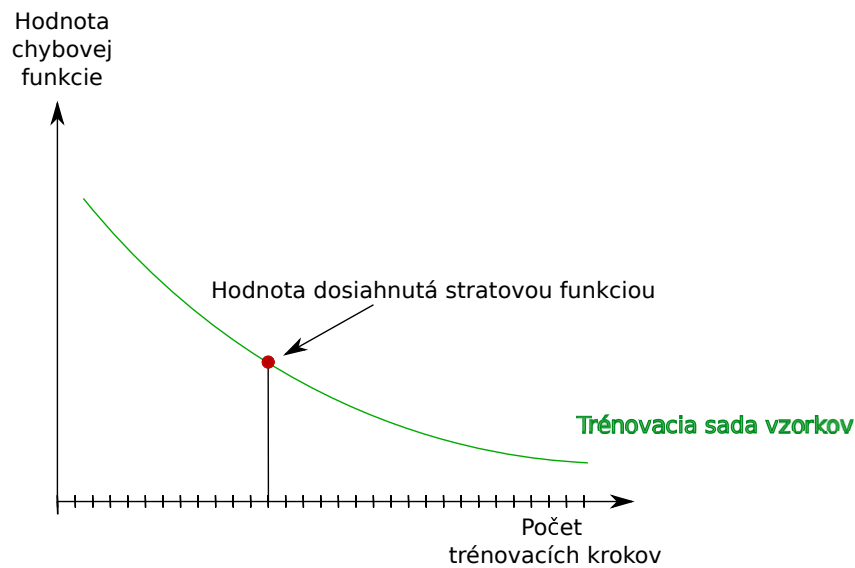
V tomto kroku sa upravujú váhy filtrov. Sú upravované v opačnom smere gradientu chyby. Pre výpočet novej váhy sa nasledujúci vzorec.

$$w = w_i - \alpha \frac{dL}{dW} \quad (3.4)$$

V rovnici 3.4 w je nová vypočítaná váha, w_i je aktuálna váha (doteraz používaná), α je parameter, ako rýchlo sa sieť učí (ako veľmi sa zmení s jedným cyklom učenia váha - ide o parameter rýchlosti učenia), dL a dW je pomer, ktorý vyjadruje hodnotu gradientu [10].

3.1.2 Chybová funkcia

Chybová funkcia [16] (v angl. Loss function) je funkcia určujúca, ako presný je špecifický algoritmus. Táto funkcia pri učení s učiteľom udáva hodnotu rozdielu medzi výstupom algoritmu a jeho očakávaným výstupom (očakávané výstupy sa nazývajú aj ground truth). Existuje veľa typov stratových funkcií, no všetky v zásade hodnotia riešenie podľa toho, ako sú vzdialené predpokladané hodnoty od hodnôt, ktoré by algoritmus mal produkovať v ideálnom stave [16].



Obr. 3.5: Príklad grafu hodnôt stratovej funkcie počas tréningu

Na obrázku 3.5 je možné vidieť príklad, ako sa hodnoty stratovej funkcie vyvíjajú počas tréningu neurónovej siete. Červený bod ukazuje jednu z dosiahnutých hodnôt počas tréningu. Pre výpočet chybovej funkcie sa používa euklidovská vzdialenosť. Všeobecne je možné vzorec pre výpočet chybovej funkcie zapísať nasledovne:

$$f(X, Y) = \sum \frac{1}{2}(X - Y)^2 \quad (3.5)$$

kde X značí očakávaný výstup algoritmu a Y značí obdržaný výstup algoritmu. Výsledok $f(X, Y)$ sa snažíme pomocou učenia dostať na čo najnižšiu hodnotu [10].

Chybová funkcia typu CrossEntropyLoss

V práci bola použitá chybová funkcia typu CrossEntropyLoss, ktorá je vhodná pre použitie pri klasifikačnom probléme s n triedami. Túto chybovú funkciu je možné vyjadriť nasledovným spôsobom:

$$f(x, c) = -x_c + \log\left(\sum_j g(x_j)\right) \quad (3.6)$$

kde $f(x, c)$ značí hodnotu chybovej funkcie pre danú triedu, x_c značí očakávanú hodnotu pravdepodobnosti pre danú triedu a x_j značí obdržané hodnoty pravdepodobnosti pre danú triedu.

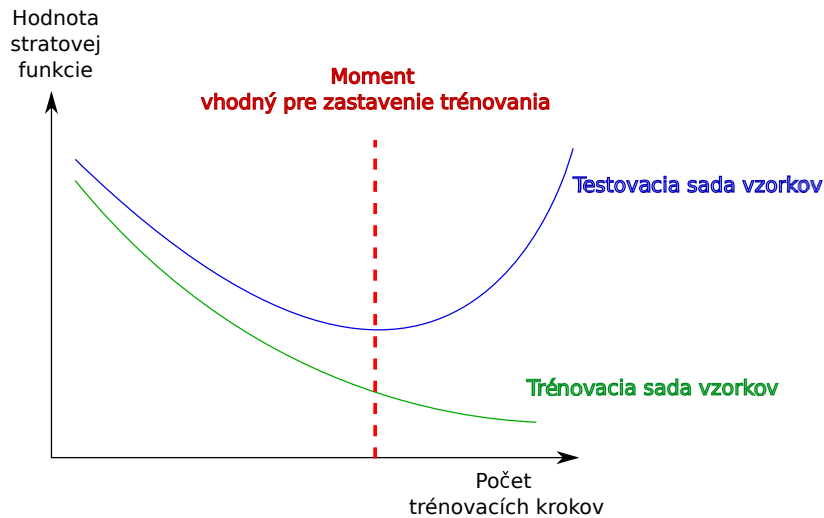
V prípade použitia váženého typu funkcie typu CrossEntropyLoss, matematický zápis (s použitím predchádzajúceho vzorca) vyzerá nasledovne:

$$l = \frac{\sum_{i=1}^N f(i, c_i)}{\sum_{i=1}^N w * c_i} \quad (3.7)$$

kde w značí váhy pre jednotlivé triedy c_i .

3.1.3 Pretrénovanie neurónovej siete

Pretrénovanie neurónovej siete je stav, kedy sa daná sieť príliš prispôsobí tréningovým dátam. Neurónová sieť pri pretrénovaní produkuje skvelé výsledky pre tréningové dáta, no zlé výsledky pre testovacie dáta (dáta, ktoré neboli používané pri tréningu). Pretrénovaniu sa dá vyhnúť napríklad rozšírením dátovej sady určenej na tréning, zjednodušením neurónovej siete, pridaním rušiacich vrstiev (v angl. dropout) alebo pomocou rozdelenia tréningového datasetu na dve skupiny (jedna na tréning a druhá na testovanie). Pri poslednom spôsobe je neurónová sieť tréningovaná pomocou tréningového datasetu, no presnosť danej siete je určená testovacím datasetom. Testovací dataset jednoducho zastaví učenie, keď sa daná sieť začne príliš prispôbovať dátam, na ktorých sa vyvíja [16].



Obr. 3.6: Znáročenie stavu pretrénovania neurónovej siete

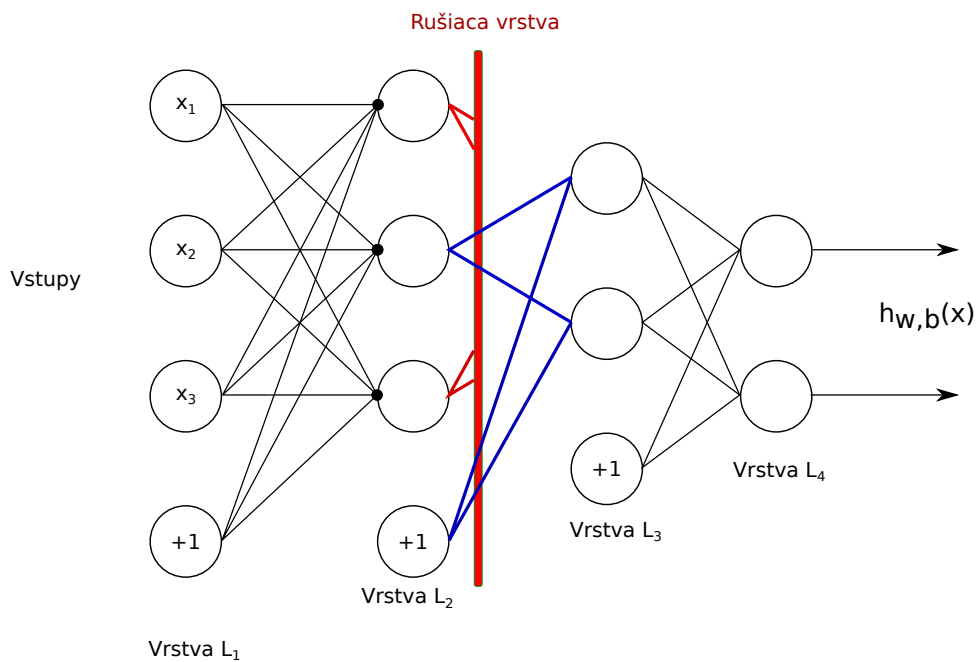
Ako je možné vidieť na obrázku 3.6, neurónová sieť sa v určitom bode začne prispôbovať tréningovým dátam. Na obrázku 3.6 je možné vidieť stúpajúci rozdiel v presnostiach medzi testovacími a tréningovými dátami. Neurónové siete bývajú vytvorené pre použitie na všeobecných (reálnych) dátach, preto je nutné nájsť vhodný moment, kedy je potrebné učenie zastaviť, v prípade ak nastáva pretrénovanie neurónovej siete.

3.2 Typy vrstiev v neurónových sieťach

Vrstvy v neurónových sieťach majú rôzne typy, podľa toho aké neuróny sú použité, a ako sú prepojené. Medzi najznámejšie typy patria vstupné vrstvy, plne prepojené dopredné vrstvy, konvolučné vrstvy, prevzorkovacie vrstvy (v angl. pooling), rušiace vrstvy (v angl. dropout) a rekurentné vrstvy [22].

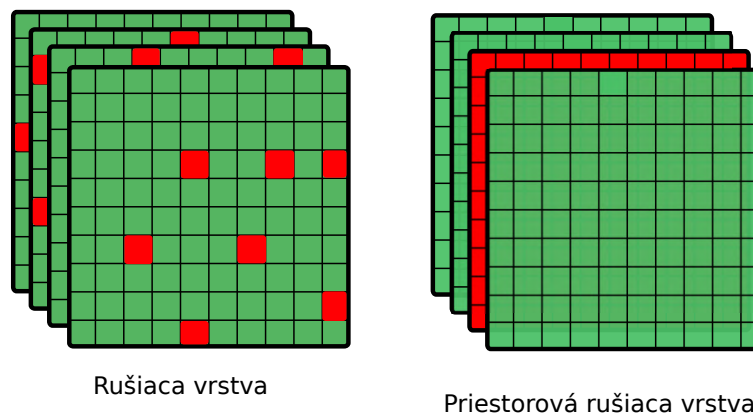
Rušiaca vrstva

Rušiaca vrstva náhodne vyberá vstupy, ktoré nastavuje na nulu, pričom závisí od nastavenej pravdepodobnosti výberu. Táto operácia účinne mení základnú architektúru medzi iteráciami a pomáha tým zabrániť pretrénovaniu siete [5]. Príklad toho ako pracuje rušiaca vrstva je možné vidieť na obrázku 3.7, kde časť vstupov vrstvy L3 (výstup vrstvy L2) bola nastavená na nulu (znázornené červenou farbou).



Obr. 3.7: Názorný príklad práce rušiackej vrstvy

Pri použití *priestorovej rušiackej vrstvy*, sú náhodne vybrané celé vrstvy vstupov, ktoré sú nastavené na nulu [40]. Rozdiel medzi rušiacou vrstvou a priestorovou rušiacou vrstvou je znázornený na obrázku 3.8.



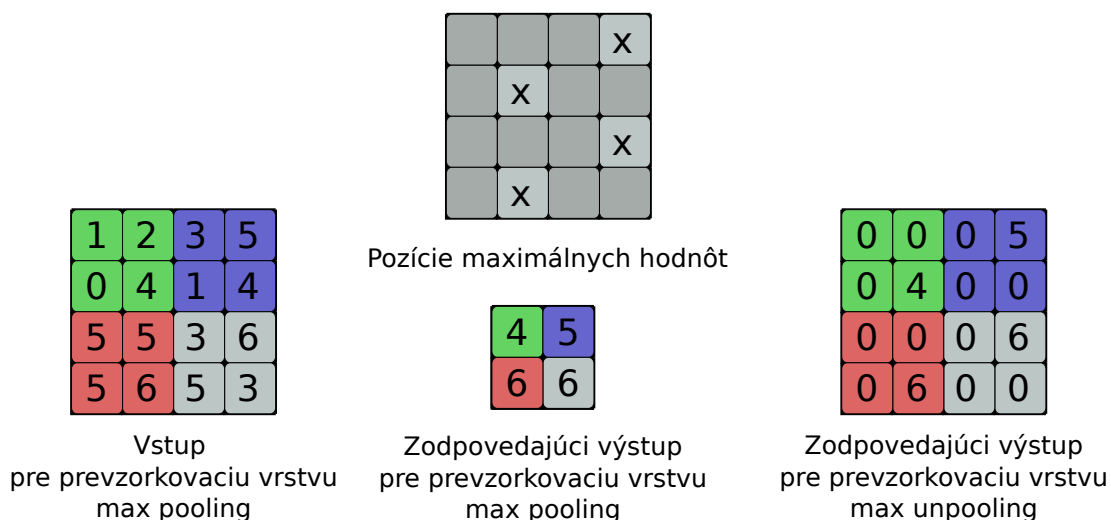
Obr. 3.8: Znázornený rozdiel vstupov pre sieť po aplikáciách rušiacej vrstvy a priestorovej rušiacej vrstvy

Prevzorkovacia vrstva

Prevzorkovacia vrstva slúži pre zmenu rozmerov výstupov vrstiev, pričom zachováva najdôležitejšie informácie [15]. Používajú sa rôzne typy (napr. max pooling, avg pooling, max unpooling, ...). Príklad použitia prevzorkovacej vrstvy typu maximum pooling a následného použitia prevzorkovacej vrstvy typu maximum unpooling je možné vidieť na obrázku 3.9.

Pre použitie *prevzorkovacej vrstvy typu maximum pooling* je potrebné definovať veľkosť priestoru, z ktorého je vybraný prvok s najvyššou hodnotou z vrstvy vstupu, (na obrázku 3.9 má tento priestor rozmer 2×2 , vyberá sa teda zo štyroch prvkov). Postupne je tento priestor posúvaný po celom poli vstupu [15]. Použitie tejto vrstvy je sprevádzané stratou informácií. Na obrázku 3.9 je vidno v ľavej časti vstup pre popisovanú vrstvu. V strednej časti obrázka 3.9 sa nachádza očakávaný výstup, pri použití 2×2 priestoru, z ktorého vrstva vyberá maximum. Pozície maximálnych hodnôt (stredná vrchná časť obrázka 3.9) je voliteľný výstup popisovanej vrstvy.

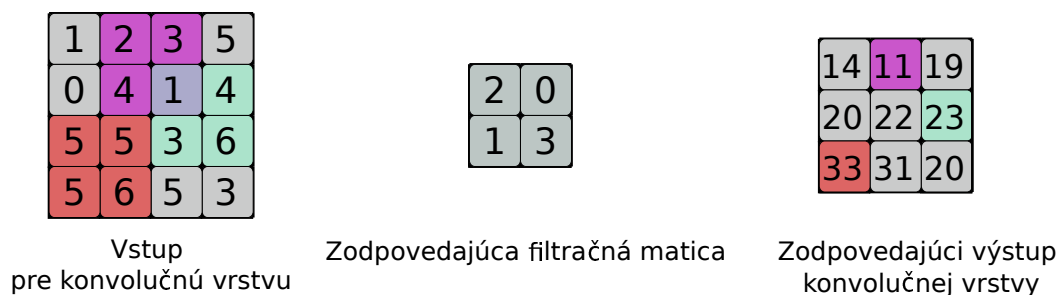
Prevzorkovacia vrstva typu maximum unpooling sa používa pre približné obnovenie pôvodných hodnôt výstupných vrstiev. Pre použitie popisovanej vrstvy je potrebné pri použití vrstvy typu maximum pooling, zachovať pozície, z ktorých sa vzali maximálne hodnoty (na obrázku 3.9 znázornené v strednej vrchnej časti). Na tieto pozície sa následne pri aplikácii vrstvy typu maximum unpooling vrátia vybrané maximálne hodnoty [26]. Na obrázku 3.9 je možné vidieť aplikáciu popisovanej vrstvy, kde v strednej časti obrázka je vstup pre túto sieť a v pravej časti obrázka pridružený výstup.



Obr. 3.9: Znáznomená činnosť prevzorkovacích funkcií typov max pooling a max unpooling

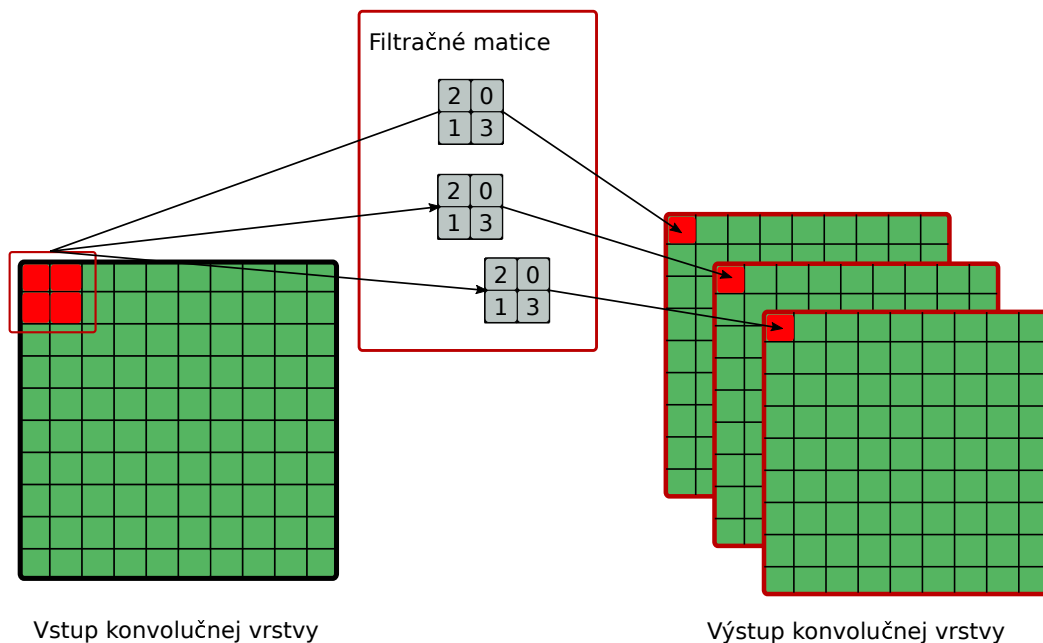
Konvolučná vrstva

Meno tejto vrstvy vychádza z operácie konvolúcie. Táto operácia potrebuje dve matice - vstupnú maticu pre vrstvu a filtračnú maticu, nazývanú tiež jadro (z angl. kernel). Pri konvolúcii je filtračná matica (obrázok 3.10, stredná časť) posúvaná po vstupnej matici (obrázok 3.10, ľavá časť), pričom na každej pozícii (prvá + každá ďalšia pozícia vytvorená posunom), je prevedené maticové násobenie. Výsledok maticového násobenia je sčítaný a následne uložený do výstupnej matice (obrázok 3.10, pravá časť) na príslušnú pozíciu.



Obr. 3.10: Príklad znázornenia použitia konvolučnej vrstvy [15]

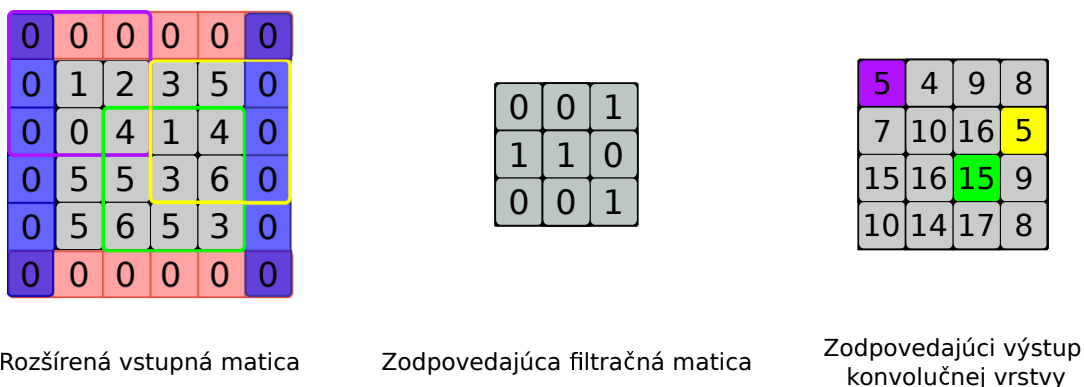
Pri tomto type vrstvy je najvýznamnejšie použitie filtračných matíc. Počet filtračných matíc môže byť v jednej konvolučnej vrstve ľubovoľný. Na obrázku 3.11 je možné vidieť zjednodušený príklad konvolúcie s použitím troch rôznych filtračných matíc. Použitie vyššieho počtu filtračných matíc spôsobuje navýšenie počtu vrstiev výstupu konvolučnej vrstvy.



Obr. 3.11: Príklad použitia vyššieho počtu filtročných matic [15]

Zvyčajné hodnoty pre veľkosť filtročnej matice sú 1×1 , 3×3 , 5×5 , 7×7 . Najčastejšie sa používajú nepárne hodnoty.

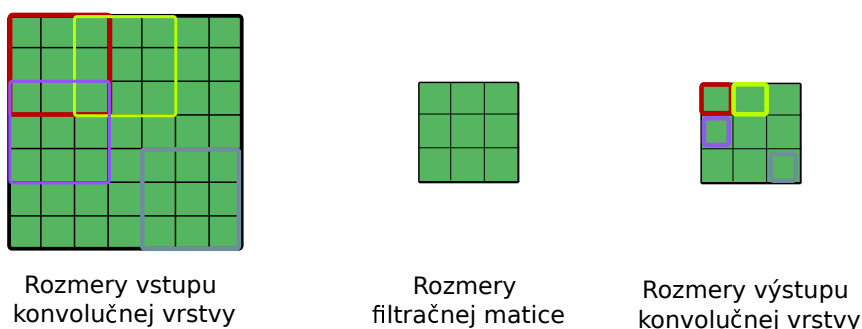
Pri konvolučných vrstvách sa častokrát používa metóda zväčšenia vstupnej matice. Ide o spôsob konvolúcie, pri ktorom sa ku okrajom pôvodnej matice pridajú ďalšie riadky alebo stĺpce [15]. Toto rozšírenie sa používa hlavne pre zaistenie jednotných rozmerov na vstupe a výstupe vrstvy. Príklad tohto rozšírenia je možné vidieť na obrázku 3.12. Na obrázku 3.12 pôvodný vstup vrstvy značí šedá farba v prvej (ľavej) časti obrázka. Tento vstup je zväčšený pridaním modrých stĺpcov a červených riadkov. Následne je vykonaná operácia konvolúcie pomocou filtročnej matice zobrazenej v strednej časti obrázka 3.12. Výsledok tejto operácie je možné vidieť v pravej časti obrázka 3.12. V prípade vyobrazenom na obrázku 3.12 bol pridaný na každú stranu len 1 stĺpec alebo 1 riadok, no je možné pridať ľubovoľné množstvo stĺpcov a riadkov.



Obr. 3.12: Príklad znázornenia ohraničenia vstupu

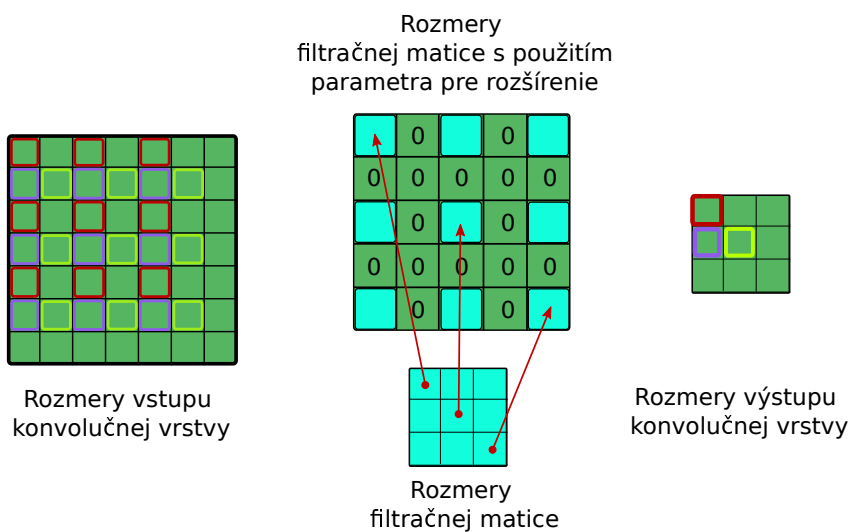
Ak sú na vstupe dáta s dimenziami $n \times n$ a následne je použitá konvolúcia s filtrom s dimenziami $f \times f$, výstup konvolúcie má dimenzie $(n + 2p_1 - f + 1) \times (n + 2p_2 - f + 1)$, kde parameter p_1 značí počet pridaných riadkov na jednu stranu vstupnej matice a parameter p_2 značí počet pridaných stĺpcov rovnako na jednu stranu vstupnej matice. Široká konvolúcia (valid convolution) znamená, že nie sú použité pridané hodnoty (vstupná matica nie je rozšírená). V tomto prípade $p = 0$, čo spôsobí, že výstup vrstvy má menšie rozmery ako vstup. V prípade, kedy je používaná úzka konvolúcia (same convolution), sú použité pridané hodnoty okolo okrajov. Rozmery vstupu a výstupu sú rovnaké ak je použitý dostatočne veľký parameter p [23].

Ďalším parametrom konvolúcie je *krok*. Krok je dvojica hodnôt, o koľko políčok sa posúva filtračná matica po x-ovej a y-ovej osi v každom kroku konvolúcie [30]. Čím je väčší krok, tým sú menšie výstupné vrstvy [15]. Príklad, konvolúcie s hodnotami kroku nastavenými na (3,3) je možné vidieť na obrázku 3.13.



Obr. 3.13: Príklad znázornenia kroku [11]

Posledným často používaným parametrom v konvolučných vrstvách je hodnota pre rozšírenie filtračnej matice používanej pri konvolúcii. Na obrázku 3.14 je znázornený spôsob rozšírenia tejto matice s použitím hodnôt 2,2 (prvá hodnota pre rozšírenie po zvislej osi, druhá pre rozšírenie po vodorovnej osi). Tento parameter sa často používa v konvolučných vrstvách, ktoré majú za úlohu segmentáciu vstupných dát [18].



Obr. 3.14: Príklad znázornenia použitia konvolučnej vrstvy s rozšírenou filtračnou maticou

Kapitola 4

State of the art

Úloha sémantickej segmentácie vozovky je oblasť, v ktorej existuje množstvo prístupov pre riešenie tohto problému. Táto oblasť čoraz väčšmi narastá spolu s príchodom autonómneho riadenia. Rôzne prístupy používajú rôzne typy dát. Vo väčšine metód sa primárne používajú konvolučné neurónové siete, ktoré vyhodnocujú kamerové snímky, laserové skeny alebo kombinácie laserových snímok a kamerových skenov. Snímky vytvorené kamerami sú výhodné pre ich jednoduchosť a rýchlosť tvorby, a taktiež hustotu informácií. Obrovskou nevýhodou kamerových snímok je ich rôznorodosť založená na aktuálnom počasí a aktuálnom čase. Laserové skeny sú naopak pomalé na vyhotovenie, ich samotné vyhotovenie vyžaduje drahšie komponenty, ako je tomu v prípade kamier, no nesmiernou výhodou laserových skenov je ich oveľa väčšia nezávislosť od aktuálnych svetelných podmienok a aktuálneho počasia. Taktiež veľmi dôležitou vlastnosťou, ktorá robí laserové skeny zaujímavejšími ako kamerové snímky, je schopnosť reprezentácie 3D priestoru [17, 2, 21].

4.1 Príklad metód používajúcich kombináciu kamerových snímok a laserových skenov

Hlavnou ideou tejto triedy metód je kombinácia výhod snímok z kamery a laserových skenov. Rozdiely v metódach tohto typu sú predovšetkým v spôsobe, akým sú snímky z kamier a laserové skeny použité pre dané konvolučné neurónové siete.

Jedna z najkomplexnejších metód, ktoré používajú kamerové snímky spolu s laserovými skenmi, bola navrhnutá Caltagironem et al. [6]. Táto konkrétna metóda premieta mračno bodov z laserového skenu do obrázku vytvoreného z kamery. Používa tri druhy konvolučných neurónových sietí, a zároveň tri prístupy k spájaniu laserových skenov a obrázkov z kamery. V prvom použitom prístupe sú laserové skeny a obrázky z kamier spojené v jeden tenzor, ktorý je použitý ako vstup pre konvolučnú neurónovú sieť. Druhý použitý prístup používa v konvolučnej neurónovej sieti dve paralelné vetvy, ktoré sa následne spoja. Pre prvú vetvu je použitý vstup vytvorený z obrázkov z kamery. Pre druhú vetvu je použitý vstup vytvorený z laserových skenov. Obe vetvy obsahujú 20 vrstiev, ktoré pracujú paralelne. Po prechode dát týmito dvoma paralelnými vrstvami sú ich výstupy spojené. Tieto spojené dáta následne pokračujú zvyškom architektúry konvolučnej neurónovej siete. Posledný tretí prístup používa taktiež neurónovú sieť obsahujúcu dve paralelné vetvy, ktoré sú spolu previazané trénovateľnými skalárnymi spojeniami na viacerých miestach.

Pri prvých dvoch metódach je potrebné rozhodnúť, kde budú dáta spojené, kdežto posledný tretí typ konvolučnej siete sa naučí dáta spájať sám. Najlepšie výsledky boli dosiahnuté použitím tretieho typu (F-measure 96.25%).

Metóda navrhnutá Chenom et al. [9] používa podobný princíp ako je použitý v treťom prístupe predchádzajúcej metódy. Hlavným rozdielom je rozmer a forma dát z laserových skenov, pričom táto metóda používa laserové dáta transformované pomocou priložených kalibračných matíc.

Výnimkou v metódach používajúcich laserové skeny a zároveň obrázky z kamier je metóda od autorov Chen et al. [8], ktorá používa laserové skeny transformované pomocou filtračných matíc pre jednu vetvu architektúry konvolučnej neurónovej siete, laserové skeny s pohľadom zvrchu pre druhú vetvu a obrázky z kamier pre tretiu vetvu. Tieto tri vetvy sú na určitom mieste architektúry neurónovej siete spojené do jednej vetvy, ktorá produkuje výsledky v 3D priestore (výsledky je možné premietnuť do 2D priestoru).

4.2 Príklad metód používajúcej iba kamerové snímky

Drvivá väčšina metód v tejto skupine používa pre segmentáciu vozovky konvolučné neurónové siete. Tieto metódy tvoria najväčší podiel medzi všetkými riešeniami. Nakoľko ide o segmentáciu snímok z kamier, pre túto oblasť existuje množstvo modelov neurónových sietí a nástrojov pre vytváranie ground truth dát. Pri tejto triede metód sú najväčšie rozdiely v spracovaní vstupov pre použitie konvolučnými neurónovými sieťami. Napríklad v metóde popísanej Alvarezom et al. [2] je obrázok konvertovaný spôsobom, ktorý odstráni z obrázku tieň, a taktiež zníži rozdiely medzi osvetlením jednotlivých obrázkov (záleží od časti dňa, v ktorej je snímka vyhotovená). Takto upravené obrázky sú následne spracované neurónovou sieťou. Hlavnú úlohu v tejto metóde hrajú informácie o farbách na snímke. Naopak v metóde popísanej Rasmussenom [28] hrajú hlavnú úlohu textúry povrchov na snímke. Táto metóda je založená na skutočnosti, že vozovka má relatívne jednotlivú textúru.

4.3 Príklad metódy používajúcej laserové skeny

Momentálne najúspešnejší príklad metódy používajúcej iba laserové skeny je metóda navrhnutá Caltagironem et al. [7]. Táto metóda taktiež používa konvolučnú neurónovú sieť. Z neusporiadaných laserových skenov sú vytvorené 2D obrázky priestoru pred vozidlom (použitý je pohľad zhora). Každý vytvorený obrázok kóduje jednu z použitých štatistík (napríklad priemerná výška). Autori považujú za výhody tejto metódy práve tvar priestoru (pohľad zhora), s ktorým neurónová sieť pracuje, nakoľko práve tento priestor je vhodnejší a presnejší pre autonómne riadenie ako pohľad zachytený pomocou kamier. Taktiež obrovská výhoda tejto metódy je možnosť vyhodnotiť 360° svojho okolia pri použití otáčajúceho sa systému LiDAR. Táto metóda roztriedi mračno bodov do 2D poľa, kde jedna bunka reprezentuje priestor $0.1\text{cm} \times 0.1\text{cm}$. Tento vstup je následne spracovaný konvolučnou neurónovou sieťou, ktorej výstup určuje príslušnosť každej bunky ku vozovke.

Nakoľko v mojej práci ide o použitie pre účely údržby vozovky, pre riešenie bola vybraná metóda navrhnutá Caltagironem et al. [7]. Výhody kombinácie tejto metódy a daného použitia sú rovnomerná a zároveň relatívne vysoká presnosť v celom vyhodnocovanom priestore, ako aj možnosť ľahko a presne spočítať plochu vozovky alebo jej okolia.

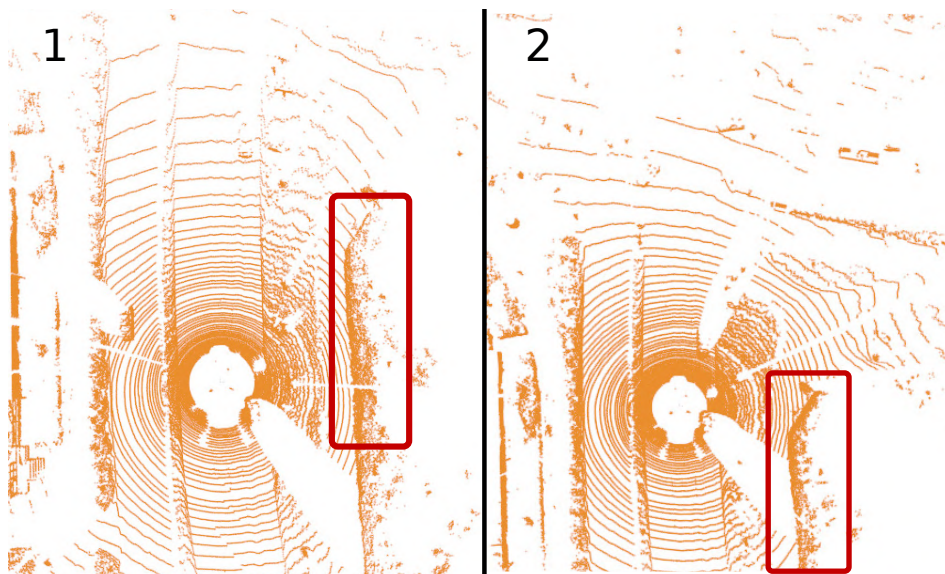
Kapitola 5

Návrh systému

Pre detekciu vozovky v LiDAR dátach bola zvolená metóda založená na plne konvolučných sieťach, ktorú navrhli Caltagirone et al. [7].

5.1 Použité dáta

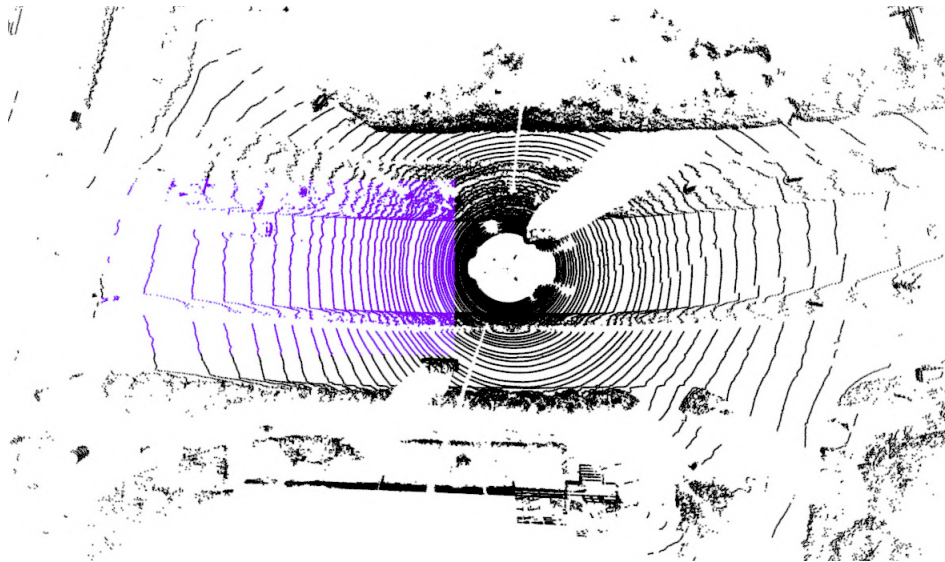
Na účel trénovania neurónovej siete bol použitý dataset, ktorý je súčasťou benchmarku KITTI Road/Lane Detection Evaluation [13]. Ide o dataset zložený z 289 vzoriek, obsahujúci tri typy vzoriek: mestská oblasť bez vodorovného dopravného značenia, mestská oblasť s vodorovným dopravným značením a mestská oblasť so zložitým vodorovným dopravným značením. Snímky jedného typu sú vytvorené ako sekvencia snímaná počas jazdy, pričom medzi niektorými snímkami je možné vidieť spoločné črty snímaného priestoru. Ak by tieto spoločné črty snímok boli príliš výrazné, mohlo by to ovplyvniť trénovanie neurónovej siete, nakoľko by sa v dataseete nachádzala určitá forma duplicit. Medzi laserovými skenmi, ktoré sú zobrazené na obrázku 5.1, je možné vidieť spoločné črty dvoch laserových snímok. Medzi danými skenmi je dostatočný rozdiel nato, aby nebol obmedzený proces trénovania neurónovej siete.



Obr. 5.1: Príklad dvoch po sebe nasledujúcich snímok mračien

5.2 Pretvorenie mračna bodov na tvar vhodný pre vstupnú vrstvu plne prepojenej konvolučnej neurónovej siete

Použitá neurónová sieť používa 6 kanálový vstup. Jeden kanál tohto vstupu má tvar 2D matice. Tieto matice obsahujú štatistické informácie o určitej časti laserového skenu. Body, z ktorých sú tieto informácie získavané, sú zobrazené na obrázku 5.2 modrou farbou. Táto oblasť má tvar 40×20 metrov. Jej začiatok sa nachádza 6 metrov pred systémom LiDAR a rozprestiera sa 10 metrov do oboch strán.



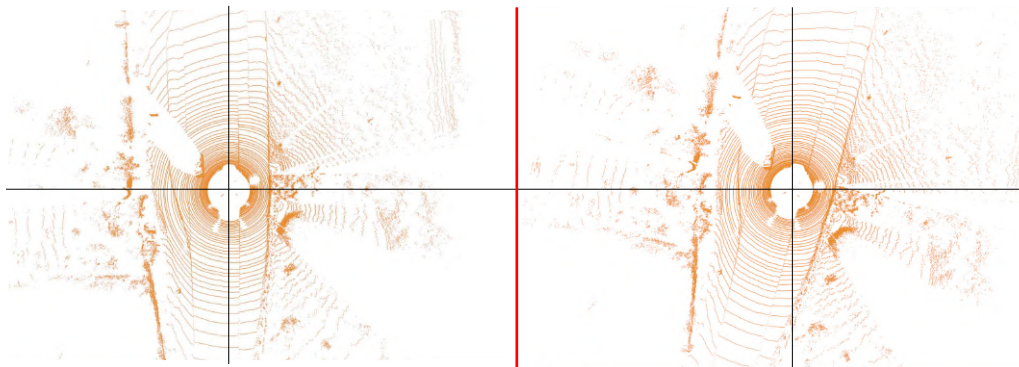
Obr. 5.2: Znáznornenie oblasti, z ktorej sú vytvorené vstupy pre neurónovú sieť

Body na obrázku 5.2 označené modrou farbou sú ďalej roztriedené do poľa s rozmermi 400×200 , kde plocha $10\text{cm} \times 10\text{cm}$ predstavuje jeden prvok poľa.

Po tomto kroku je v každom prvku poľa jeden, niekoľko alebo žiadny bod. Z vytvoreného poľa je vygenerovaných 6 štatistík (priemerná výška bodov, maximálna výška bodov, minimálna výška bodov, štandardná odchýlka výšky bodov a priemerná odrazivosť snímaných bodov). Je vytvorených 6 polí s rozmermi 400×200 , pričom každý kanál neurónovej siete prijíma jedno z vygenerovaných polí.

5.3 Zväčšenie datasetu použitého pre neurónovú sieť

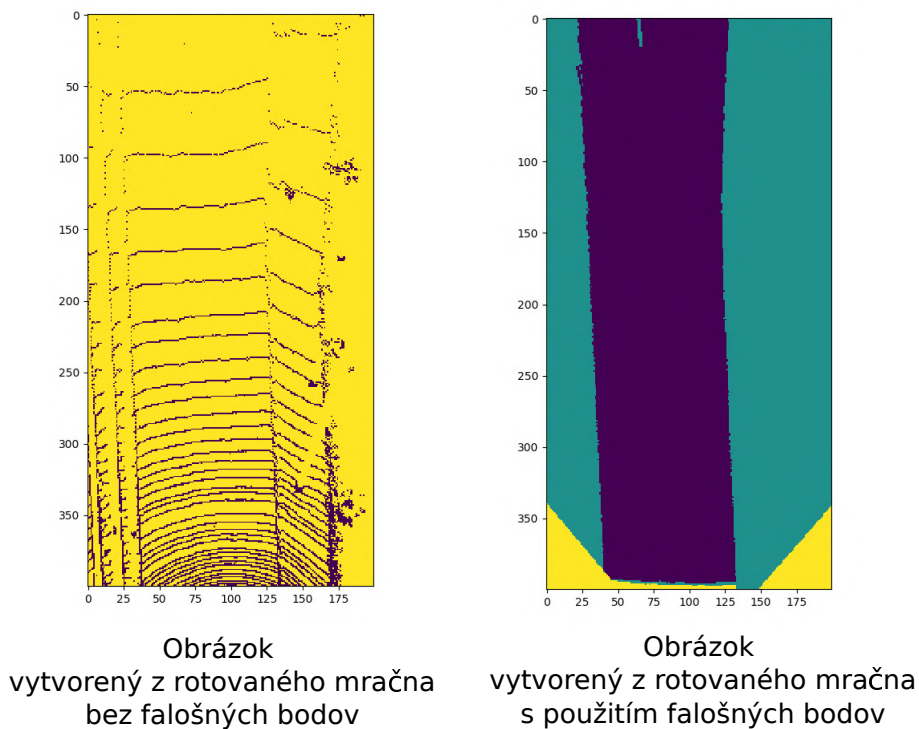
Nakoľko dataset o veľkosti 289 vzoriek nie je vhodný pre efektívne a kvalitné tréningovanie neurónovej siete, bolo potrebné tento dataset zväčšiť. Pre toto zväčšenie sú použité rotácie pôvodných 289 vzoriek datasetu, ako je vidno na obrázku 5.3, kde sa v ľavej časti obrázka nachádza vizualizácia mračna bodov pred rotáciou a v pravej časti obrázka je vizualizované mračno po rotácii o 6° vzhľadom na plochu, ktorú pretínajú priamky x a y (horizontálna a vertikálna priamka prechádzajúca stredom obrázka).



Obr. 5.3: Príklad rotácie jednej vzorky datasetu o 6°

5.3.1 Projekcia ground truth obrázkov do mračien bodov

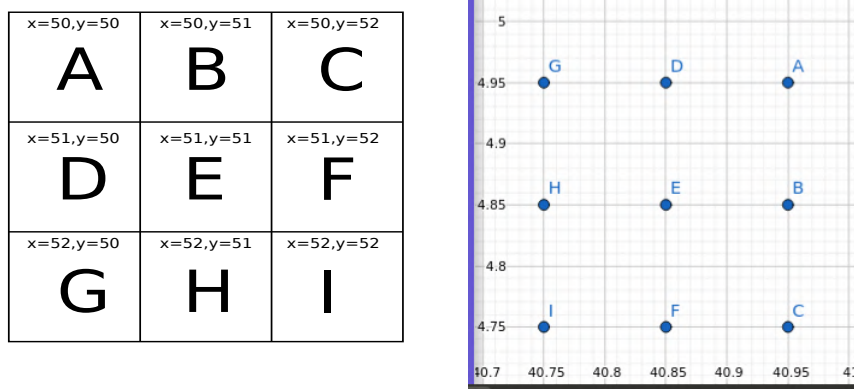
Aby bolo možné novovzniknuté vzorky použiť pre tréovanie neurónovej siete, je potrebné, aby každá nová vzorka mala svoj zodpovedajúci ground truth obrázok. Pre dosiahnutie tohto cieľa bol každému bodu nachádzajúcemu sa v oblasti záujmu (označené modrou farbou na obrázku 5.2) určený zodpovedajúci pixel v ground truth obrázku, a následne bola informácia o triede tohto pixelu priradená danému bodu. Po rotácii týchto mračien sú vygenerované vstupy pre sieť a ich príslušné ground truth obrázky. Pri pokusoch o vytvorenie ground truth obrázkov bol objavený problém týkajúci sa hustoty mračna bodov. Mračno totižto nie je také husté, aby bolo možné projektovať každý pixel z pôvodného ground truth obrázka. V ľavej časti obrázka 5.4 je možné vidieť hustotu novo vytvoreného ground truth obrázka, žltá farba značí absenciu hodnoty.



Obr. 5.4: Znázornený problém s hustotou ground truth obrázkov

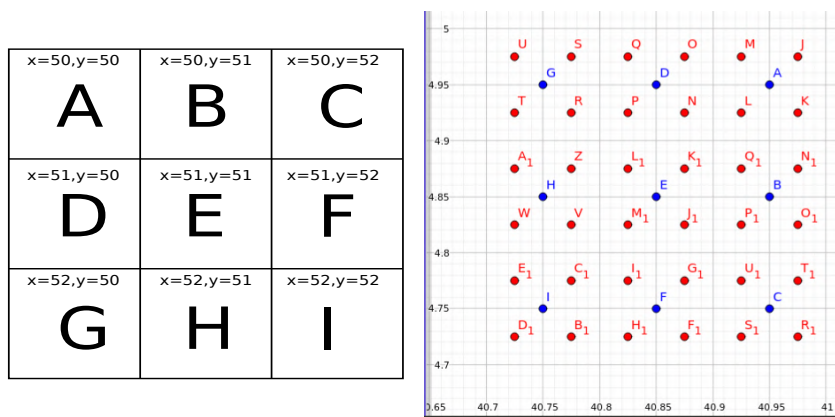
Tento problém bol vyriešený pridaním falošných bodov do mračna (pravá časť obrázka 5.4). Z každého elementu z pôvodného ground truth obrázka bolo pomocou vytvorených mapovacích funkcií, premietnutých niekoľko falošných bodov do nového mračna. Nakoľko by tieto body spôsobovali problém pri vytváraní vstupov pre sieť, keďže by mohli narúšať hodnoty jednotlivých štatistík (napríklad hustota bodov v danej oblasti), bolo potrebné ku každému novému bodu pridať informáciu o tom, že tento bod sa nesmie brať do úvahy pri výpočtoch štatistík.

Obrázok 5.5 popisuje premietanie jednotlivých elementov v poli ground truth obrázkov do laserového skenu. Prvá súradnica poľa je namapovaná pomocou mapovacej funkcie $f(x) = -0.1 * x + 49.95$ na prvý parameter budúceho falošného bodu (súradnica na osi x). Druhá súradnica poľa je rovnako namapovaná pomocou mapovacej funkcie $f(x) = -0.1 * j + 9.95$ na druhý parameter falošného bodu (súradnica na osi y).



Obr. 5.5: Príklad mapovania jednotlivých elementov poľa do priestoru laserového skenu

Nakoľko sú mračná bodov rotované, pri mapovaní boli pôvodné body rozptýlené do svojho okolia rovnakým spôsobom, aký je znázornený na obrázku 5.6, kde každý element poľa je premietnutý do laserového skenu spolu 5-krát (na obrázku 5.6 jeden modrý bod a 4 červené v jeho okolí).



Obr. 5.6: Príklad mapovania jednotlivých elementov poľa do priestoru laserového skenu s rozptýlením bodov

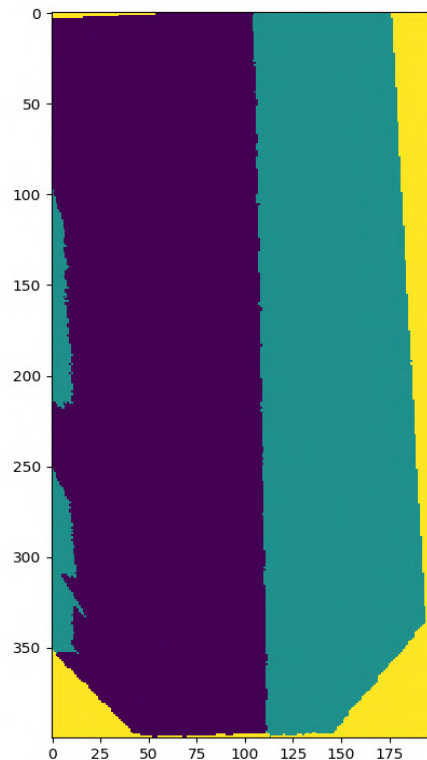
Výsledok tohto prístupu je možné vidieť na obrázku 5.4 v pravej časti obrázka.

5.3.2 Rotácia mračien bodov

Pre zväčšenie datasetu bola každá vzorka z pôvodného datasetu rotovaná 20 krát. Uhol pre rotácie bol vybraný z intervalu $\langle -30^\circ; 30^\circ \rangle$ s krokom 3° . V tomto okamihu existovalo 21 verzií každého mračna (rotované verzie + pôvodné mračno). Príklad rotácie mračna bodov je možné vidieť na obrázku 5.3. Príklad rotácie ground truth obrázka je možné vidieť na obrázku 5.7.

V nasledujúcom kroku boli z rotovaných mračien extrahované vstupy pre sieť, a taktiež daným vstupom zodpovedajúce ground truth obrázky.

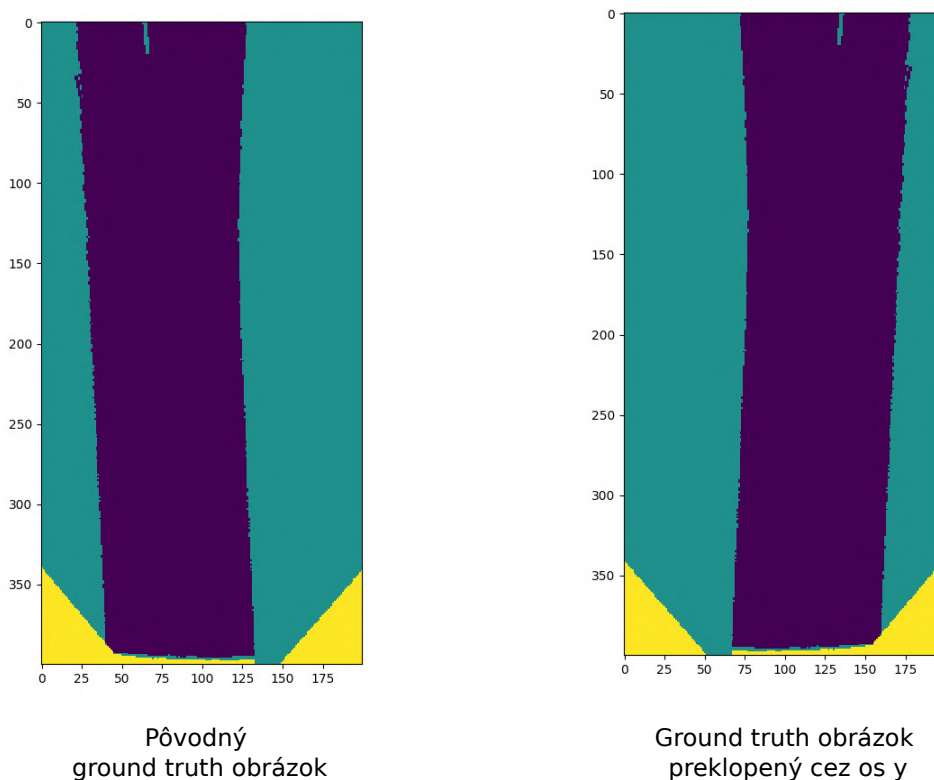
V pravej časti obrázka 5.4 je možné vidieť ground truth obrázok, vytvorený z mračna bodov, ktoré nebolo rotované (resp. bolo rotované o 0°). Na obrázku sa nachádzajú tri hodnoty (tri triedy): hodnota pre vozovku (tmavo-fialová, trieda A), hodnota pre priestor mimo vozovky (zeleno-modrá, trieda B) a tretia hodnota (žltá, trieda C) značí triedu, pri ktorej nebolo možné premietnuť hodnoty z ground truth obrázkov priradených k obrázkom z kamery (pretože LiDAR zachytí 360° svojho okolia, kým kamera zachytí len zlomok tohto okolia). Obrázok 5.7 znázorňuje ground truth obrázok vytvorený z rotáciou transformovaného mračna bodov. Žltá farba v tomto prípade značí elementy, ku ktorým nebola pridružená hodnota príslušnej triedy (v mračne sa nenachádzal v týchto miestach dostatok bodov). Miesta, kde pre rotáciu mračna nebolo možné určiť hodnotu príslušajúcej triedy, sú nastavené na rovnakú hodnotu ako je hodnota triedy C.



Obr. 5.7: Ground truth obrázok použitý pre trénovanie, vytvorený z transformovaného mračna bodov

Preklopenie vzoriek cez os y

Z 21 dvojíc laserových skenov a ground truth obrázkov bolo vytvorených ďalších 21 dvojíc. Jedná sa o cez vertikálnu os (os y) preklopené verzie laserových skenov a im prislúchajúcich ground truth obrázkov. Táto operácia zdvojnásobí veľkosť datasetu, ktorý je v konečnom dôsledku 42-krát väčší ako jeho pôvodná verzia. Príklad vizualizácie preklopenia ground truth obrázkov je možné vidieť na obrázku 5.8.

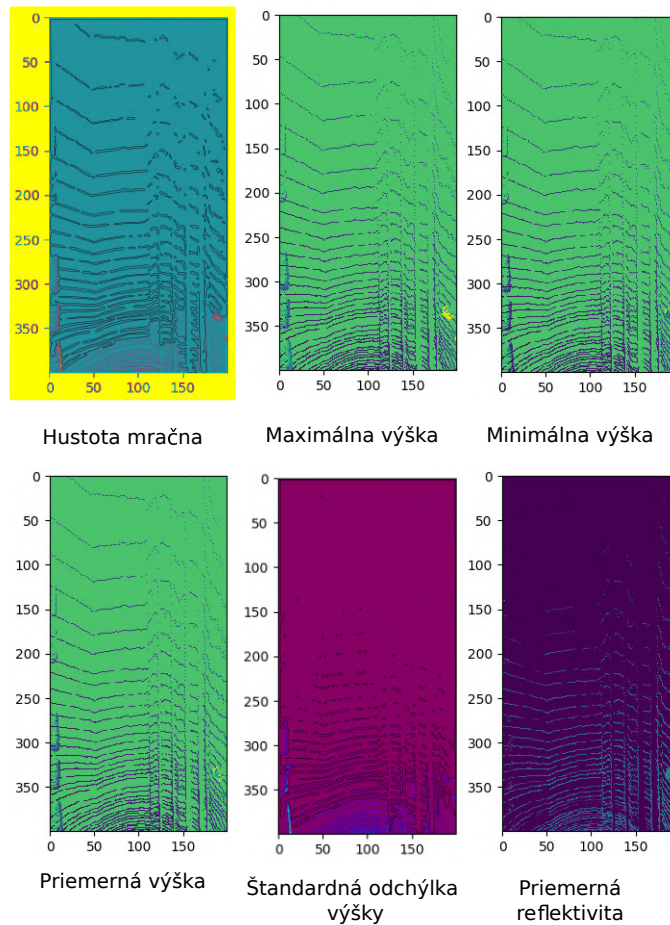


Obr. 5.8: Ground truth obrázok a jeho cez os y preklopená verzia

5.3.3 Vytvorenie vstupu z transformovaných mračien

Z každého transformovaného mračna bolo nutné okrem ground truth obrázkov, extrahovať tiež vstup pre tréovanie siete. Vstup pre sieť je tvorený šiestimi štatistikami o bodoch nachádzajúcich sa v oblasti, kde súradnica x patrí intervalu $< 6; 46 >$ a súradnica y patrí intervalu $< -10; 10 >$. Táto oblasť je rozdelená pomyselnou mriežkou do 2D poľa s veľkosťou 400×200 . Mračno je teda rozdelené na 80 000 malých štvorcov o veľkosti $0.1\text{cm} \times 0.1\text{cm}$ pri pohľade zhora. Body sú roztriedené do jednotlivých buniek v poli. Každému bodu v definovanom intervale pripadá určité miesto v poli. Následne sú o každom elemente poľa (jeden element bežne obsahuje nula až stovky bodov) vypočítané nasledovné štatistiky. Počet bodov, priemerná výška bodov, maximálna výška bodov, minimálna výška bodov, priemerná reflektivita bodov a štandardná odchýlka výšky bodov. Tieto štatistiky sú ukladané do rovnako veľkých poľ 400×200 , pričom každému elementárnemu priestoru v mračne bodov prislúcha 6 vytvorených štatistík. Pri počítaní štatistík je potrebné brať ohľad na pridané body počas projekcie ground truth obrázkov do mračna bodov - tieto body nesmú ovplyvniť výsledné štatistiky. Príklad generovania štatistík je možné vidieť

na obrázku 5.9. Na tomto obrázku boli časti zobrazujúce hustotu mračna a štandardnú odchýlku výšky upravené pomocou filtrov pre lepšiu viditeľnosť.

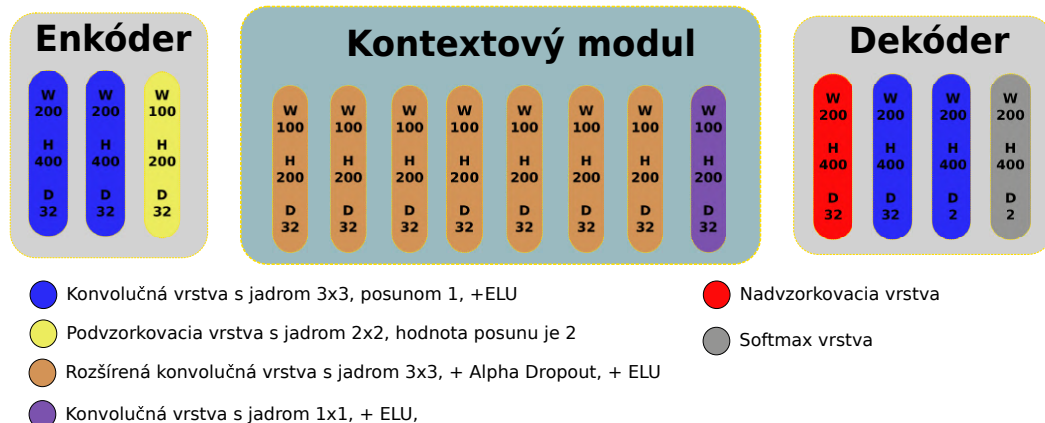


Obr. 5.9: Zobrazenie generovaných štatistík z mračna bodov

5.4 Použitý model neurónovej siete

Model neurónovej siete bol autormi metódy špeciálne navrhnutý práve pre úlohu pixelovej sémantickej segmentácie. Na obrázku 5.10 je zobrazená sieť použitá v tejto práci. *Vstupná vrstva siete* má šesť kanálov, pričom každý kanál prislúcha každej použitej štatistike (bod 5.1.1). *Enkodér* prijíma 6 kanálový vstup o veľkosti 400×200 , s účelom podvzorkovania vrstiev vstupov, aby sa redukovali požiadavky na pamäť plne prepojenej konvolučnej siete. Podvzorkovanie je vykonávané pomocou prevzorkovacej vrstvy typu max pooling. Jeho výstupy pokračujú ku *kontextovému modulu*, ktorého úlohou je agregácia viacúrovňového kontextu, pomocou rozšírenej konvolúcie. Nasleduje *dekodér*, slúžiaci pre nadvzorkovanie vrstiev vstupov na veľkosť, ktorá bola prítomná na vstupe. Nadvzorkovanie je vykonávané pomocou prevzorkovacej vrstvy typu max unpooling nasledovanou dvomi konvolučnými vrstvami. Posledným dielom siete (na obrázku 5.10 je posledná vrstva siete súčasťou dekodéra - šedá farba) je *softmax vrstva*, vracajúca mapy dôveryhodnosti tried, kde hodnota každého elementu predstavuje pravdepodobnosť, že daný vstup patrí k danej triede.

Krok vo všetkých konvolučných vrstvách nastavený na hodnotu 1. Krok v prevzorkovacích vrstvách bol nastavený na hodnotu 2. Pravdepodobnosť rušiacich vrstiev bola nastavená na hodnotu 0.25. Výstup každej konvolučnej vrstvy je predaný aktivačnej funkcii typu ELU. Pri popise architektúry je uvažovaný vstup pre neurónovú sieť s veľkosťou 400×200 .



Obr. 5.10: Model plne prepojenej konvolučnej neurónovej siete použitej v práci

Prvá konvolučná vrstva enkodéru kóduje 6 kanálový vstup neurónovej siete na 32 kanálový výstup. Táto, a rovnako aj ďalšia konvolučná vrstva používajú filtračné matice s veľkosťou 3×3 a pre zväčšenie vstupných matíc používajú parameter 1×1 (jeden pridaný riadok na každú stranu vstupnej matice). Výstup druhej konvolučnej vrstvy je následne pomocou prevzorkovacej vrstvy typu maximum pooling prevzorkovaný na rozmer 200×100 (toto je zároveň výstup enkodéra). Popisovaná prevzorkovacia vrstva používa filtračnú maticu s veľkosťou 2×2 , pričom táto vrstva ukladá pozície maximálnych hodnôt.

Kontextový modul tvorí 8 konvolučných vrstiev. Prvých 7 vrstiev používa filtračné matice s veľkosťou 3×3 a posledná vrstva používa filtračnú maticu s veľkosťou 1×1 . Za každou touto konvolučnou vrstvou (na obrázku 5.10 hnedá farba) nasleduje rušiaca vrstva. Prvá konvolučná vrstva používa pre zväčšenie vstupnej matice parameter 1×1 . Táto vrstva kóduje 32 kanálový vstup na 128 kanálový výstup. Druhá až siedma konvolučná vrstva kontextového modulu používajú pre zväčšenie vstupnej matice a pre rozšírenie filtračnej matice rovnaké parametre a to: $2^{n-2} \times 2^{n-1}$, kde n značí poradie vrstvy v kontextovom module. Siedma vrstva kontextového modulu teda používa parametre 32×64 . Posledná ôsma vrstva používa filtračné matice s veľkosťou 1×1 . Táto vrstva kóduje 128 kanálový vstup na 32 kanálový výstup. Výstup tejto vrstvy je výstupom kontextového modulu.

Prvou vrstvou dekodéra je prevzorkovacia vrstva typu maximum unpooling, pričom táto vrstva mení rozmery vstupných matíc z 200×100 na 400×200 . Nasledujúca konvolučná vrstva používa filtračnú maticu s rozmermi 3×3 a parameter vstupu pre ohraničenie je 1×1 . Posledná konvolučná vrstva používa rovnaké filtračné matice, a taktiež rovnaký parameter pre ohraničenie vstupu, no jej hlavnou úlohou je dekodovať 32 kanálový vstup na 3 kanálový výstup.

Kapitola 6

Implementácia

Táto kapitola popisuje tvorbu datasetu a jeho použitie na tréovanie konvolučnej neurónovej siete. Na tréovanie neurónovej siete, bol použitý projekt uverejnený na stránkach vast.ai, kde je možné prenajať si server s grafickou kartou podľa výberu. Pre vytvorenie základnej štruktúry priečinkov a inštaláciu potrebných modulov bol vytvorený skript `installPackages.sh`. Tento skript bol použitý vždy ako prvý pri práci na prenajatom serveri. Zároveň bol vytvorený skript `experiment.sh`, ktorý sa staral o vytvorenie datasetu na prenajatom serveri a následné tréovanie neurónovej siete.

Pre tvorbu neurónovej siete bol zvolený modul jazyka Python s názvom Pytorch. Z tohto modulu boli využité balíky `torch` a `torch.nn`. Model konvolučnej neurónovej siete je podtriedou triedy `Module` z použitého balíka `torch.nn`.

6.1 Príprava a spracovanie datasetu

Pôvodné vzorky datasetu vo formáte `.bin`, sú prevedené na formát `.pcd` pomocou skriptu `binToPCL.sh`, ktorý postupne načíta všetky súbory zo zložky `binFiles` a pomocou `c++` programu `kitti2pcd` od autora Yani Ioannou¹ ich prevádza na formát `.pcd`.

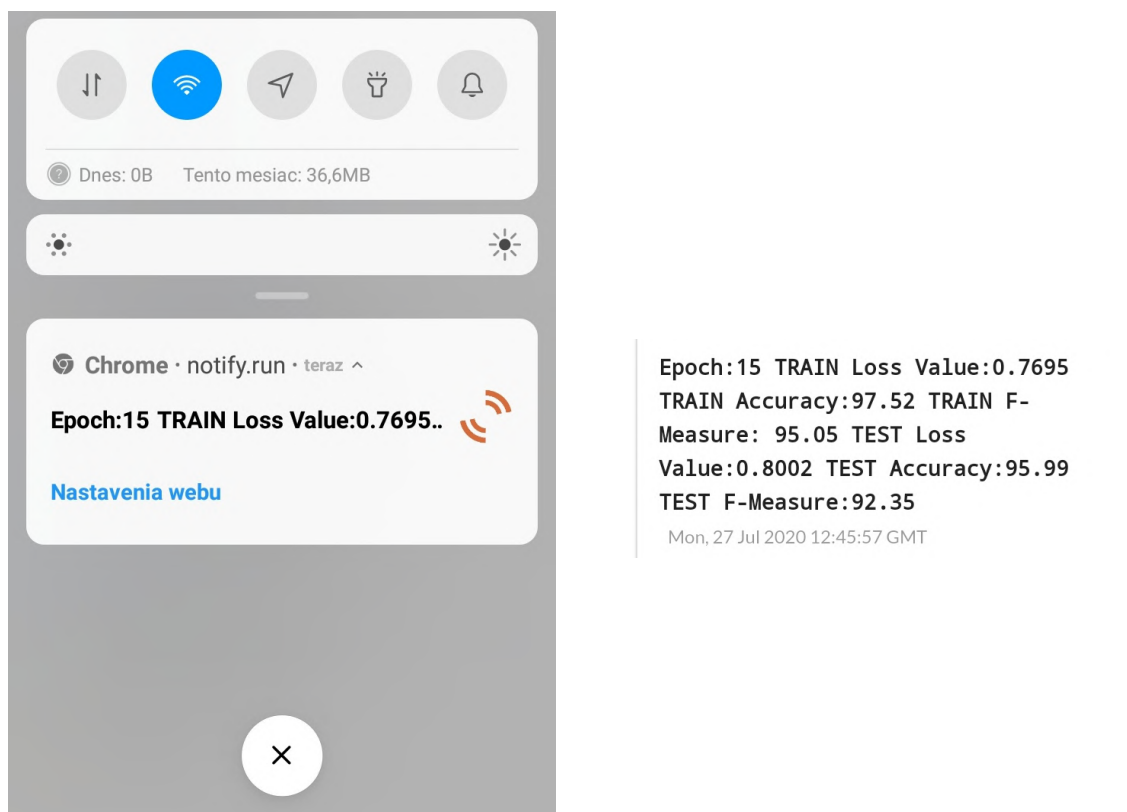
Pre transformáciu informácií z ground truth obrázkov do laserových skenov bol použitý program `createRotatedPcl.py`. Tento program získa obsah priečinka s názvom `pclFiles` a pre každý laserový sken, ktorý sa vo vnútri tohto priečinka nachádza, prevedie popisované priradenie informácií z ground truth obrázkov do laserových skenov. Laserové skeny obsahujúce informácie o ground truth obrázkoch ukladá tento program do priečinka s názvom `pclFiles/pclFilesWithClasses`.

Následne je počas vykonávania programu `createRotatedPcl.py` použitý `c++` program `pclRotator` pre transformáciu mračna bodov pomocou knižnice Point Cloud Library (PCL), ktorý načíta uložený súbor z priečinka `pclFiles/pclFilesWithClasses` a vytvorí jeho 21 transformovaných verzií, ktoré sú uložené do priečinka `pclFiles/rotatedPCL`. Po skončení behu programu `createRotatedPcl.py` je pre generovanie tenzorov spustený program s názvom `createNetworkInputAndGT.py`, ktorý z každého súboru nachádzajúceho sa v priečinku `pclFiles/rotatedPCL` vytvorí vstupný tenzor pre sieť a jemu zodpovedajúci výstupný tenzor. Pri ukladaní týchto vstupných a výstupných tenzorov sú vykonané preklopenia cez os `y` daných tenzorov a tieto preklopené verzie tenzorov sú taktiež uložené.

¹www.github.com/yanii/kitti-pcl

Proces tréovania na prenajatom serveri

Dataset bol vždy znova vytvorený z pôvodných 289 vzoriek, nakoľko ide o jednoduchšiu a lacnejšiu možnosť, ako dataset vždy stiahnuť na server. Na konci vykonávania každého programu, som využil projekt dostupný na stránkach notify.run, pomocou ktorého boli do osobného smartfónu posielané push-notifikácie o aktuálnom dianí na serveri. Príklad tohto upozornenia je možné vidieť v ľavej časti obrázka 6.1. Po otvorení tohto upozornenia sú zobrazené dáta uvedené v pravej časti obrázka 6.1.



Obr. 6.1: Príklad prijatej notifikácie

6.2 Tréovanie neurónovej siete

Samotný model neurónovej siete bol počas začiatku práce, teda počas hľadania logických chýb, tréovaný na malom datasete, čo umožňovalo použitie priemernej grafickej karty (Geforce GTX 960M). Opäť počas hľadania logických chýb (nastavenia serveru, problémy s datasetom a pod.), bol zvolený menej výkonný variant (a teda aj lacnejší) GTX 1070Ti. Pre potreby skutočného tréovania bola zvolená grafická karta RTX 2080Ti. Počas tréovania bolo taktiež využívané posielanie push-notifikácií do osobného smartfónu. Tieto informácie poskytovali prehľad o aktuálnej presnosti danej neurónovej siete. Pre výpočet presnosti neurónovej siete bol používaný postup implementovaný v programe `accuracyCalc.py`. Bol použitý Adam optimalizačný algoritmus. V modeli neurónovej siete boli použité rušiacе vrstvy z balíka `torch` typu `AlphaDropout`.

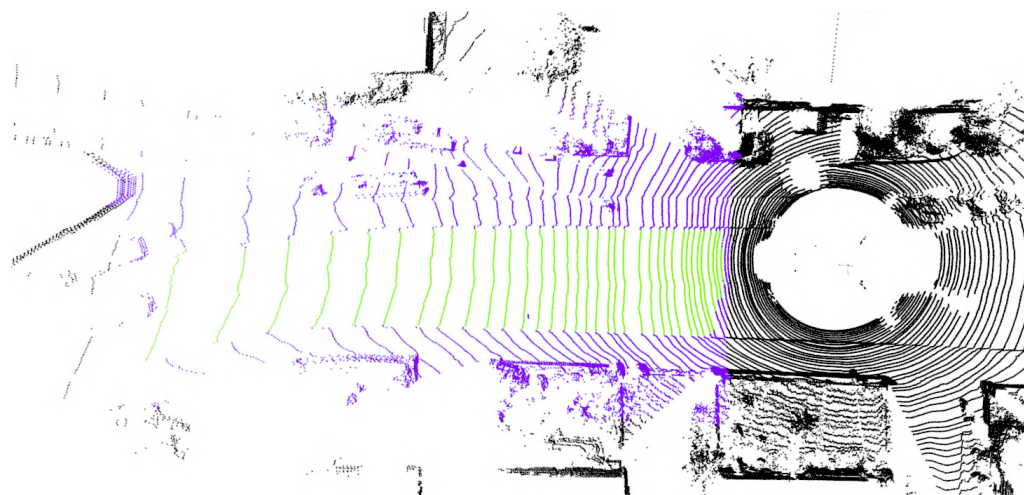
Po použití $n - 2$ (n značí počet jadier dostupného procesora) jadier procesora na serveri bolo zaznamenané obrovské zrýchlenie v oblasti prípravy datasetu. Taktiež markantné

zrýchlenie bolo dosiahnuté nástrojom CUDA, pomocou ktorého bola na výpočty medzi tenzormi využitá grafická karta namiesto procesora.

Na tréovanie neurónovej siete bol vytvorený program `baseCNN.py`, ktorého vykonávanie je možné prerušiť príkazom `touch stop` v základnom adresári projektu (dosiahnutý stav je po zastavení možné znova načítať z uloženého súboru). Program pri nájdení súboru s názvom `stop` zastaví tréovanie a uloží natréovanú neurónovú sieť do súboru `Model.tar`. Taktiež uloží dáta o priebehu tréovania neurónovej siete do súboru `results.txt`. Pre zálohovanie súborov `Model.tar` a `results.txt` boli použité skripty s názvami `sendResults.sh` a `sendModel.sh` (dáta boli zálohované na školský server).

6.3 Použitie natréovanej neurónovej siete

Pre použitie neurónovej siete je vytvorený program `useModel.py`, ktorý dokáže spracovať mračná bodov vo formáte `.pcd`, ale aj `.bin`. Výstup neurónovej siete je následne spracovaný a uložený vo formáte zafarbeného mračna bodov do súboru `result.pcd` (zelená farba značí vozovku, modrá priestor mimo vozovky a čierna priestor, ktorý nebol vyhodnocovaný). Tento výsledok je možné následne zobrazit pomocou `c++` programu `pclResultView`. Príklad zobrazeného výsledku je možné vidieť na obrázku 6.2.



Obr. 6.2: Znáznorenie výsledku segmentácie vozovky

Kapitola 7

Experimenty

Implementácia neurónovej siete bola otestovaná na rôzne modifikácie. Experimenty s implementáciou neurónovej siete mali za cieľ zvýšiť výsledky neurónovej siete. Experimenty s neurónovou sieťou boli vykonané s datasetom, ktorý je súčasťou benchmarku KITTI Road/Lane Detection Evaluation [13]. Ground truth obrázky pre základných 289 vzoriek poskytol po krátkej e-mailovej komunikácii jeden z autorov použitej metódy - Luca Caltagirone. Pre lepšie možnosti vyhodnotenia jednotlivých modifikácií, boli takmer všetky experimenty prevedené v troch behoch. Jeden beh experimentu predstavuje jedno trénovanie neurónovej siete od úplného začiatku. Experimenty, ktoré už po prvom behu ukazovali očividne negatívny výsledok, boli prevedené iba jedenkrát. Neurónová sieť bola trénovaná pomocou dvadsiatich trénovacích cyklov, čo na prenájatom serveri trvalo priemerne 3.5 hodiny. Všetky behy trénovania neurónových sietí začali s parametrom rýchlosti učenia nastaveným na hodnotu 0.0001. Pre dotrénovanie neurónovej siete bol kontrolovaný pokrok neurónovej siete, ktorý po zastavení spôsobil zníženie parametra rýchlosti učenia na jednu pätinu jeho pôvodnej hodnoty.

7.1 Rozdelenie datasetu pre trénovanie a testovanie

V skorom štádiu práce bol dataset rozdelený spôsobom, kedy každá desiatka vzorka bola použitá pre testovanie a zvyšné vzorky pre trénovanie. Nakoľko sa ale v datasete nachádza istá forma duplicit (kapitola 5.1), bolo nutné dataset systematickejšie usporiadať. Pre ukážku rozdelenia datasetu je uvažovaný dataset bez transformácií, teda 289 vzoriek, obsahujúci tri typy vzoriek: mestská oblasť bez vodorovného dopravného značenia, mestská oblasť s vodorovným dopravným značením a mestská oblasť so zložitým vodorovným dopravným značením. Z každej triedy vzoriek bolo prvých 20 vzoriek určených na testovanie a zvyšok triedy bol použitý na trénovanie. V použitom datasete (s transformovanými snímkami) na trénovanie siete, boli vzorky do skupiny na trénovanie alebo testovanie rozdelené spôsobom, kedy mračno bolo pridelené skupine spolu so všetkými svojimi rotovanými alebo prevrátenými (cez vertikálnu os) verziami. Celková veľkosť datasetu je teda 12180 vzoriek. 2520 vzoriek je použitých na testovanie výsledkov, ktoré neurónová sieť dosahuje, a tiež na detekciu pretrénovania neurónovej siete. Zvyšné vzorky sú použité na trénovanie neurónovej siete. Počas trénovania ani raz nenastala situácia, kedy by došlo k pretrénovaniu.

7.2 Určenie presnosti neurónovej siete

Počas tréovania neurónovej siete je potrebné určiť jej aktuálny stav. Pre túto činnosť sa používa chybová matica, nachádzajúca sa na obrázku 7.1.

		Správne hodnoty	
		Trieda true	Trieda false
Klasifikované hodnoty	Trieda true	TP	FP
	Trieda false	FN	TN

Tabuľka 7.1: Znázornenie chybovej matice

Chybovú maticu počítame pre triedu vstupu (napríklad na obrázku 3.4 v ľavej spodnej časti sa nachádzali 3 triedy). V znázornenej tabuľke 7.1 sú spomínané parametre pomenované TP, TN, FP, FN. Parameter TP (z angl. true positive) značí počet dát, ktoré boli správne klasifikované ako daná trieda. Parameter TN (z angl. true negative) značí počet dát, ktoré boli správne určené ako dáta neprislúchajúce ku danej triede. Parameter FP (z angl. false positive) značí počet dát, ktoré boli nesprávne klasifikované ako daná trieda (často sa používa v angl. false alarm). Posledný parameter FN (z angl. false negative) značí počet dát, pre ktoré bola nesprávne určená nepríslušnosť ku danej triede (v angl. sa používa aj miss) [37].

V práci boli pre hodnotenie neurónovej siete použité dva parametre, úspešnosť a F-measure, vypočítané zo spomínanej chybovej matice. Úspešnosť, ktorá určuje, ako často daná sieť správne klasifikuje dáta sa počíta nasledovným spôsobom.

$$U = \frac{TP + TN}{TP + TN + FP + FN} \quad (7.1)$$

F-measure používa pre svoj výpočet parametre presnosť a senzitivitu. Hodnota presnosti určuje úspešnosť klasifikovania hodnôt ako danú triedu. Pre jej výpočet sa používa nasledovný vzorec [20].

$$P = \frac{TP}{TP + FP} \quad (7.2)$$

Hodnota senzitivity určuje, aká veľká časť hodnôt klasifikovaných ako daná trieda prislúcha ku danej triede [20].

$$S = \frac{TP}{TP + FN} \quad (7.3)$$

Parameter F-measure (nazývaný aj f1 score) vyjadruje harmonický priemer predchádzajúcich dvoch parametrov. Jeho výpočet je nasledovný [19].

$$F = \frac{2 * P * S}{P + S} \quad (7.4)$$

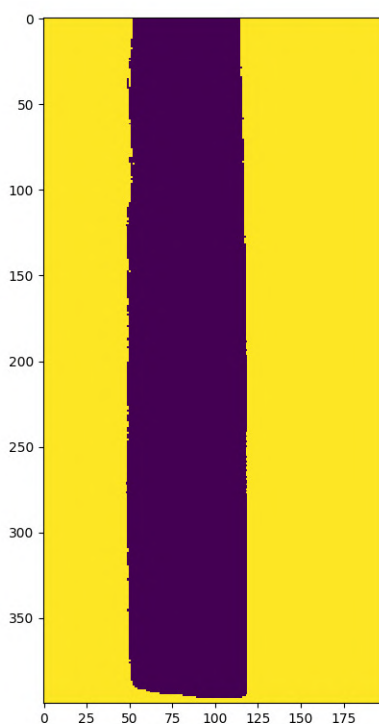
Počas vyhodnocovania úspešnosti neurónovej siete boli primárne používané údaje o úspešnosti a F-measure údaje. Výsledky neurónovej siete nie je možné úplne porovnávať s výsledkami metód dostupnými na stránkach KITTI, nakoľko neurónová sieť nebola testovaná na dataseťe používanom pre porovnanie výsledkov jednotlivých metód. Úspešnosť

a F-measure hodnota boli merané primárne na pôvodnej časti datasetu určenej na testovanie (60 netransformovaných vzoriek, s ktorými neurónová sieť nebola trébovaná), no hodnoty týchto parametrov boli sledované taktiež pre celú testovaciu časť datasetu. Nakoľko primárne vzorky datasetu a ich ground truth obrázky sú najpresnejšie, pre porovnanie modifikácií sú používané práve hodnoty pre pôvodnú časť testovacieho datasetu.

7.3 Vytvorenie primitívneho klasifikátora

Analýzou vzoriek pôvodného datasetu (bez transformácií a preklopení) bola zistená určitá podobnosť naprieč celým datasetom. Konkrétne ide o skutočnosť, že vozovka sa vždy nachádza pred vozidlom, a častokrát pokračuje rovno. Bol preto vykonaný pokus pre zistenie minimálnych hodnôt úspešnosti a hodnôt F-measure, ktoré by mala trébovaná neurónová sieť bez ťažkostí dosiahnuť.

Pre tieto účely bol vytvorený primitívny klasifikátor, pomocou určenia početnosti tried v jednotlivých bunkách ground truth obrázkov určených na trébovanie neurónovej siete. Následne bol vytvorený univerzálny výstup, kde každej výstupnej bunke bola priradená hodnota triedy s najvyššou početnosťou v danom mieste. Primitívny klasifikátor potom pre akýkoľvek vstup vráti spomínaný univerzálny výstup. Tento výstup sa nachádza na obrázku 7.1. Najlepšie hodnoty dosiahol primitívny klasifikátor pri trébovaní a testovaní na pôvodnom datasete.



Univerzálny výstup
vytvorený z 229 originálnych
vzoriek

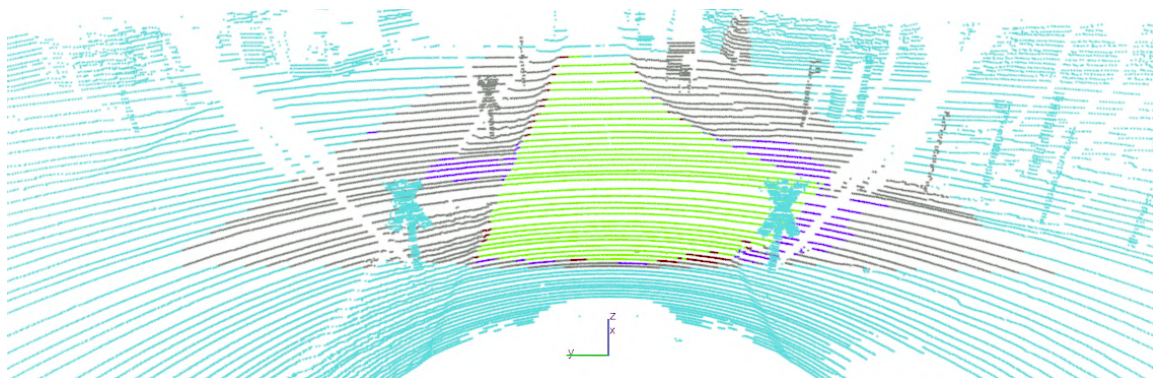
Obr. 7.1: Znázornenie univerzálného výstupu

Použitím tohto primitívneho klasifikátora boli dosiahnuté hodnoty 83.36% pre úspešnosť a 76.87% hodnota F-measure.

Pri zobrazení filtračných matíc vyplynulo, že tento primitívny klasifikátor pomerne správne určuje priestor mimo vozovku, no samotnú vozovku neurčuje nepresne. Tento jav mal za následok vysokú hodnotu úspešnosti no nízku hodnotu F-measure. Cieľom počas tréningovania neurónovej siete bolo dosiahnuť lepšie výsledky ako najlepší dosiahnutý výsledok primitívneho klasifikátora.

7.4 Výsledok prvého behu

Natrénovaním prvej neurónovej siete boli dosiahnuté hodnoty 92.58% pre F-measure a 94.71% pre úspešnosť. Na obrázku 7.2 je vidieť príklad použitia tejto natrénovanej neurónovej siete.



Obr. 7.2: Segmentácia mračna pomocou natrénovanej neurónovej siete, zelená farba značí správne klasifikovanú vozovku, šedá farba správne klasifikovaný priestor nepatriaci k vozovke, modrá farba značí zle určený priestor nepatriaci vozovke a červená farba značí priestor zle klasifikovaný ako časť vozovky

7.5 Experimenty s parametrami `weight` a `reduction` chybovej funkcie typu `CrossEntropyLoss`

Nakolko dataset obsahuje vzorky, v ktorých sa nachádzajú miesta, pre ktoré nebolo možné z rôznych dôvodov jednoznačne určiť ich príslušnosť k vozovke, boli prevedené experimenty s cieľom minimalizovať negatívny vplyv týchto neurčitých miest v datase. Toto je možné dosiahnuť pomocou zmeny spôsobu výpočtu chybovej funkcie. Táto zmena spočíva vo vynechaní chýb na miestach, pre ktoré nevieme určiť ich príslušnosť k vozovke. Pre tento účel existuje nástroj v Pytorch implementácií chybovej funkcie typu `CrossEntropyLoss`, ktorý umožňuje použitie parametru `weight`, ktorý nastavuje vplyv jednotlivých tried pre výslednú hodnotu chybovej funkcie. Následne je potrebné vyhodnotiť chybu naprieč prvkami, pričom najčastejšie sú dve riešenia: vyhodnotenie priemernej chyby naprieč prvkami a suma týchto chýb. V Pytorch implementácií je pre tento účel dostupný parameter `reduction`, ktorý určuje spôsob výpočtu výslednej hodnoty naprieč prvkami, pričom boli použité dva varianty: `mean` a `sum`. Variant `mean` spôsobí, že hodnota chybovej funkcie je tvorená priemerom chýb vo výstupe neurónovej siete vzhľadom ku ground truth obrázkom, kdežto variant `sum` spôsobí, že výsledná hodnota chybovej funkcie je suma týchto chýb.

	Parameter reduction	Prvý beh	Druhý beh	Tretí beh	Priemer
Bez par.	mean	92.58	93.23	93.36	93.057
weight	sum	93.66	92.70	91.33	92.56
S par.	mean	94.26	94.25	94.48	94.33
weight	sum	94.19	94.07	94.40	94.22

Tabuľka 7.2: Hodnoty F-measure dosiahnuté neurónovými sieťami s použitím modifikácií chybovej funkcie, vidno pozitívny vplyv použitia parametra `weight` a taktiež vidno pozitívny vplyv parametra `reduction` s hodnotou `mean`

Z tabuľky 7.2 je zrejmé, že použitie parametra `weight` má význam pre výslednú úspešnosť neurónovej siete, nakoľko väčšina hodnôt dosiahnutých v jednotlivých behov s použitím parametra `weight` má značne vyššiu úspešnosť ako každý z behov bez použitia parametra `weight`. Taktiež z uvedenej tabuľky vyplýva pozitívny vplyv na kvalitu neurónovej siete použitie parametra `reduction` s hodnotou `mean`.

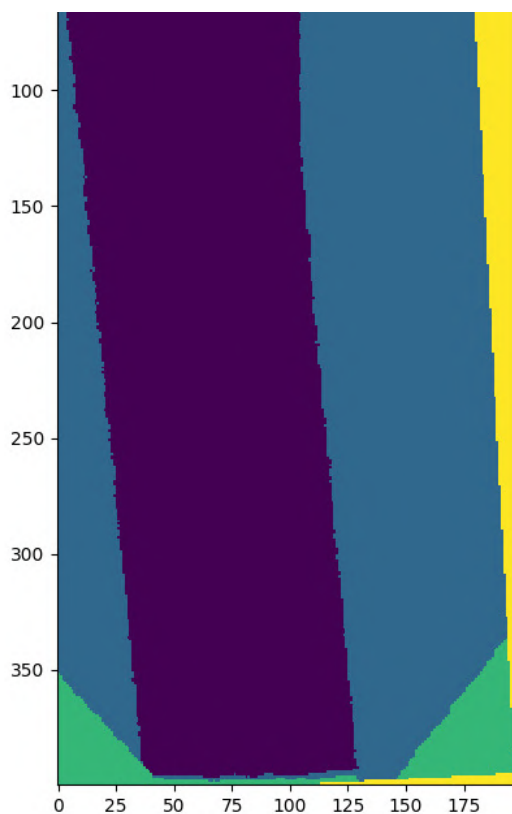
7.6 Experimenty s počtom použitých tried

Bol vykonaný experiment, kedy očakávaný výstup neurónovej siete (ground truth obrázky) nebol pre 3 triedy ako je to znázornené na obrázku 5.7. Tieto tri pôvodné triedy predstavujú priestor prislúchajúci vozovke, priestor mimo vozovku a priestor, pre ktorý nebolo možné určiť jeho príslušnosť k vozovke, či už kvôli rotáciám datasetu alebo absencii priestoru v kamerových snímkach prislúchajúcemu danému priestoru v laserovom skene. V popisovanej modifikácii riešenia bola posledná trieda rozdelená na dve triedy: priestor vzniknutý rotáciou mračna (žltá farba na obrázku 7.3) a priestor vzniknutý kvôli rozdielnej reprezentácii priestoru medzi kamerovými snímkami a laserovými skenmi (zelená farba na obrázku 7.3). Dôvod na toto rozdelenie používaných tried bola skutočnosť, že spodné rohy snímky sú vždy vyhodnotené ako priestor mimo vozovky, a teda pri použití parametra `weight` chybovej funkcie typu `CrossEntropyLoss`, je možné vynechať iba miesta vzniknuté rotáciou a spodné rohy obrázka sa môže sieť naučiť klasifikovať ako priestor mimo vozovku. Očakávaný výstup je znázornený na obrázku 7.3.

S použitím parametra `reduction` nastaveným na hodnotu `mean` a s použitím parametra `weight`, boli dosiahnuté výsledky uvedené v posledných dvoch riadkoch tabuľky 7.3. Tento parameter bol otestovaný pre dva spôsoby jeho použitia. Prvým spôsobom bolo použitie tohto parametra, pre zníženie hodnoty chybovej funkcie vzhľadom na priestory vzniknuté rotáciou (žltá farba na obrázku 7.3). Tento spôsob je v tabuľke 7.3 uvedený ako `weight=(1,1,1,0)`. Druhým spôsobom bolo použitie tohto parametra, pre zníženie hodnoty chybovej funkcie, vzhľadom na obe triedy, pri ktorých nebolo možné určiť ich príslušnosť k vozovke. Tento spôsob je v tabuľke 7.3 uvedený ako `weight=(1,1,0,0)`. Riadok `weight=(1,1,1,1)` znázorňuje výsledok neurónovej siete bez použitia parametra `weight`.

	Prvý beh	Druhý beh	Tretí beh	Priemer
<code>weight=(1,1,1,1)</code>	93.66%	94.18%	93.12%	93.65%
<code>weight=(1,1,0,0)</code>	93.95%	93.96%	93.73%	93.88%
<code>weight=(1,1,1,0)</code>	95.02%	94.95%	94.21%	94.73%
Porovnanie s tromi triedami	94.26%	94.25%	94.48%	94.33%

Tabuľka 7.3: Prvé tri riadky obsahujú hodnoty F-measure pre neurónové siete používajúce 4 triedy s rôznymi modifikáciami parametra `weight`, v poslednom riadku je pre porovnanie najlepšia verzia z predchádzajúceho experimentu, ako vidno v stĺpci *priemer*, nastalo zlepšenie približne o 0.4%

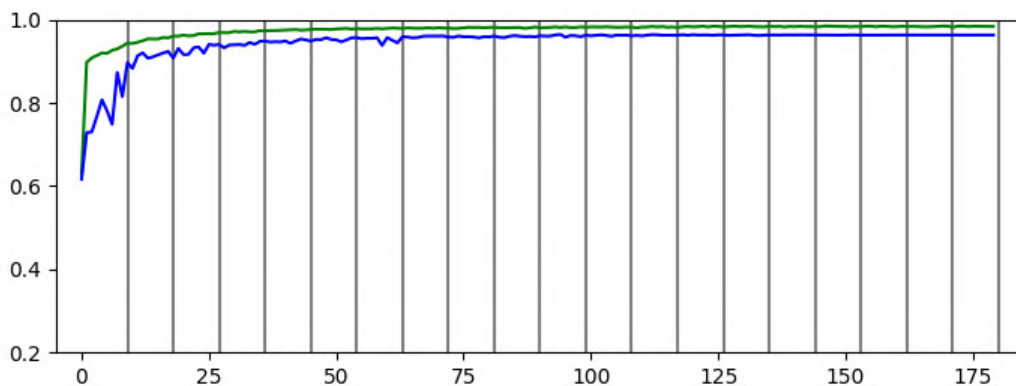


Obr. 7.3: Znáznorný očakávaný výstup s použitím neurónovej siete predikujúcej 4 triedy, fialová farba značí vozovku, modrá priestor mimo vozovky, zelená farba značí priestor, ktorému nebola priradená hodnota kvôli rozličnosti dát kamerových snímkov a laserových skenov a žltá farba značí priestor vzniknutý rotáciou mračna

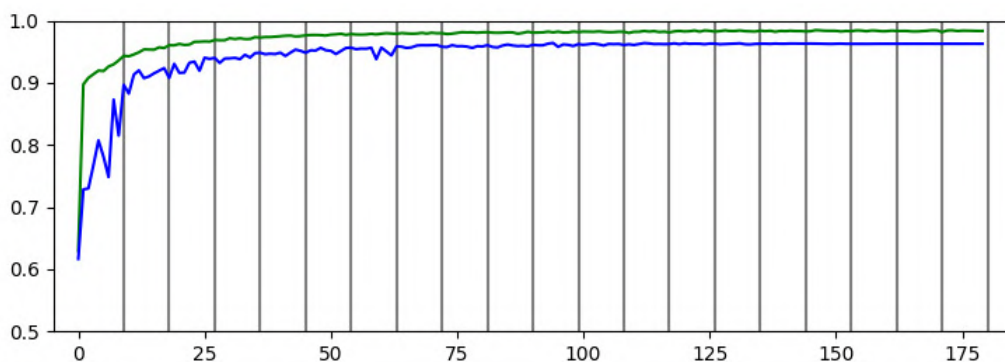
7.7 Najlepšie dosiahnuté výsledky tréovania neurónovej siete

Počas všetkých tréovaní neurónových sietí sa podarilo dosiahnuť najlepšie výsledky s neurónovou sieťou používajúcou chybovú funkciu typu `CrossEntropyLoss` s parametrom `reduction` nastaveným na hodnotu `mean` a parametrom `weight` použitým pre poslednú zo štyroch tried výstupu danej siete. Pre testovací dataset bola výsledná hodnota úspešnosti 96.12% a F-measure hodnota 95.02%. Tieto dosiahnuté výsledky sú veľmi uspokojivé pre daný účel práce. Z analýzy rôznych výsledkov bolo zistené, že vyššia hodnota úspešnosti

oproti hodnote F-measure je spôsobená výsledkami, kedy bol priestor zle klasifikovaný ako vozovka. Vývoj parametra F-measure, je zobrazený na obrázku 7.4. Na obrázku 7.5 je možné vidieť priebeh hodnôt presnosti počas tréovania.



Obr. 7.4: Znáznorenie hodnôt metriky F-measure, zelená krivka predstavuje dosiahnuté hodnoty pre tréovaciu sadu, modrá krivka zobrazuje hodnoty pre testovaciu sadu

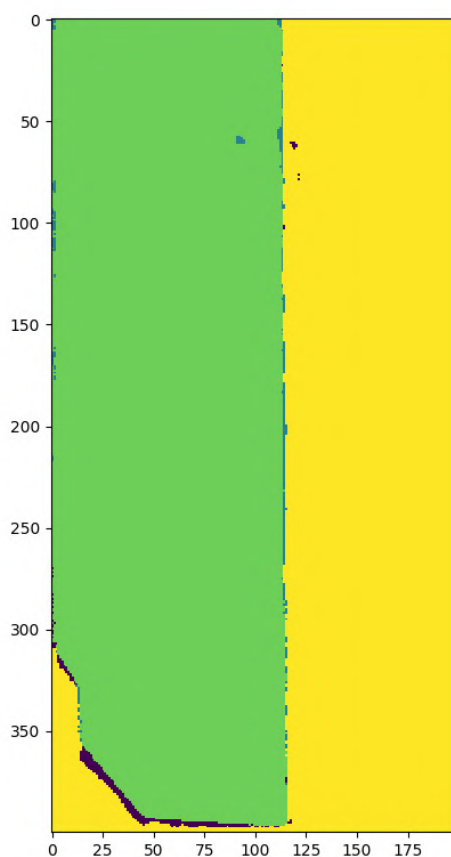


Obr. 7.5: Úspešnosť počas tréovania, zelená krivka predstavuje dosiahnuté hodnoty pre tréovaciu sadu, modrá krivka zobrazuje hodnoty pre testovaciu sadu

Neurónová sieť dosahuje na celom testovacom datasete, teda aj s transformovanými vzorkami úspešnosť 96.25% a F-measure hodnotu 92.84%.

Najlepšie segmentované mračno bodov

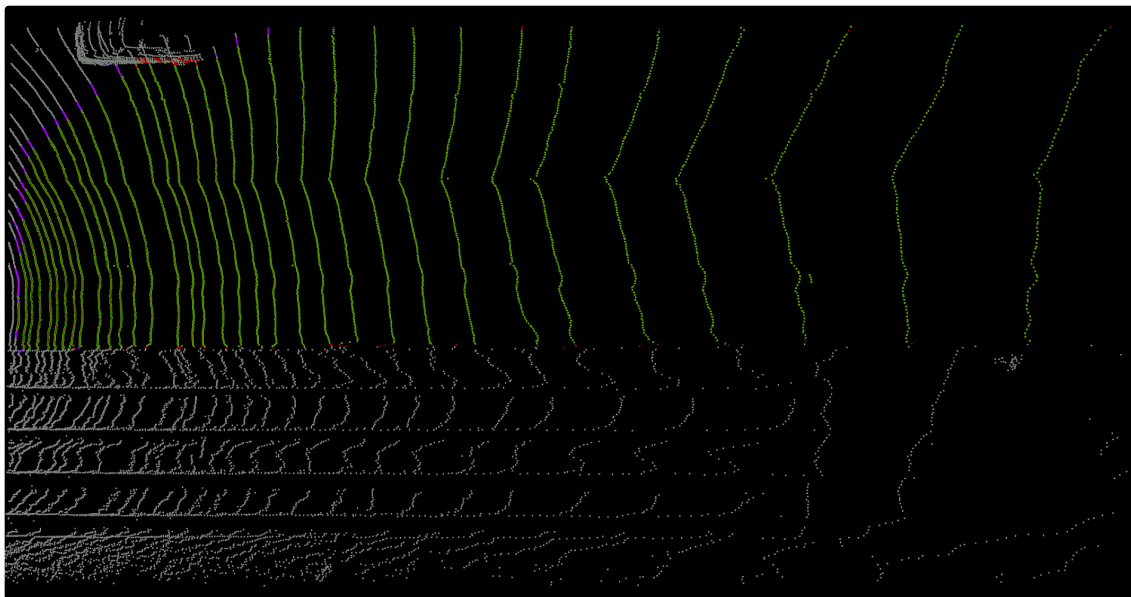
Najlepší výsledok z testovacej časti datasetu je znázornený na obrázku 7.6. Jeho projekcia do pôvodného mračna bodov (segmentácia laserového skenu) je vidno na obrázku 7.7. Na obrázku 7.6 v pravej časti značí žltá farba priestor mimo vozovky a fialová farba priestor, kde sa nachádza vozovka. V ľavej časti obrázku 7.6 značí zelená farba priestor, ktorý má byť správne klasifikovaný ako vozovka, no neurónová sieť toto neurčila správne. Naopak žltá farba v ľavej časti obrázku 7.6 značí priestor, ktorý bol klasifikovaný nesprávne ako časť vozovky. Nakoľko sa v týchto priestoroch nachádzala trieda pre miesta, kde nie je možné určiť príslušnosť priestoru k vozovke, je možné považovať túto žltú farbu v ľavom rohu za chybu datasetu.



Segmentovaná vozovka
pomocou trénovanej neurónovej siete

Obr. 7.6: Zobrazený najlepší výsledok segmentácie vozovky pomocou natrénovanej neurónovej siete, fialová farba označuje priestor ktorý bol nesprávne označený ako vozovka, modrá farba označuje priestor, ktorý bol označený nesprávne ako priestor mimo vozovky, zelená farba označuje správne určený priestor, kde sa nachádza vozovka a žltá farba označuje správne určený priestor, kde sa nenachádza vozovka

Na obrázku 7.7 je možné vidieť samotnú segmentáciu laserového skenu, ktorá bola prevedená projekciou výsledku uvedeného na obrázku 7.6 do laserového skenu, z ktorého bol vytvorený vstup pre neurónovú sieť.



Obr. 7.7: Znáznorená segmentácia laserového skenu pre najlepšiu vzorku, zelená farba značí správne predikovanú vozovku, fialová farba značí nesprávne predikovanú vozovku, červená farba značí nesprávne predikovaný priestor mimo vozovky a šedá farba značí správne predikovaný priestor mimo vozovky

Zodpovedajúca kamerová snímka sa nachádza na obrázku 7.8.



Obr. 7.8: Snímka z kamery patriaca k najlepšie vyhodnotenému laserovému skenu

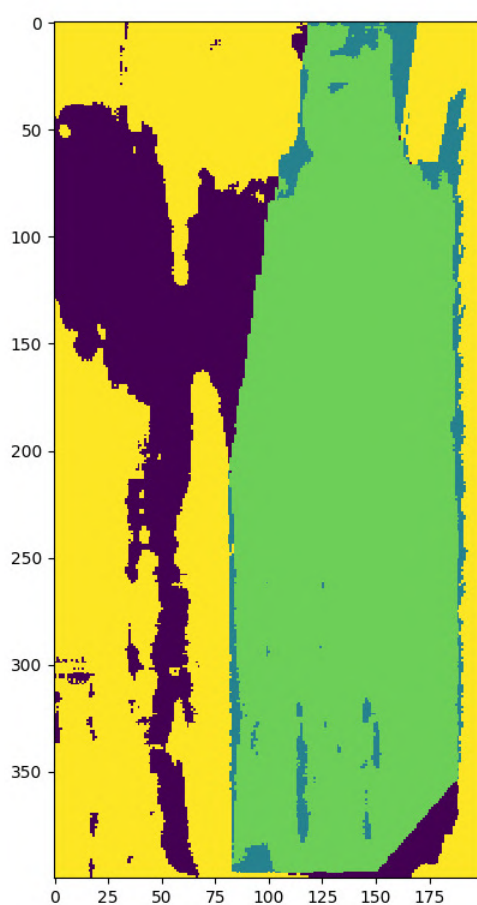
Dosiahnutý výsledok je možné pripísať jednoduchosti danej snímky a častému výskytu elementov vzorky v datasete. Týmito elementami sú napríklad rovná vozovka, železničná alebo električná trať.

Najhoršie segmentované mračno bodov

Najhorší výsledok z testovacej časti datasetu je znázornený na obrázku 7.10. Jeho projekcia do pôvodného mračna bodov (segmentácia laserového skenu) je vidno na obrázku 7.11. Farebné označenia sú rovnaké ako pri najlepšom výsledku.

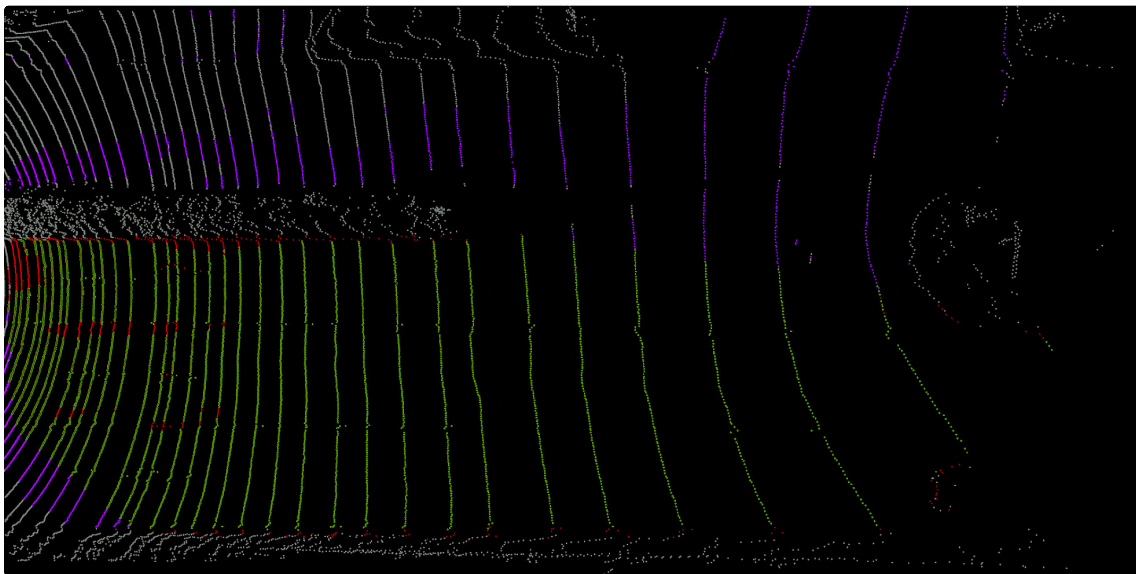


Obr. 7.9: Snímka z kamery patriaca k najhoršie vyhodnotenému laserovému skenu



Segmentovaná vozovka
pomocou trénovanej neurónovej siete

Obr. 7.10: Zobrazený je najhorší výsledok segmentácie vozovky testovacej časti datasetu vyhodnotený pomocou natrénovanej neurónovej siete, fialová farba označuje priestor ktorý bol nesprávne označený ako vozovka, modrá farba označuje priestor, ktorý bol označený nesprávne ako priestor mimo vozovky, zelená farba označuje správne určený priestor, kde sa nachádza vozovka a žltá farba označuje správne určený priestor, kde sa nenachádza vozovka



Obr. 7.11: Znázornená segmentácia laserového skenu pre najhoršiu vzorku, zelená farba značí správne predikovanú vozovku, fialová farba značí nesprávne predikovanú vozovku, červená farba značí nesprávne predikovaný priestor mimo vozovky a šedá farba značí správne predikovaný priestor mimo vozovky

Ako je vidno na obrázku 7.9 ide o komplikovanú križovatku. Segmentáciu laserového skenu nepriaznivo ovplyvňuje nákladné vozidlo zakrývajúce veľkú časť snímaného prostredia. Z pozorovaní výsledkov viacerých snímkov bolo zistené, že trávnaté plochy v strede vozovky majú nepriaznivý vplyv na výslednú segmentáciu, ako je tomu aj v tomto prípade. Na obrázku 7.10 sa nachádza zobrazený výstup neurónovej siete a jemu pridružený ground truth obrázok. Výsledok segmentácie vozovky je možné vidieť na obrázku 7.11, farebné označenia sú opäť rovnaké ako pri najlepšom výsledku. Taktiež je očividné, že oblasť, ktorá bola anotovaná ako priestor mimo vozovku, je z časti vozovka v protismere. Tento problém sa v datasete nachádza viackrát.

7.8 Zhodnotenie výsledkov dosiahnutých experimentmi

Pomocou ladenia viacerých parametrov bol dosiahnutý veľmi dobrý výsledok úspešnosti neurónovej siete. Najlepšie dosiahnuté hodnoty sú 96.12% pre úspešnosť neurónovej siete a 95.02% hodnota F-measure. Táto najlepšia neurónová sieť používa parameter `reduction` nastavený na hodnotu `mean` a parameter `weight` nastavený na hodnotu `(1,1,1,0)`, čo znamená, že výsledok chybovej funkcie nie je ovplyvnený miestami vo vzorkách, kde sa nachádzajú rotácie. Za druhý najlepší výsledok tréningu neurónovej siete je považovaná neurónová sieť s hodnotou presnosti 96.22% a hodnotou 94.95% pre parameter F-measure. Táto neurónová sieť používa rovnaké parametre ako predchádzajúca. Z porovnania najlepšieho a najhoršieho segmentovaného mračna je možné posúdiť, že natrénovaná neurónová sieť je veľmi presná pre rovné cesty alebo jednoduché prostredia, no jej presnosť je menšia pri segmentácii laserových skenov nasnímaných v blízkosti komplikovaných križovatiek s menej často vyskytujúcim sa tvarom.

Z tabuľky 7.4 vyplýva priemerné zlepšenie F-measure hodnoty o 2.17%. Rozdiel medzi najlepšou a najhoršou neurónovou sieťou je 3.69%.

	Reduction	weight	Prvý beh	Druhý beh	Tretí beh	Priemer
4 Triedy	mean	(1,1,1,1)	93.66%	94.18%	93.12%	93.65%
		(1,1,0,0)	93.95%	93.96%	93.73%	93.88%
		(1,1,1,0)	95.02%	94.95%	94.21%	94.73%
3 Triedy	mean	(1,1,1)	92.58%	93.23	93.36	93.06
	mean	(1,1,0)	94.26%	94.25%	94.48%	94.33%
	sum	(1,1,1)	93.66	92.70	91.33	92.56
	sum	(1,1,0)	94.19	94.07	94.40	94.22

Tabuľka 7.5: Tabuľka obsahujúca výsledky každej natrénovanej neurónovej siete, výsledok finálnej neurónovej siete je zvýraznený červenou farbou

Kapitola 8

Záver

Cieľom tejto bakalárskej práce bola segmentácia objektov v laserových skenoch pomocou konvolučnej neurónovej siete. Zvoleným objektom pre segmentáciu bola vozovka, nakoľko sa jedná o jeden z najdôležitejších prvkov v skúmaných laserových skenoch. Na tento účel bola vybraná a implementovaná jedna z existujúcich metód. Ground truth obrázky pre základný dataset (bez rotácií a preklopení) bol získaný od jedného z tvorcov metódy. Informácie z daných obrázkov boli premietnuté do mračna bodov a tie boli následne transformované. Z týchto transformovaných mračien boli vytvorené dvojice laserových skenov a im prislúchajúcich ground truth obrázkov. Použitý bol dataset o celkovej veľkosti 12180 vzoriek. Táto veľkosť bola dosiahnutá pomocou rotácií a preklopení cez os y . Po vytvorení datasetu sa podarilo úspešne implementovať neurónovú sieť a nájsť vhodné parametre pre chybovú funkciu, ako aj vhodný počet použitých tried. Výsledná neurónová sieť dokáže s úspešnosťou 96.12% previesť segmentáciu vozovky v laserovom skene (F-measure hodnota 95.02%).

Neurónová sieť poskytuje najhoršie výsledky v oblastiach s komplikovanými križovatkami menej často vyskytujúcich sa tvarov, naopak jednoduché križovatky zvláda vyhodnotiť veľmi dobre. Zaujímavým zistením bolo, že pre neurónovú sieť nepredstavuje problém relatívne hustá premávka. Celkovo si neurónová sieť pri segmentácii vozovky počína veľmi dobre, až na zopár výnimočných alebo zvláštnych prípadov. Zlepšenie úspešnosti neurónovej siete je možné dosiahnuť väčším a presnejším datasetom, hlavne pridaním menej často vyskytujúcich sa typov križovatiek. Presnosti by mohlo pomôcť použitie datasetu, ktorý by obsahoval iba dve triedy - priestor patriaci vozovke a priestor nepatriaci vozovke.

Literatúra

- [1] 3D, S. *Time-of-Flight vs. Phase-Based Laser Scanners: Right Tool for the Job* [online]. <https://spar3d.com/>, máj 2004 [cit. 2020-07-08]. Dostupné z: <https://www.spar3d.com/news/related-new-technologies/time-of-flight-vs-phase-based-laser-scanners-right-tool-for-the-job/>.
- [2] ALVAREZ, J. M., LOPEZ, A. a BALDRICH, R. Illuminant-invariant model-based road segmentation. In: *2008 IEEE Intelligent Vehicles Symposium*. 2008, s. 1175–1180.
- [3] ANTHONY, M. a BARTLETT, P. L. *Neural network learning: theoretical foundations*. Cambridge: Cambridge University Press, 2010. OCLC: 704340213. ISBN 9780511624216 9780521573535 9780521118620.
- [4] ARNX, A. *First neural network for beginners explained (with code)* [online]. <https://towardsdatascience.com>, január 2019 [cit. 2020-01-22]. Dostupné z: <https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf>.
- [5] BROWNLEE, J. *A Gentle Introduction to Dropout for Regularizing Deep Neural Networks* [online]. <https://machinelearningmastery.com/>, august 2019 [cit. 2020-01-23]. Dostupné z: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>.
- [6] CALTAGIRONE, L., BELLONE, M., SVENSSON, L. a WAHDE, M. LIDAR–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*. Sep 2018, roč. 111, s. 125–131.
- [7] CALTAGIRONE, L., SCHEIDEGGER, S., SVENSSON, L. a WAHDE, M. Fast LIDAR-based Road Detection Using Fully Convolutional Neural Networks. *CoRR*. 2017, abs/1703.03613.
- [8] CHEN, X., MA, H., WAN, J., LI, B. a XIA, T. Multi-view 3D Object Detection Network for Autonomous Driving. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. Jul 2017. Dostupné z: <http://dx.doi.org/10.1109/CVPR.2017.691>.
- [9] CHEN, Z., ZHANG, J. a TAO, D. Progressive LiDAR adaptation for road detection. *IEEE/CAA Journal of Automatica Sinica*. 2019, roč. 6, č. 3, s. 693–702.
- [10] DESHPANDE, A. *A Beginner's Guide To Understanding Convolutional Neural Networks* [online]. <https://adeshpande3.github.io>, jún 2016 [cit. 2020-01-23]. Dostupné z: <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>.

- [11] DESHPANDE, A. *A Beginner's Guide To Understanding Convolutional Neural Networks Part 2* [online]. <https://adeshpande3.github.io>, jún 2016 [cit. 2020-02-20]. Dostupné z: <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/>.
- [12] DONOGHUE, D., WATT, P., COX, N. a WILSON, J. Remote sensing of species mixtures in conifer plantations using LiDAR height and intensity data. *Remote Sensing of Environment*. október 2007, roč. 110, č. 4, s. 509–522. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0034425707001812>. ISSN 00344257.
- [13] FRITSCH, J., KUEHNL, T. a GEIGER, A. A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. In: *International Conference on Intelligent Transportation Systems (ITSC)*. 2013.
- [14] KARAGIANNAKOS, S. *Localization and Object Detection with Deep Learning* [online]. 2019 [cit. 2020-07-01]. Dostupné z: https://theaisummer.com/Localization_and_Object_Detection/.
- [15] KARN, U. *An Intuitive Explanation of Convolutional Neural Networks* [online]. <https://ujjwalkarn.me>, august 2016 [cit. 2020-01-23]. Dostupné z: <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>.
- [16] KOGAN, G. *How neural networks are trained* [online]. <https://ml4a.github.io>, december 2019 [cit. 2020-01-22]. Dostupné z: https://ml4a.github.io/ml4a/how_neural_networks_are_trained/.
- [17] KOZONEK, N., ZELLER, N., BOCK, H. a PFEIFLE, M. ON THE FUSION OF CAMERA AND LIDAR FOR 3D OBJECT DETECTION AND CLASSIFICATION. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2019, XLII-2/W16, s. 149–156. Dostupné z: <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W16/149/2019/>.
- [18] LEI, X., PAN, H. a HUANG, X. A Dilated CNN Model for Image Classification. *IEEE Access*. 2019, roč. 7, s. 124087–124095.
- [19] LIPTON, Z. C., ELKAN, C. C. a NARYANASWAMY, B. C. Optimal Thresholding of Classifiers to Maximize F1 Measure. *Machine Learning and Knowledge Discovery in Databases Lecture Notes in Computer Science*. May 2014, s. 225–239.
- [20] MAJTÁN, B. M. *Trénovatelná segmentace obrazu s použitím hlubokých neuronových sítí*. Brno, CZ, 2016. Bakalářská práce. Vysoké učení technické v Brně, FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=126382.
- [21] MARDIROSIAN, R. *Lidar vs. Camera: Driving in the Rain* [online]. [cit. 2019-07-10]. Dostupné z: <https://ouster.com/blog/lidar-vs-camera-comparison-in-the-rain/>.
- [22] MERCHANT, A. *Neural Networks Explained* [online]. <https://medium.com/>, december 2018 [cit. 2020-01-22]. Dostupné z: <https://medium.com/datadriveninvestor/neural-networks-explained-6e21c70d7818>.

- [23] NEWS, B. #004 CNN Padding / Master Data Science [online]. <https://bitprime.co>, január 2020 [cit. 2020-01-24]. Dostupné z: <https://bitprime.co/004-cnn-padding-master-data-science/>.
- [24] NOAA. *What is LIDAR* [online]. <https://web.archive.org/>, január 2013 [cit. 2020-01-20]. Dostupné z: <https://web.archive.org/web/20130530144617/http://oceanservice.noaa.gov/facts/lidar.html>.
- [25] OKUNSKY, M. V. a NESTEROVA, N. V. *Velodyne LIDAR method for sensor data decoding* [online]. Materials Science and Engineering516: <https://iopscience.iop.org>, január 2018 [cit. 2020-01-20]. Dostupné z: <https://iopscience.iop.org/article/10.1088/1757-899X/516/1/012018/pdf>.
- [26] OMID E. DAVID, N. S. N. *Convolutional Autoencoders* [online]. Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel, Center for Automation Research, University of Maryland, College Park, MD, USA: <https://www.researchgate.net/>, september 2016 [cit. 2020-01-22]. Dostupné z: https://www.researchgate.net/publication/306081538_DeepPainter_Painter_Classification_Using_Deep_Convolutional_Autoencoders.
- [27] ONDREJ, K. *Detekce a lokalizace dopravních značek*. Brno, CZ, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.fit.vut.cz/study/thesis/18882/>.
- [28] RASMUSSEN, C. Grouping dominant orientations for ill-structured road following. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. 2004, s. I–I.
- [29] REN, Y. *A Step-by-Step Implementation of Gradient Descent and Backpropagation* [online]. <https://towardsdatascience.com>, máj 2019 [cit. 2020-01-24]. Dostupné z: <https://towardsdatascience.com/a-step-by-step-implementation-of-gradient-descent-and-backpropagation-d58bda486110>.
- [30] ROSEBROCK, A. *Keras Conv2D and Convolutional Layers* [online]. <https://www.pyimagesearch.com/>, december 2018 [cit. 2020-01-22]. Dostupné z: <https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/>.
- [31] SARWADNYA, R. R. a MUKHEDKAR, M. *LiDAR Mapping: A Remote Sensing Technology* [online]. Electronics and Telecommunication, S. P. Pune University, Pune, Maharashtra, India: <https://www.ijsr.net/>, december 2014 [cit. 2020-01-20]. Dostupné z: <https://www.ijsr.net/archive/v3i12/MDExMjEOMDM=.pdf>.
- [32] SERENGIL, S. I. *ReLU as Neural Networks Activation Function* [online]. <https://sefiks.com>, august 2017 [cit. 2020-01-22]. Dostupné z: <https://sefiks.com/2017/08/21/relu-as-neural-networks-activation-function/>.
- [33] SERENGIL, S. I. *ELU as a Neural Networks Activation Function* [online]. <https://sefiks.com>, január 2018 [cit. 2020-01-22]. Dostupné z: <https://sefiks.com/2018/01/02/elu-as-a-neural-networks-activation-function/>.

- [34] SHANE, J. *Neural networks, explained* [online]. <https://physicsworld.com/>, jún 2018 [cit. 2020-01-22]. Dostupné z: <https://physicsworld.com/a/neural-networks-explained/>.
- [35] SHEEHAN, S. a SONG, Y. Deep Learning for Population Genetic Inference. *PLOS Computational Biology*. Marec 2016, roč. 12, s. e1004845.
- [36] SIDDHARTH SHARMA, S. S. *ACTIVATION FUNCTIONS IN NEURAL NETWORKS* [online]. Department of Computer Science and Engineering, Global Institute of Technology, Jaipur: International Journal of Engineering Applied Sciences and Technology, apríl 2020 [cit. 2020-07-1]. Dostupné z: <https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf>.
- [37] SILIPO, R. *Confusion Matrix and Class Statistics* [online]. <https://towardsdatascience.com/>, september 2019 [cit. 2020-06-22]. Dostupné z: <https://towardsdatascience.com/confusion-matrix-and-class-statistics-68b79f4f510b>.
- [38] SINGH, G. *The Problem with Softmax Activation Function* [online]. <https://www.medium.com>, október 2019 [cit. 2020-06-13]. Dostupné z: <https://medium.com/towards-artificial-intelligence/the-problem-with-softmax-activation-function-62f01353906f>.
- [39] SLAWOMIR PIATEK, P. *ToF & FMCW automotive LiDARs: design challenges and solutions* [online]. <https://hub.hamamatsu.com/>, december 2018 [cit. 2020-07-08]. Dostupné z: <https://hub.hamamatsu.com/us/en/webinar/tof-&-fmcw-automotive-lidars/index.html>.
- [40] TSANG, S.-H. *Review: Tompson CVPR'15 — Spatial Dropout (Human Pose Estimation)* [online]. <https://towardsdatascience.com>, máj 2019 [cit. 2020-01-23]. Dostupné z: <https://towardsdatascience.com/review-tompson-cvpr15-spatial-dropout-human-pose-estimation-c7d6a5cecd8c>.
- [41] WALIA, A. S. *Activation functions and it's types-Which is better?* [online]. <https://towardsdatascience.com>, máj 2017 [cit. 2020-01-22]. Dostupné z: <https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f>.

Príloha A

Obsah príloženého pamäťového média

Na priloženom pamäťovom médiu sa nachádzajú tieto súbory:

- /src/- zložka obsahujúca zdrojové súbory
 - /binFiles/ - Zložka s binárnymi verziami mračien bodov
 - /pclResultView/ - Cmake program pre zobrazovanie výsledkov
 - /pclRotator/ - Cmake program pre rotáciu mračna bodov
- /doc/ - zložka obsahujúca zdrojové súbory textovej časti práce
- /xzelen24_Bakalarska_Praca.pdf - výsledný text bakalárskej práce

Príloha B

Manuál

Pokyny pre použitie na operačnom systéme Linux:

1. Vstúpte do adresára `/src/`
2. Spustite skript `installPackages.sh`
3. Použite program `./useModel.py`, s argumentom názvu súboru s laserovým skenom