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

ROZPOZNÁNÍ VÝROBCE A MODELU AUTOMOBILU V OBRAZE

VEHICLE MAKE AND MODEL RECOGNITION IN IMAGE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **MAREK HRIVŇÁK**
AUTHOR

VEDOUCÍ PRÁCE **prof. Ing. ADAM HEROUT, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Hrivňák Marek**
Program: Informační technologie
Název: **Rozpoznání výrobce a modelu automobilu v obraze**
Vehicle Make and Model Recognition in Image
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte problematiku rozpoznání modelu automobilu v obraze.
2. Získejte datové sady pro učení a vyhodnocování studovaných metod.
3. Pořiďte vlastní dílčí datovou sadu pro rozšíření existujících dat; zaměřte se na některý vhodný aspekt.
4. Implementujte vybranou metodu (metody).
5. Experimentujte s vybranou metodou na vhodných datech, vylepšujte ji a charakterizujte její chování.
6. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- Bharath Ramsundar, Reza Bosagh Zadeh: TensorFlow for Deep Learning: From Linear Regression to Reinforcement Learning, O'Reily Media, 2018
- Richard Szeliski: Computer Vision: Algorithms and Applications, Springer, 2011
- Jakub Sochor, Adam Herout, Jiří Havel: BoxCars: Improving Fine-Grained Recognition of Vehicles Using 3-D Bounding Boxes in Traffic Surveillance, CVPR, 2016
- Yu Zhou ; Yao Yu ; Sidan Du: Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture, IEEE Tran. ITS, 2016
- Kun Huang, Bailing Zhang: Fine-grained vehicle recognition by deep Convolutional Neural Network, CISP-BMEI, 2016

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 až 5.

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

Vedoucí práce: **Herout Adam, prof. Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 14. května 2020
Datum schválení: 5. listopadu 2019

Abstrakt

Táto práca sa zaoberá trénovaním konvolučnej neurónovej siete na rozpoznávanie vozidiel z obrazu, príprave trénovacích dát a následne vytvoreniu metódy na vylepšenie rozpoznania. Riešenie sa zameriava na vplyv a využitie ohraničovacieho rámca a využitím dátovej augmentácie pre čo najvýššiu úspešnosť rozpoznania vozidiel z obrazu. Zároveň sa práca zameriava na porovnanie rozpoznania s využívaním obaľovacieho kvádra a poukázaniu na výraznejšie priblíženie, v niektorých prípadoch aj prekonanie, úspešnosti rozpoznania. V práci je použitá dátová sada BoxCars116k, ktorá je voľne dostupná a vytvorená skupinou GRAPH@FIT. Ako súčasť tejto práce som zároveň zozbieral obrázky vozidiel, ktoré môžu byť použité ako súčasť väčšej dátovej sady. Riešenie tejto práce zvyšuje úspešnosť rozpoznania vozidiel z obrazu až o 8 % v porovnaní s inými konvolučnými neurónovými sieťami bez aplikovanej metódy. Súčasťou práce sú aj vykonané experimenty, ktoré poukazujú na vplyv rôznych činiteľov na úspešnosť práce.

Abstract

This thesis focuses on training convolutional neural network for vehicle recognition in image, preparation of training data and improvement of classification accuracy. Solution focuses on effect of using 2D bounding box and data augmentation for better recognition accuracy. In this thesis, I also elaborate the comparison with papers using 3D bounding box and showing, my method approaches in some cases even outperforms method using 3D bounding box. BoxCars116k data set is used, which is freely available and collected by the GRAPH@FIT research group. In order to support the main data set, I also collected some vehicle images. As a result of the analysis, it is observed that accuracy of vehicle recognition increased 8% points in comparison with other convolutional neural networks without the proposed modifications. As part of my thesis I also performed several experiments, which show effect of different factors on classification accuracy.

Kľúčové slová

rozpoznanie vozidiel, konvolučná neurónová sieť, dátová sada, ohraničovací rámec, augmentácia dát, klasifikácia

Keywords

vehicle recognition, convolutional neural network, data set, 2D bounding box, data augmentation, classification

Citácia

HRIVŇÁK, Marek. *Rozpoznání výrobce a modelu automobilu v obraze*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

Rozpoznání výrobce a modelu automobilu v obraze

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána prof. Ing. Adama Herouta, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Marek Hrivňák
28. mája 2020

Poďakovanie

Rád by som poďakoval svojmu vedúcemu práce prof. Ing. Adamovi Heroutovi, Ph.D. za pomoc, odborné vedenie, cenné rady a veľkú trpezlivosť.

Obsah

Kapitola 1

Úvod

Téme rozpoznania objektov z obrazu sa s rozvojom počítačového videnia venuje každým rokom väčšia pozornosť a výnimkou nie je ani doprava. V porovnaní s inými objektmi sú vozidla špecifické svojím vzhľadom a rôznorodosťou. Kombinácie značiek, modelov a ročníkov všetkých vozidiel poskytuje naozaj širokú škálu objektov, ktoré je potrebné medzi sebou odlíšiť. K tomu prispieva aj fakt, že sa počet vozidiel medziročne stále zvyšuje, a tak je automatizované rozpoznanie a analýza vozidiel potrebná hneď vo viacerých oblastiach.

Rozpoznanie vozidiel z obrazu si v posledných rokoch získalo veľkú priazeň, čoho výsledkom je množstvo publikovaných vedeckých prác zaoberajúcich sa práve tejto téme. K rozpoznaniu vozidiel z obrazu je hneď niekoľko prístupov. Niektoré z nich sú obmedzené využívaním 3D modelov, iné zase rozdeľujú vozidlo na ďalšie časti, čo si vyžaduje naviac ešte detektor týchto častí. Jeden zo spôsobov je aj vytvorenie obaľovacieho kvádra (3D bounding box) okolo auta a následne pomocou neho vozidlo rozbalia do roviny. Všetky tieto prístupy k rozpoznaniu dosahujú skvelé výsledky, avšak potrebujú na lepšie rozpoznanie vytvoriť niečo (napr. 3D model, bounding box), čo celý proces rozpoznania spomaľuje a sťažuje, a preto je ich využitie v aplikáciách v reálnom čase problematické.

Mojím cieľom v tejto práci bolo v prvom rade zistiť aký vplyv má využitie ohraničovacieho rámca (2D bounding box) na presnosť rozpoznania vozidiel z obrazu a ďalej poskytnúť metódu, ktorá by k rozpoznaniu vozidiel pristupovala bez využitia či už 3D modelu alebo 3D bounding boxu, ktoré proces rozpoznania spomaľujú. Podstatou tejto práce je využitie neurónovej siete, ktorá sa za pomoci ohraničovacieho rámca (2D bounding box) a dátovej augmentácie snaží priblížiť k čo najlepším výsledkom. Tieto výsledky sú následne porovnávané s modelmi využívajúcimi 3D bounding box a poukazujú na priblíženie úspešnosti a pri niektorých architektúrach konvolučných neurónových sietí až k prekonaniu úspešnosti, a to aj bez komplikovanej tvorby 3D bounding boxu. Zároveň som v tejto práci vytvoril aj menšiu anotovanú dátovú sadu, ktorá môže byť neskôr pripojená k už existujúcim.

Práca je rozdelená do niekoľkých kapitol. Kapitola 2 poskytuje prehľad tradičných prístupov rozpoznania vozidiel. Kapitola 3 sa venuje základnému princípu fungovania konvolučných neurónových sietí a ich architektúre. Ďalej je popísaných hneď niekoľko najznámejších a najvýkonnejších architektúr, ktoré sa pri rozpoznávaní obrazu veľmi často využívajú. Kapitola 4 popisuje aktuálne dátové sady vozidiel ako aj sadu, ktorú som pri tejto práci využíval - jej obsah, problémy aj výhody. Ďalšia kapitola 5 popisuje akým spôsobom sú dáta pripravované pred samotným trénovaním, ako prebieha ich augmentácia aj popis samotného trénovania a ohodnotenia modelu. Nakoniec sú v kapitole 6 popísané dosiahnuté výsledky a experimenty.

Kapitola 2

Tradičné prístupy rozpoznania vozidiel v obraze

Rozpoznávanie objektov v obraze (Image recognition)[2] je súčasťou počítačového videnia (Computer Vision), je to schopnosť počítača rozoznávať objekty z obrazu, ako sú: ľudia, zvieratá, stromy, autá a podobne. Jedná sa o spojenie obrazu z kamery a umelej inteligencie, ktoré následne rozpozná objekt z obrazu. Ľudský mozog si vytvára obraz jednoducho, s ľahkosťou dokáže rozoznať mačku od psa, človeka od stromu, či auto od vtáka. Tento proces je však pre počítač omnoho náročnejší a nedokáže jednoducho určiť, o aký objekt v obraze sa jedná.

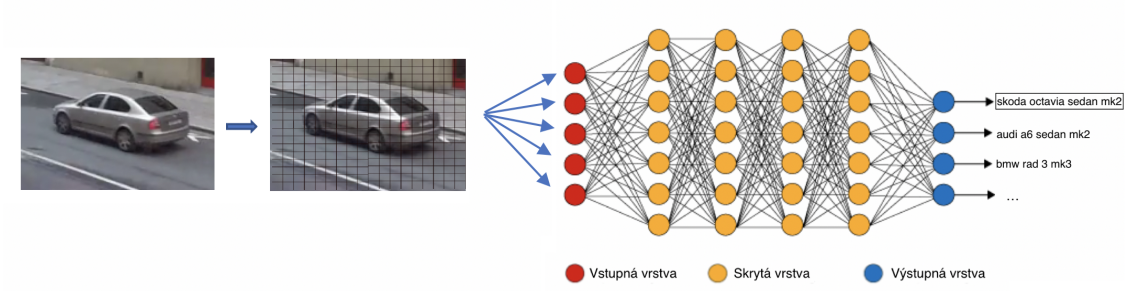
2.1 Základný princíp rozpoznávania vozidla v obraze

Pri rozpoznávaní obrazu sa výskumy sústreďujú zväčša na rozpoznávanie vtákov [6], [28], rastlín [23], či psov [10]. V súčasnej dobe sa do popredia dostáva aj rozpoznávanie vozidiel [14], [32]. Rozpoznávanie vozidiel sa využíva pri sledovaní, a to z rôznych dôvodov ako: riadenie mestskej dopravy alebo hľadanie podozrivého pri policajnom vyšetrovaní. Rozpoznávanie vozidiel tradične prebieha len na úrovni typu vozidla, čiže rozlišuje len typ ako dodávka, kamión, osobné auto, a podobne. Toto rozpoznávanie však nie je vždy dostatočné, hlavne ak sa jedná o vyhľadanie konkrétneho auta ako pri spomínanom policajnom vyšetrovaní. Z tohto dôvodu sa v tejto práci venujem rozpoznávaniu vozidiel podľa značky, modelu a generácie.

Rozpoznávanie je vykonávané hlbokým učením[1] (deep learning) s využitím konvolučných neurónových sieti (CNN) bližšie popísaných v kapitole 3. Algoritmy pre rozpoznávanie objektov v obraze sú väčšinou trénované na veľkom množstve fotiek dostupných v dátovej sade 4. Rozpoznávanie prebieha v 4 hlavných krokoch:

1. Príprava označených fotiek
2. Extrakcia pixelov z obrazu
3. Natrénovanie modelu CNN
4. Rozpoznávanie objektu z obrazu

[1] disciplína v rámci strojového učenia, ktorá sa zaoberá využitím algoritmov (väčšinou neurónových sieti) s veľkým počtom vrstiev (layers) reprezentujúcich dáta.



Obr. 2.1: Znázornenie priebehu rozpoznávania.

2.2 Metódy rozpoznávania vozidiel v obraze

V rámci rozpoznania vozidiel v obraze sa výskumy sústreďujú na viacero metód. Týmito metódami sa snažia docieliť čo najlepšie výsledky. Niektoré tieto metódy sa zaoberajú rozpoznávaním s využitím 3D objektov, avšak preto sú obmedzené len na použitie 3D modelu, čím sa ich využitie trochu komplikuje. Iné zase využívajú ohraničovacie rámce (2D bounding box) alebo obaľovacie kvádre (3D bounding box) k presnejším výsledkom. Niektoré metódy, ktoré sa využívajú všeobecne na rozpoznávanie objektov v obraze sa používajú aj konkrétne pri rozpoznávaní vozidiel, takýmto to príkladom je metóda rozdelenia vozidla na jednotlivé časti. Táto podkapitola sa sústredí na opis týchto metód a vysvetlenie ich použitia.

2.2.1 Metóda s využitím 3D modelu

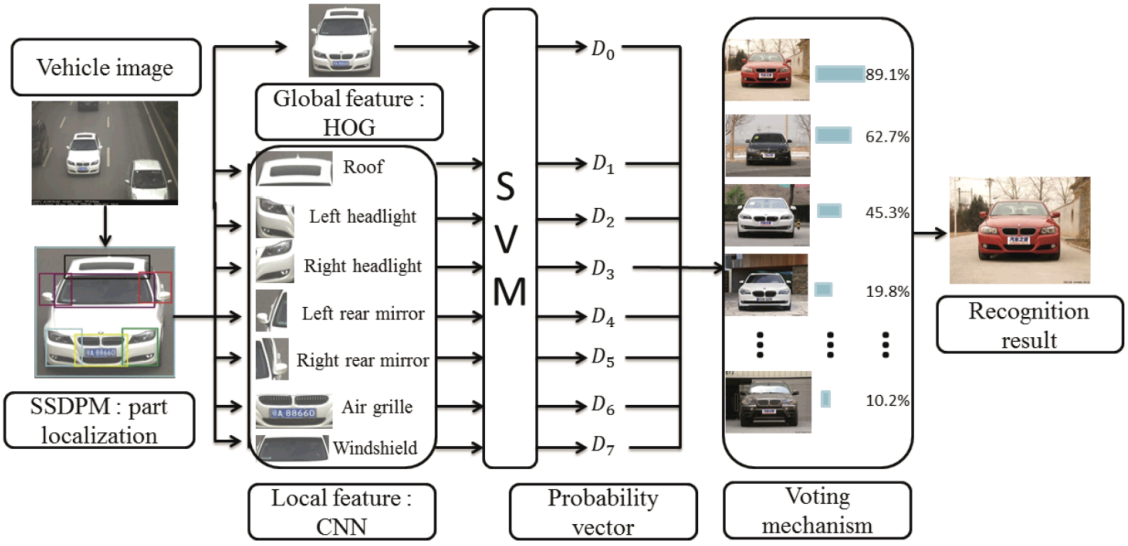
Vozidlá sú špecifické hlavne svojím 3D zobrazením, a preto sa táto metóda snaží na rozdiel od ostatných, ktoré sa sústredia len na 2D obrázky vytvoriť 3D model, ktorý poskytuje lepšie výsledky pri rozpoznávaní.

Yaming Wang et al. [30] sa pri svojom výskume zaoberali vytváraním 3D póz z 2D obrázkov a ich 3D modelov. Zároveň vytvorili 2 dátové sady, kde každá z nich obsahuje 2 časti: 2D obrázky a 3D modely. 2D obrázky vozidiel vyzbierali z voľne dostupných dátových sád a pomocou repozitára ShapeNet [1], ktorý obsahuje veľké množstvo 3D modelov, určených pre rozpoznávanie vozidiel, dostupných aj s názvom značky a modelu sa snažili priradiť 3D model k ich obrázku auta. V prípade, že žiadnu zhodu nenašli, manuálne vybrali vizuálne podobný model a priradili ho k obrázku auta.

Ďalší, využívajúci 3D model vozidiel pri svojom výskume boli Yen-Liang Lin et al. [13], tí sa snažili zlepšiť rozpoznávanie vozidiel použitím 3D CAD modelov. Tieto 3D CAD modely získavajú z 2D obrázkov, extrakciou polohy častí na základe modelov deformovateľných častí (DPM), a potom pomocou regresných modelov odhadujú umiestnenie orientačných bodov. Následne umiestnia orientačné body 3D modelu na predpovedané 2D orientačné body, extrahujú prvky a vložia tieto prvky do SVM klasifikátora. Zároveň pre túto štúdiu vytvorili novú dátovú sadu FG3DCar, ktorá obsahuje 300 obrázkov s 30 rôznymi triedami modelov áut pod rôznymi uhlami pohľadu. Ich prístup s využitím 3D-modelov porovnali s najnovšími prístupmi založenými len na 2D obrázkoch a preukázali jasné zlepšenie rozpoznávania vozidiel.

Podobným spôsobom sa rozpoznávaniu venovala aj štúdia od Krause J. et al. [11]. V tejto štúdii používali spojenie 2 reprezentácií 2D objektov a pomocou nich vytvorili 3D objekt. Následne v rozsiahlych experimentoch ukazujú, že využitie 3D objektov prekonáva presnosť rozpoznávania 2D objektov a zároveň demonštrujú ich účinnosť pri odhadovaní

3D geometrie z obrázkov. Ako súčasť ich práce predstavili novú dátovú sadu obsahujúcu 207 tried, ktorá sa dá rozdeliť na ďalšie dve: malú, ktorá obsahuje len 10 modelov značky BMW a veľkú, ktorá obsahuje rôznych 197 tried.



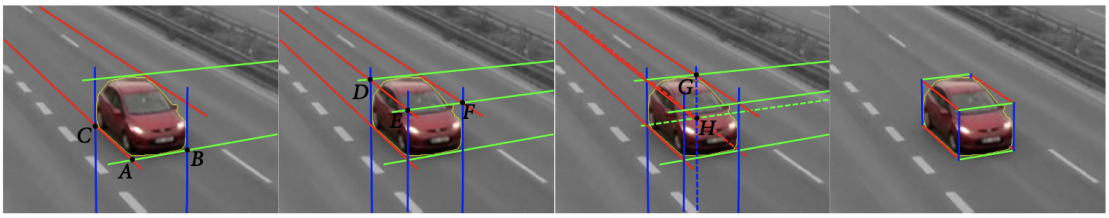
Obr. 2.2: Ukážka metódy rozdelenia vozidla na časti [29].

2.2.2 Metóda využívajúca rozdelenie vozidla na časti

Medzi často používané metódy pri rozpoznávaní obrazu patrí aj metóda rozdelenia objektu na viacero častí. Tieto metódy sa veľmi často používajú napríklad pri rozpoznávaní vtákov [31], ale tak isto aj pri rozpoznávaní vozidiel. Výhodou tejto metódy je, zameranie sa na konkrétne časti objektu, a tak dokáže rozoznať aj menšie rozdiely medzi objektami.

Touto metódou sa vo svojej štúdií zaoberali aj Qi Wang et al. [29]. Ako prvé museli lokalizovať dané časti vozidla pomocou DPM[2]. Medzi lokalizované časti vozidla patria: strecha, ľavé spätné zrkadlo, čelné sklo, pravé spätné zrkadlo, ľavé svetlá, mriežka na nasávanie vzduchu a pravé svetlá. Následne jednotlivé časti a globálny popis vozidla použili pri rozpoznaní. Z toho sa vytvoril pravdepodobnostný vektor a nakoniec pomocou hlasovacieho mechanizmu založeného na diskriminačnej schopnosti sa vybralo konkrétne vozidlo. Celý tento priebeh je zobrazený na obrázku 2.2. Popritom vytvorili aj novú dátovú sadu, ktorá obsahuje aj anotáciu jednotlivých lokalizovaných časti. Aby si zabezpečili hustú premávku vybrali si miesta situované v centre mesta, a tak boli schopní vyhotoviť čo najväčšie množstvo záberov. Zozbierali 4584 obrázkov vozidiel, 8 značiek a 50 modelov. Čiže každá značka má 5 až 9 modelov a z toho 40 obrázkov každého modelu. Každý rovnaký model vozidla z rôznych rokov (teda prešiel nejakými vizuálnymi zmenami) považujú za odlišný model. Následne na experimentoch vykonaných na vytvorenej dátovej sade ukázali, že priemerná presnosť dosahovala až 92,3 %, čo je o 3,4-7,1 % viac ako pri najmodernejších prístupoch rozpoznávania vozidiel z obrazu.

[2]Deformable Parts Model - prístup k lokalizácií častí objektu publikovaný Felzenszwalbom [4]



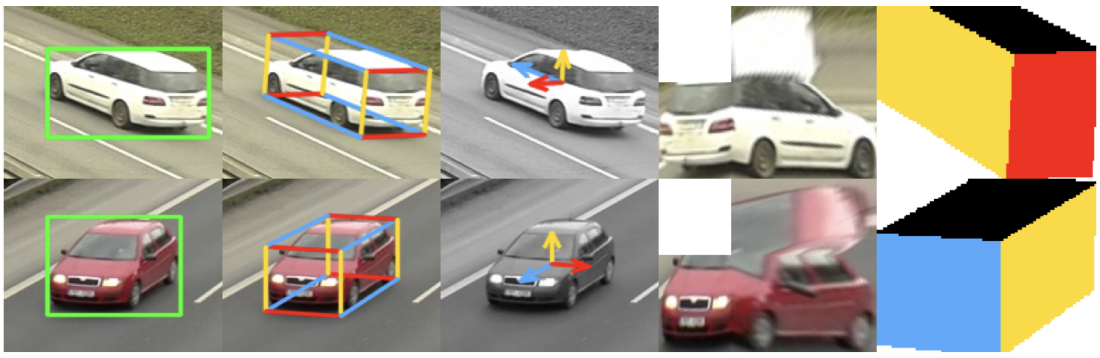
Obr. 2.3: Znázornenie vytvárania 3D obaľovacieho kvádra [20].

2.2.3 Metóda využívajúca 3-D Bounding Box

Jeden z možných prístupov k rozpoznávaniu vozidiel v obraze je aj metóda, ktorá využíva obaľovací kváder (3D bounding box). Táto metóda sa zameriava na vytvorenie 3D ohraničenia okolo vozidla, čo dopomáha k následne vyššej úspešnosti pri jeho rozpoznaní.

Touto metódou sa vo svojej práci zaoberali J. Sochor, J. Špaňhel a A. Herout [20]. Cieľom ich práce bolo zamerať sa na rozpoznávanie vozidiel z obrazu v doprave, pričom zábery vyhotovené z dopravy boli z cestných kamier a neboli obmedzené len na pohľad spredu alebo zozadu. Celu prácu založili na vytvorení 3D bounding boxu okolo auta. Tento 3D bounding box môže byť zhotovený automaticky z údajov o pozorovaní dopravy (tento proces je možné vidieť na obrázku 2.3). Zároveň kedy nie je možné vyhotoviť tento bounding box normálnym spôsobom, predložili metódu na odhad 3D bounding boxu z obrázku. Následne tieto kvádre pomáhajú pri lokalizácii bočnej, prednej alebo zadnej časti a strechy vozidla. Pomocou lokalizovaných segmentov vozidla sa následne rozbalí obrázok vozidla do roviny (obrázok 2.4). Toto rozbalenie obrázka do roviny je použité namiesto pôvodných obrázkov, ako vstup do konvolučnej neurónovej siete. Výhodou tohto rozbalenia je lokalizácia jednotlivých častí vozidla, normalizácia jeho pozície v obrázku, a to všetko bez využitia akéhokoľvek algoritmu na lokalizáciu časti vozidla.

Pre dosiahnutie väčšej rôznorodosti trénovacích dát vytvorili aj 2 vhodné techniky augmentácie. Tá prvá sa zaoberá faktom, že pre rozpoznávanie objektov z obrazu nie je farba veľmi dôležitá a tak každému pixelu z obrázka pridali rovnakú náhodnú hodnotu. Druhá zase náhodnú časť v obrázku vyplní šumom.



Obr. 2.4: Zobrazenie rozbalenia 3D obaľovacieho kvádra na reálnom príklade [20].

sieť	bez modifikácie	3D obaľovací kváder
AlexNet	66,65/77,75	77.67/88.28
VGG16	77,26/86,71	83.79/92.23
VGG19	76,74/86.06	83.91/92.17
ResNet50	75.48/84.61	82.27/90.79

Tabuľka 2.1: Tabuľka zobrazujúca úspešnosť rozpoznania bez modifikácie a s využitím 3D bounding boxu [20].

Taktiež bolo súčasťou ich práce vytvoriť vhodnú dátovú sadu pre rozpoznávanie, preto zozbierali vhodné zábery z kamier cestnej premávky a vytvorili dátovú sadu BoxCars116k o veľkosti 116286 obrázkov vozidiel. Táto dátová sada je bližšie popísaná v podkapitole 4.2.

Výsledkom ich práce bolo zvýšenie presnosti pri rozpoznávaní vozidiel, a to až o takmer 12 % a zníženie chybovosti pri rozpoznávaní o takmer 50 % v porovnaní so základnými konvolučnými neurónovými sieťami (tabuľka 2.1).

Za nevýhodu tejto práce považujem zložitosť vytvorenia obaľovacieho kvádra. Toto riešenie si vyžaduje dlhšiu dobu na výpočet, a tak je ťažšie použiteľné v aplikáciách v reálnom čase.

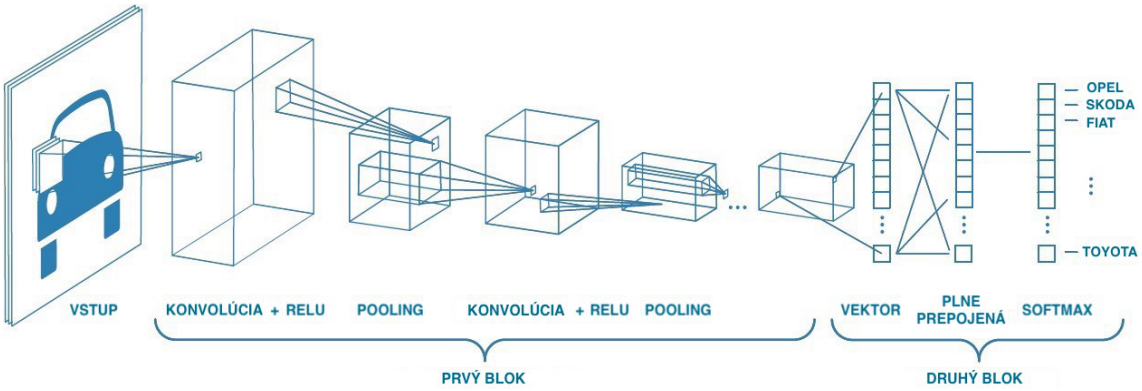
Kapitola 3

Konvolučné neurónové siete a ich použitie pri rozpoznaní v obraze

Konvolučná neurónová sieť (ConvNet/CNN) [17], [22] je špeciálna architektúra neurónovej siete, predstavená v roku 1988. Jedná sa o algoritmus hlbokého učenia, ktorý dokáže zo vstupného obrázku priradiť dôležitosť rôznym aspektom, a následne ich jeden od druhého odlíšiť. Architektúra CNN je analogická so štruktúrou konektivity neurónov v ľudskom mozgu. Jednotlivé neuróny reagujú na podnety iba v obmedzenej oblasti zorného poľa známeho ako receptívne pole. Súbor takýchto polí sa prekrýva tak, aby pokryl celú vizuálnu oblasť. Konvolučná neurónová sieť je schopná úspešne zachytiť priestorovú a časovú závislosť v obraze pomocou príslušných filtrov. Architektúra sa tak dokáže lepšie prispôsobiť obrázkom dátovej sady z dôvodu zníženia počtu zahrnutých parametrov a opätovného použitia váh. Inými slovami táto sieť môže byť vytrénovaná tak, aby lepšie pochopila sofistikovanosť obrázku.

Počítač na rozdiel od ľudského mozgu vníma obrázok ako maticu pixelov (obrázok 3.1). Aj z tohto dôvodu počítač k rozpoznaniu objektu z obrazu pristupuje odlišným spôsobom. Zatiaľ čo človek by sa snažil v obrázku s vozidlom nájsť nejaké charakteristické črty ako je veľkosť svetiel vozidla alebo dĺžka vozidla, počítač vníma charakteristické črty ako zakrivenosť, či ohraničenie. Následne pomocou skupín konvolučných vrstiev vytvorí počítač abstraktnejšie koncepty.

Obr. 3.1: Maticové vyjadrenie RGB obrázka ($4 \times 4 \times 3$), rozdeleného na trojfarebné vrstvy [17].



Obr. 3.2: Architektúra jednoduchej CNN. [17]

3.1 Architektúra konvolučných neurónových sietí

Architektúra konvolučných neurónových sietí [19] je tvorená podobne ako u tradičných neurónových sietí postupnosťou jednotlivých vrstiev. Každá vrstva transformuje vstupný obraz s diferencovateľnou funkciou, ktorá môže alebo nemusí mať parametre. Za výhodu CNN sa považuje menšie množstvo parametrov a jednoduchšie trénovanie oproti plne prepojenej sieti.

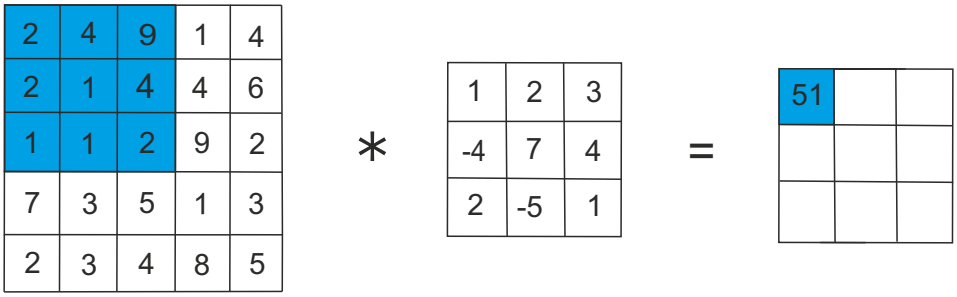
Architektúru CNN je možno rozdeliť do dvoch blokov:

1. **Prvý blok** - vytvára špecifickosť tohto typu neurónovej siete, pretože funguje ako extraktor danej vlastnosti (feature). Za týmto účelom porovnáva vzory využitím konvolučného filtrovania. Prvá vrstva filtruje obraz pomocou niekoľkých konvolučných jadier a vracia mapy vlastností (feature maps), ktoré sú potom normalizované alebo majú upravenú veľkosť. Tento proces sa môže opakovať niekoľkokrát: filtrujeme mapy vlastností získané pomocou nových jadier, ktoré nám poskytnú zase nové mapy na normalizáciu a úpravu veľkostí a zase opakujeme. Nakoniec sa hodnoty posledných máp zlúčia do vektora. Tento vektor definuje výstup prvého bloku a vstup druhého bloku.

2. **Druhý blok** - nie je charakteristický pre CNN. V skutočnosti je na konci všetkých neurónových sietí použitý na klasifikáciu. Hodnoty vstupných vektorov sa transformujú (pomocou niekoľkých lineárnych kombinácií a aktivačných funkcií), aby sa na výstup vrátil nový vektor. Tento posledný vektor obsahuje toľko prvkov, koľko máme tried: daný prvok predstavuje pravdepodobnosť, že obraz patrí do danej triedy. Každý prvok je preto medzi 0 a 1 a súčet všetkých má hodnotu 1. Tieto pravdepodobnosti sa vypočítavajú pomocou poslednej vrstvy tohto bloku (a teda siete), ktorá využíva logistickú funkciu (binárna klasifikácia) alebo aktivačnú (softmax) funkciu.

Architektúra CNN je tvorená 4 typmi vrstiev:

1. Konvolučná vrstva
2. Aktivačná vrstva
3. Pooling vrstva
4. Plne prepojená vrstva



Obr. 3.3: Zobrazenie konvolúcie, kde prvá matica predstavuje vstupný obraz, $*$ je znakom konvolúcie, druhá matica je konvolučný filter a posledná matica je výsledná matica po konvolúcií prvého kroku.

3.1.1 Konvolučná vrstva

Konvolučná vrstva je kľúčovou súčasťou konvolučných neurónových sietí a je vždy aspoň prvou vrstvou. Jej účelom je získať skupinu vlastností z obrazu, a to sa vykoná konvolučným filtrovaním.

Diskrétna konvolúcia je definovaná ako:

$$g(x,y) = f(x,y)h(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} f(x-i, y-j)h(i,j) \tag{3.1}$$

kde g je výsledný obraz, f je vstupný obraz a h je filter.

Princípom konvolúcie je, že na vstupnú maticu (obraz) sa do ľavého horného rohu priloží menšia matica filter (jadro). Úlohou filtra je vynásobenie jeho hodnôt hodnotami vstupnej matice, nad ktorými je priložený. Všetky násobky sa následne sčítajú a výsledkom je jedna hodnota, ktorá sa zapíše do výslednej matice. Tak sa filter posunie o krok (napr. jeden pixel doprava) a svoju úlohu opäť zopakuje. Po prejdení celého riadka sa zase posunie o riadok nižšie, a takto postupuje cez celú maticu. Tento proces zobrazuje obrázok 3.3.

Táto operácia je z ľudského hľadiska analogická s identifikáciou hrán a jednoduchých farieb na obrázku. Aby sa však rozpoznali vlastnosti vyššej úrovne, ako sú svetlá alebo kolesá vozidla, je potrebná celá sieť. Sieť bude pozostávať z niekoľkých konvolučných sietí zmiešaných s nelineárnymi a pooling vrstvami. Keď obraz prechádza jednou konvolučnou vrstvou, výstup prvej vrstvy je vstupom pre druhú vrstvu, a tak to pokračuje pre každú vrstvu.

Konvolučná neurónová sieť sa líši od ostatných spôsobom akým sú vrstvy naskladané, a taktiež parametrami. Konvolučná vrstva má 4 hyperparametre:

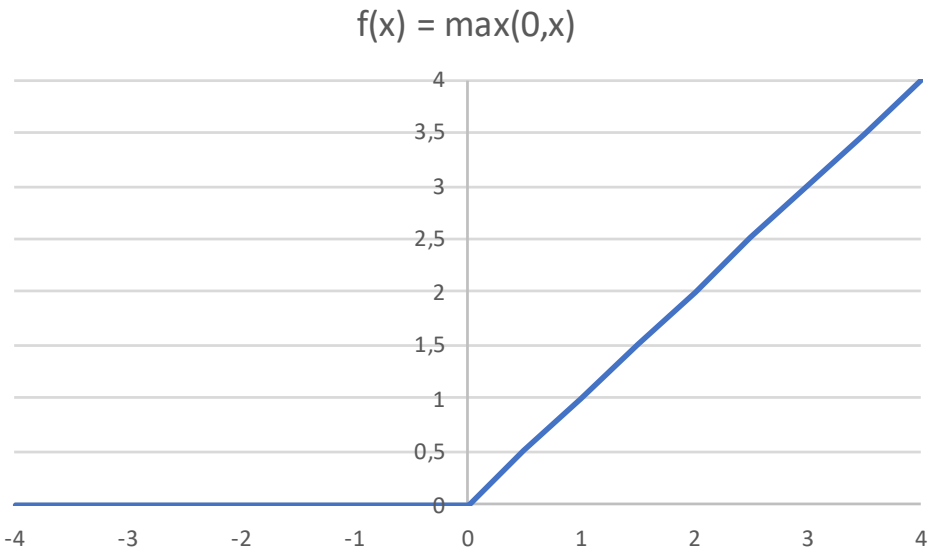
1. Počet filtrov K
2. Veľkosť filtra F – každý filter má rozmery $F \times F \times D$ pixelov.
3. Krok S, o ktorý sa posunie filter (napr. jeden pixel)
4. Zero-padding – nastane vtedy, keď okolo okrajov vstupnej matice pridáme pixely s hodnotou nula. To ma za následok, že výstupná matica bude rovnakej veľkosti ako vstupná.

3.1.2 Aktivačná vrstva

Aktivačná vrstva nasleduje bezprostredne za konvolučnou. Má aktivačnú funkciu, ktorej cieľom je zaviesť nelinearitu do CNN.

ReLU (Rectified Linear Unit) je aktivačná funkcia, ktorá nahrádza všetky záporné hodnoty prijaté ako vstupy nulami.

Namiesto ReLU sa môžu použiť aj iné nelineárne funkcie, ako je tanh alebo sigmoid. Avšak najčastejšie používaná je ReLU, pretože je jednoduchá, rýchla a funguje dobre. Trénovanie siete s ReLU má tendenciu konvergovať oveľa rýchlejšie a spoľahlivejšie ako pri iných aktivačných funkciách.



Obr. 3.4: Grafické znázornenie aktivačnej funkcie ReLU.

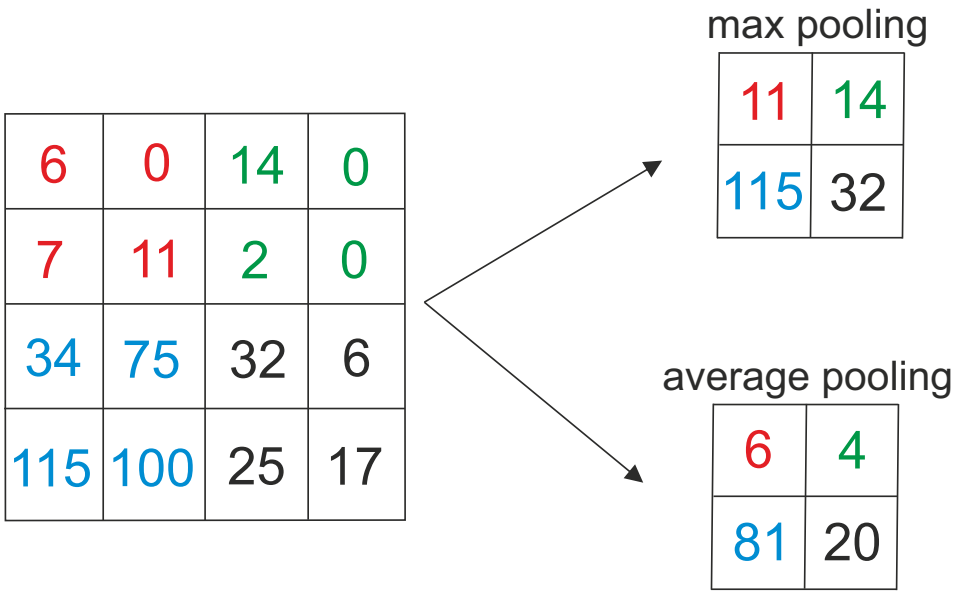
3.1.3 Pooling vrstva

Pooling vrstva sa nachádza často medzi konvolučnými, a jej úlohou je spájať dohromady podobné príznaky, čím dochádza k zmenšeniu priestorovej veľkosti siete. Zároveň znižuje počet parametrov a výpočtov v sieti, a tak zvyšuje efektívnosť siete a predchádza nadmernému učeniu (over-learning). Je užitočná aj pri extrahovaní dominantných vlastností, ktoré sú rotačne a pozične nemenné, a tým udržiavajú efektívnosť trénovania modelu. Zmysel poolingu je v tom, že ak už boli niektoré vlastnosti (napr. hrany) identifikované v predchádzajúcom procese konvolúcie, podrobný obrázok už nie je potrebný na ďalšie spracovanie a skomprimuje sa.

Poznáme 2 typy operácie pooling:

1. Max Pooling – maximálna hodnota prítomná vo vybranej oblasti sa zachová a všetky ostatné hodnoty sa zahodia. Zároveň funguje ako prostriedok na potlačenie šumu.

2. Average Pooling – zachováva priemernú hodnotu zo všetkých hodnôt vo vybranej oblasti.



Obr. 3.5: Grafická ukážka funkcie pooling. Na obrázku je vidieť, ako funkcia max pooling si vyberie vždy najväčšiu hodnotu, zatiaľ čo average pooling vždy priemernú hodnotu.

3.1.4 Plne prepojená vrstva

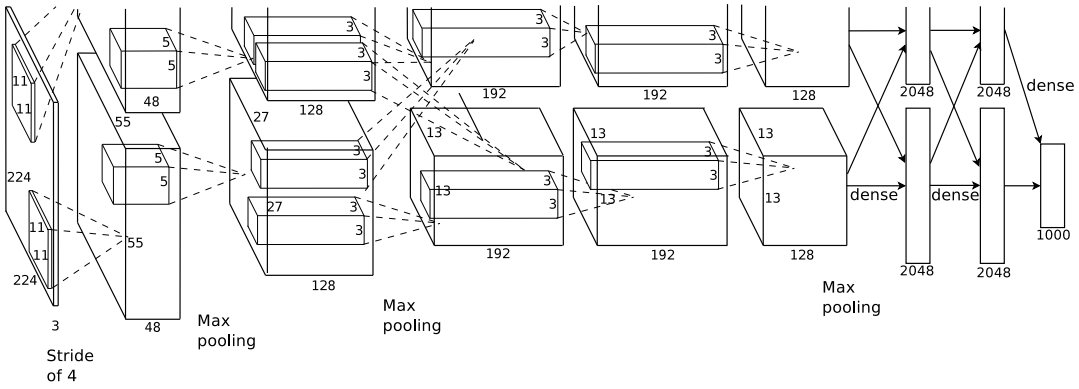
Plne prepojená vrstva je vždy posledná vrstva neurónovej siete. Táto vrstva obdrží na vstupe vektor a vracia vektor o veľkosti N, kde N je počet tried ktoré rozpoznávame z obrazu. Každý element vektora indikuje pravdepodobnosť, akou obrázok na vstupe CNN patrí danej triede. Na výpočet pravdepodobnosti plne prepojená vrstva vynásobí každý vstupný prvok váhou, vytvorí súčet, a potom použije aktivačnú funkciu (logistická ak $N = 2$, softmax, ak $N > 2$). Toto je ekvivalentné vynásobeniu vstupného vektora maticou obsahujúcou váhy. Skutočnosť, že každá vstupná hodnota je spojená so všetkými výstupnými hodnotami vysvetľuje pojem plne prepojená.

3.2 Používané modely pre rozpoznanie z obrazu

Pri rozpoznaní objektov z obrazu sa veľmi často využívajú rôzne už vytvorené architektúry konvolučných neurónových sieti. Tieto architektúry sa už roky vyvíjajú, zlepšujú, prispôsobujú, čo viedlo naozaj k výborným výsledkom v hlbokom učení. Dobrým meradlom týchto architektúr sú rôzne súťaže, založené na natrénovaní rovnakej dátovej sady (napr. ImageNet[1]) s čo najlepším výsledkom.

V tejto podkapitole sa budem venovať architektúram CNN, ktoré sú veľmi populárne pri rozpoznaní z obrazu, a zároveň boli použité v tejto práci. Ako prvá je popísana architektúra AlexNet, na ktorej základoch sa potom odvíjaju ďalšie. Na konci tejto podkapitoly je porovnanie týchto architektúr na dátovej sade ImageNet.

[1] http://www.image-net.org/



Obr. 3.6: Ilustrácia architektúry AlexNet, ktorá explicitne ukazuje vymedzenie zodpovedností medzi dvoma GPU. Jedna GPU spúšťa časti vrstvy v hornej časti obrázka, zatiaľ čo druhá spúšťa časti vrstvy v spodnej časti obrázka [12].

3.2.1 AlexNet

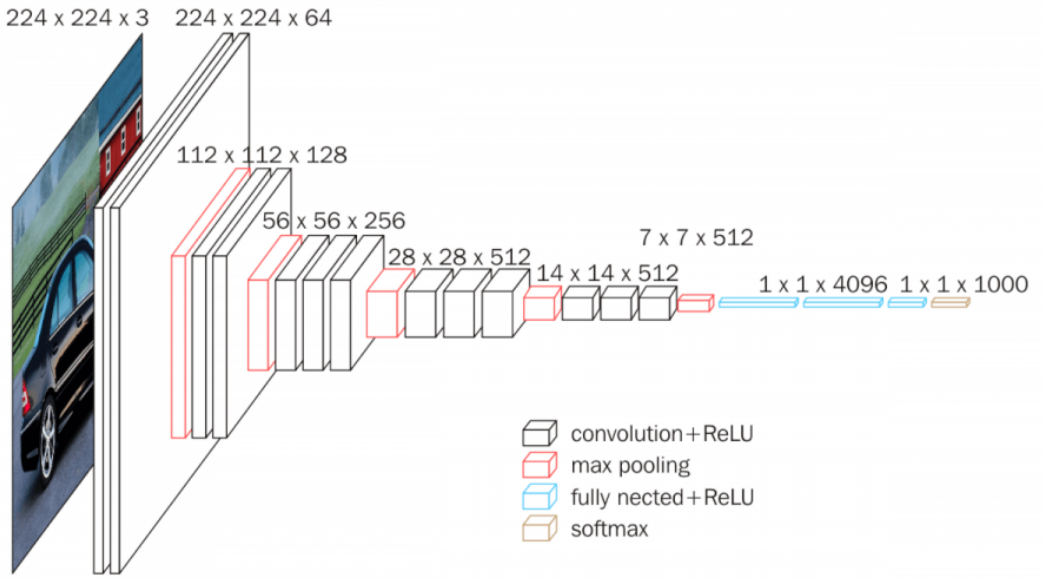
AlexNet je konvolučná neurónová sieť, ktorú publikovali vo svojej práci [12] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. Táto práca predstavuje obrovský posun vpred v počítačovom videní a prináša novú generáciu CNN pre rozpoznanie v obraze. AlexNet [16] významne znižuje chybovosť, keď z 5 najlepších predpovedí siete (top-5 error rate) sa chybovosť pohybuje na úrovni 15,3 %, zatiaľ čo druhá najlepšia sieť dosahuje chybovosť až 26,2 %. Je považovaná za jednu zo sietí, ktoré mali najväčší vplyv na vývoj počítačového videnia. V dobe písania tejto práce, prácu v ktorej bola sieť AlexNet predstavená bola citovaná už vyše 61000-krát.

AlexNet je omnoho väčšia ako predchádzajúce CNN použité v počítačovom videní. Má 60 miliónov parameterov a 650 000 neurónov. Skladá sa z 5 konvolučných vrstiev a 3 plne prepojených vrstiev.

Viacero konvolučných filtrov extrahuje zaujímavé vlastnosti z obrázka. V jednej konvolučnej vrstve je obvykle veľa filtrov rovnakej veľkosti. Napríklad prvá konvolučná vrstva obsahuje 96 filtrov veľkosti 11×3. Za prvými dvoma konvolučnými vrstvami nasleduje Max Pooling vrstva. Tretia, štvrtá a piata vrstva sú priamo prepojené. Za piatou opäť nasleduje Max Pooling vrstva, ktorej výstup ďalej pokračuje do 2 plne prepojených vrstiev, a z toho sa pomocou funkcie Softmax[2] určia pravdepodobnostné hodnoty priradené k triedam. Využíva nenasýtenú aktivačnú funkciu ReLU, ktorá preukázala zlepšenie výkonnosti pri trénovaní v porovnaní s funkciami tanh a sigmoid.

Výsledky v ich práci poukazujú, že veľká a hlboká konvolučná neurónová sieť je schopná dosiahnuť rekordné výsledky na veľmi náročnej dátovej sade s využitím výhradne kontrolovaného (riadeného) učenia (supervised learning). Z tejto štúdie jasne vyplýva, že odstránenie hoci jednej vrstvy ma výrazný vplyv na jej výkonnosť. Napríklad odstránenie ktorejkoľvek zo stredných vrstiev ma za následok stratu asi 2 % pri úspešnosti rozpoznania.

[2] Aktivačná funkcia, popísaná v kapitole 3.1.4.



Obr. 3.7: Ilustrácia architektúry VGG16 [15].

3.2.2 VGG

VGG16 a VGG19 [18] sú konvolučné neurónové siete predstavené skupinou *Visual Geometry Group* z Oxfordskej univerzity. V ich práci sa zaoberali vplyvom hĺbky CNN na presnosť pri rozpoznávaní z obrazu. Hlavným prínosom tejto práce je dôkladné vyhodnotenie sietí s rastúcou hĺbkou pomocou architektúry s veľmi malými (3×3) konvolučnými filtrami, čo poukazuje na výrazne zlepšenie oproti sieti AlexNet výmenou veľkých konvolučných filtrov (11 a 5 v prvej a druhej konvolučnej vrstve). Toto zistenie im zabezpečilo skvelé umiestnenie v súťaži ImageNet Challenge 2014. Bol to zároveň prvý rok, kedy sa chybovosť modelov hlbokého učenia dostala pod 10 %. Úspešnosť skupiny VGG dokazuje aj fakt, že práve na základoch ich sietí sa vybudovalo ďalšie množstvo iných modelov.

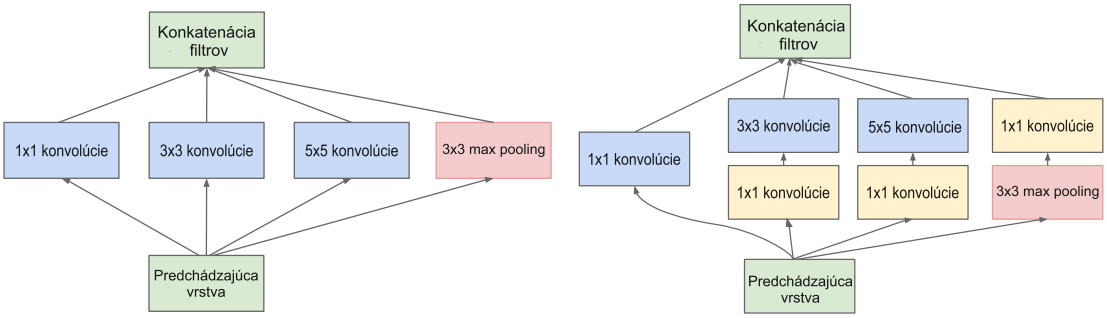
VGG16

VGG16 je CNN, ktorej meno naznačuje, že sa skladá zo 16 vrstiev. Na vstup sa implicitne vkladá RGB obrázok s veľkosťou 224×224. VGG16 má 13 konvolučných vrstiev, 5 Max Pooling vrstiev a 3 plne prepojené vrstvy, avšak len 16 z nich je váhových (majú trénovateľné váhy - konvolučné a plne prepojené). Natrénovaná na dátovej sade ImageNet dosahuje pri výbere z 5 najlepších (top-5 accuracy) úspešnosť 90,1 %[3].

VGG19

VGG19 je CNN, veľmi podobná VGG16 na vstupe sa očakáva RGB obrázok s veľkosťou 224×224 ale skladá sa z 19 vrstiev. Má 16 konvolučných vrstiev, 5 Max Pooling vrstiev a 3 plne prepojené vrstvy, avšak z nich je len 19 váhových (konvolučné a plne prepojené). Natrénovaná na dátovej sade ImageNet dosahuje pri výbere z 5 najlepších (top-5 accuracy) úspešnosť 90 %[3] .

[3]Úspešnosť je prevzatá z knižnice Keras: https://keras.io/applications/



(a) Inception modul, verzia naive

(b) Inception modul s redukciou rozmerov

Obr. 3.8: Ilustrácia inception modulu [25].

3.2.3 Inception

Inception nazývaná aj ako GoogLeNet [25] predstavená v roku 2014 tvorí veľký míľnik vo vývoji konvolučných neurónových sieti pri rozpoznaní z obrazu. V porovnaní s inými sieťami, ktoré sa počtom vrstiev prehlbujú so zámerom zvýšenia výkonnosti, Icneption ponúka skupinu trikov na zlepšenie či už rýchlosti výpočtov alebo vyššiu úspešnosť pri rozpoznaní.

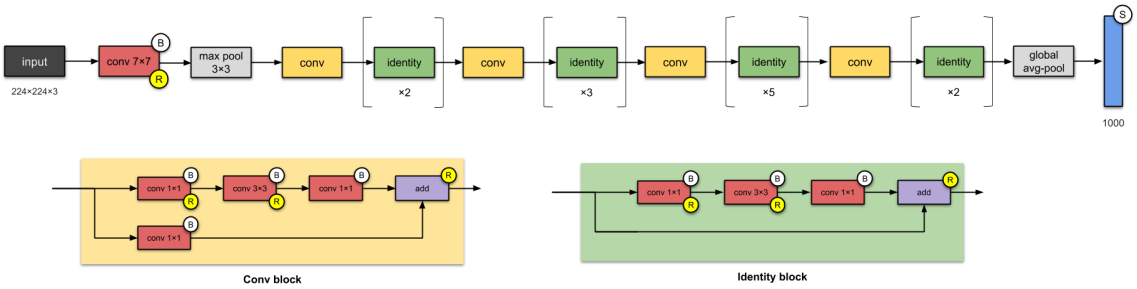
Tvorcovia sa zaoberali myšlienkou mať konvolučné filtre rôznej veľkosti na rovnakej úrovni, a tým zabezpečiť, aby sa sieť namiesto prehlbovania radšej rozšírila. Ako realizáciu tejto myšlienky vytvorili tzv. inception modul.

Na obrázku 3.8 (a) je znázornený inception modul *naive*, ktorý vykonáva konvolúcie na vstupe s 3 rôznymi filtrami (1×1, 3×3, 5×5). Okrem toho sa vykonáva aj max pooling. Výstup je prepojený a posiela sa do ďalšieho inception modulu. Avšak, tento modul by bol veľmi náročný na výpočet a tak autori limitoval počet vstupných kanálov pridaním konvolúcií s filtrom veľkosti 1×1 (obrázok 3.8 (b)). To malo za následok redukciu rozmerov, čo zmenšuje výpočtovú náročnosť. Následne sa tak využitím modulu s redukciou rozmerov vytvorila CNN Inception v1.

Inception v1 má 9 inception modulov usporiadaných lineárne. Je hlboká 22 vrstiev a využíva average pooling na konci posledného modulu. Má teda viac vrstiev ako siete AlexNet alebo VGG, ale menej ako siete ResNet. K sieti sú pridané aj dva pomocné klasifikátory, ktorými aplikovali funkciu softmax na výstupy z dvoch inception modulov a vypočítali stratu na rovnakom vstupe. Využitie týchto klasifikátorov pomáha pri probléme miznúceho gradientu (vanishing gradient problem), ktorý sa vyskytuje pri akejkoľvek hlbšej sieti.

Postupne sa potom zo základov siete Inception v1 vyvinuli ďalšie, ktoré boli vždy o niečo zmenené/vylepšené. Medzi tieto siete patria:

1. Inception v2 [26] – 2015
2. Inception v3 [26] – 2015
3. Inception v4 [24] – 2016
4. Inception-ResNet [24] – 2016 - inšpirovaná sieťou ResNet

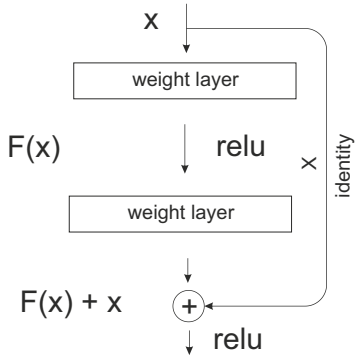


Obr. 3.9: Zjednodušená ilustrácia architektúry ResNet50 [9].

3.2.4 ResNet

S postupným vývojom konvolučných neurónových sietí sa ich počet vrstiev stále zväčšoval, čím sa dosahovala väčšia presnosť pri rozpoznaní. Avšak s nárastom počtu vrstiev v CNN sa presnosť začala postupne nasycovať (saturovať), alebo dokonca rapídne degradovať. To malo za následok postupný útlm zvyšovania presnosti, a tak systém prehlbovania CNN pomaly prestával plniť svoj účel. Týmto problémom sa zaoberala aj skupina ľudí z firmy Microsoft, a tak vytvorili CNN pod názvom ResNet.

ResNet (residual neural network) [7], [5] je jedna z najprevratnejších prác v počítačovom videní. ResNet umožňuje trénovať až stovky alebo dokonca tisíce vrstiev a stále dosahuje presvedčivý výkon. Základnou myšlienkou ResNet je zavedenie skráteného spojenia (identity shortcut connection), ktoré preskočí jednu alebo viac vrstiev, ako je zobrazené na obrázku 3.10.



Obr. 3.10: Skrátené spojenie. [7]

Poznáme viac druhov ResNet sietí. Tie sa odvíjajú od ich hĺbky, teda počtu vrstiev. Medzi najznámejšie patria ResNet34, ResNet50, ResNet101, ResNet152. Hĺbka siete sa dá jednoducho odvodiť keďže číslo z názvu označuje počet vrstiev (ResNet50 - 50 vrstiev). Platí, že čím je sieť hlbšia tým, je trénovanie ťažšie a zdĺhavejšie. Vyriešený problém s degradovaním presnosti pri väčšom množstve vrství v sieti je zobrazený v tabuľke 3.1.

model	top-1 err	top-5 err
ResNet34	25,03	7,76
ResNet50	22,85	6,71
ResNet101	21,75	6,05
ResNet152	21,43	5,71

Tabuľka 3.1: Tabuľka zobrazujúca chybovosť jednotlivých typov sietí ResNet natrénovaných na dátovej sade ImageNet. Potvrdzuje, že s rastúcim počtom vrstiev v sieti sa chybovosť znižuje. [7]

3.2.5 Vyhodnotenie modelov

Každý zo spomínaných modelov patrí k špičke konvolučných neurónových sietí pri rozpoznaní obrazu. Ako presne bude použitý model úspešný pri rozpoznávaní obrázkov z ktorejkoľvek sady, je ešte pred samotným natrénovaním veľmi ťažko určiť. Ako všeobecný prehľad úspešnosti modelov sa používa natrénovanie a následne vyhodnocovanie na už spomínanej dátovej sade ImageNet. To, že model má na dátovej sade ImageNet najvyššiu úspešnosť ešte nezaručuje, že bude najlepším riešením pre každý problém, a preto je najlepšou voľbou, ak sa pri riešení práce využije a porovná hneď viacero z týchto modelov.

Model	Top-1 Accuracy	Top-5 Accuracy
AlexNet	0.570	0.803
VGG16	0,713	0,901
VGG19	0,713	0,900
ResNet50	0,749	0,921
InceptionV3	0,779	0.937

Tabuľka 3.2: Tabuľka zobrazujúca úspešnosť modelov na dátovej sade ImageNet. Top-1 Accuracy je presnosť, pri ktorej je úspech ak má trieda očakávaného prvku najvyššiu pravdepodobnosť a Top-5 Accuracy je presnosť, pri ktorej je úspech ak sa trieda očakávaného prvku nachádza medzi 5 triedami s najvyššou pravdepodobnosťou. Hodnoty v tabuľke sú prevzaté z dokumentácie knižnice Keras.

Kapitola 4

Dátové sady

Dátové sady sú pre trénovanie konvolučných neurónových sieti veľmi dôležité a majú veľký vplyv na čo najpresnejší výsledok. Vzhľadom nato, že značiek a modelov áut je na svete veľké množstvo je o to dôležitejšie pripraviť čo najväčšie a najrôznorodejšie množstvo dát, vhodných obrázkov z rôznych uhlov a situácií určených na trénovanie. Tieto dáta musia korelovať s výstupom, ktorý očakávame pri predikcii, teda v tomto prípade ide o obrázky rôznych áut v cestnej premávky zozbierané z rôznych uhlov, či miest. Táto kapitola sa zaoberá dátovými sadami, ktoré sú nevhodné pre túto prácu, popisuje vhodnú vybranú dátovú sadu, spôsob vytvorenia dátovej sady jej spracovanie, anotácia a následne využitie.

4.1 Nevhodné dátové sady

Voľne dostupnných dátových sád pre rozpoznávanie vozidiel je veľké množstvo, ale len málo ktoré korelovali s výstupom ktorý očakávame. Problémom však je, že veľké množstvo z nich [8] nie je vhodne anotované, a teda neobsahuje pre túto prácu potrebné informácie ako je typ a model vozidla. Väčšinou sa tieto sady zameriavajú len na určitú časť vozidla (napr. predná maska) alebo sú zozbierané len z jedného miesta, a tak nereflektujú skutočnú cestnú premávku.

Iné dátové sady zase obsahujú obrázky vozidiel s veľmi dobrou kvalitou a z veľkej blízkosti a teda sa nezhodujú s reálnou situáciou cestnej premávky. Takýmto príkladom je napríklad známa sada [11] zo Stanfordskej univerzity, kde vyzbierali 16 185 fotiek 196 tried alebo sada [27], ktorá obsahuje až 291 752 fotiek 9 170 rôznych tried a model vyrobených od roku 1950 až po 2016. Ďalšou nevýhodou práve týchto sád je, že sú vytvorené v USA a je známe, že pre americký trh sa dodávajú niektoré autá odlišné či už názvom, tvarom alebo nimi európsky trh vôbec nedisponuje.

Medzi ďalšie problematické a pre túto prácu nevhodné patria dátové sady [3], ktoré obsahujú veľmi malé množstvo obrázkov a tried a teda sú nevhodné pre správne natrénovanie konvolučnej neurónovej siete a ich následné reálne využitie.

4.2 Použitá dátová sada

Vzhľadom, na spomínané nevhodné dátové sady 4.1, ktoré sú veľmi malé alebo majú zlú anotáciu, vytvorila skupina GRAPH@FIT dátovú sadu BoxCars116k[20]. Táto sada je vhodná, veľká a dobre anotovaná a preto vyhovuje práve potrebám tejto práce. Dátová sada ďalej obsahuje aj prídavok mojej vlastnej sady zhotovenej ako súčasť tejto práce.

Obr. 4.1: Náhodný výber fotiek z BoxCars116k [20].

4.2.1 Dátová sada - BoxCars116k

BoxCars116k [20] je verejne dostupná[1] dátová sada určená pre výskum, ktorú zozbierala a vytvorila skupina GRAPH@FIT na základe svojej práce o rozpoznávaní áut s využitím obaľovacieho kvádra. Táto dátová sada sa zameriava na fotky zozbierané z monitorovacích kamier. Tieto kamery boli umiestnené v blízkosti ulíc a monitorovali prechádzajúce vozidla.

Dátová sada je tvorená zo sady BoxCars21k [21], ktorá bola upravená aj s jej anotáciou a ďalšími fotkami vytvorenými z videí predchádzajúcej práce skupiny GRAPH@FIT. Táto nová dátová sada bola anotovaná viacerými ľuďmi, ktorí sa o túto problematiku zaujímajú a majú dobré znalosti o rôznych typoch a modeloch vozidiel. Ku každému vozidlu tak pridali jeho značku, model a rok výroby. Do anotácie sa zároveň pridali súradnice ohraničovacieho rámca a obaľovacieho kvádra.

	počet
trasy	27 496
fotky	116 286
kamery	137
značky	45
značky&modely	341
značky&modely&generácie	421
značky&modely&generácie&rok výroby	693

Tabuľka 4.1: Tabuľka zobrazujúca základne údaje o dátovej sade BoxCars116k [20].

Dátová sada obsahuje 27 496 vozidiel (116 286 obrázkov) 45 rôznych značiek so 693 triedami (výrobca, model, generácia a rok výroby) zozbierané zo 137 rozdielnych kamier. V porovnaní s inými dátovými sadami, BoxCars116k obsahuje zvyčajne malé obrázky vozidiel z cestnej premávky z rôznych uhlov. Prehľad informácií je v tabuľke 4.1.

Sada BoxCars116k je primárne určená k trénovaniu neurónovej sieti a jej rozpoznávaniu áut v cestnej premávke. Preto sú vytvorené rozdelenia pre trénovanie a následné ohodnotenie tak, aby reflektovali fakt, že zvyčajne nieje známy smer ktorým budú autá zaznamenávané kamerou cestnej premávky. Tak sa pre vhodné rozdelenie náhodne vybrali

[1] https://medusa.fit.vutbr.cz/traffic

	hard	medium
triedy	107	79
trénovacie a validačné kamery	81	81
testovacie kamery	56	56
trénovacie trasy	11 653	12 084
trénovacie fotky	51 691	54 653
validačné trasy	637	611
validačné fotky	2 763	2 802
testovacie trasy	11 125	11 456
testovacie fotky	39 149	40 842

Tabuľka 4.2: Tabuľka zobrazujúca informácie o prerozdelení sady BoxCars116k

kamery a ich zaznamenané trasy[2] vozidiel pre trénovanie a vozidlá z iných kamier pre testovanie. Takéto rozdelenie ma za následok, že uhol nasnímaného vozidlá z testovacej sady sa môžu jemne líšiť oproti vozidlu z trénovacej sady. Trénovacie dáta sa potom ďalej ešte rozdelili na trénovacie a validačné.

Vytvorené boli 2 rozdelenia. V prvom (hard) rozdelení sa zameriavajú na rozpoznávanie konkrétneho auta aj s rokom výroby. V druhom rozdelení (medium) sa zanedbáva rozdiel vozidla podľa roku výroby, čiže generácia vozidla a všetky generácie sú zaradené pod jednu triedu. Vybrané boli typy vozidiel, ktorých trasy boli zaznamenané kamerou v trénovacej sade aspoň 15-krát a v testovacej sade aspoň raz. Bližšie informácie k prerozdeleniu sú zobrazené v tabuľke 4.2.

4.2.2 Vlastná dátová sada

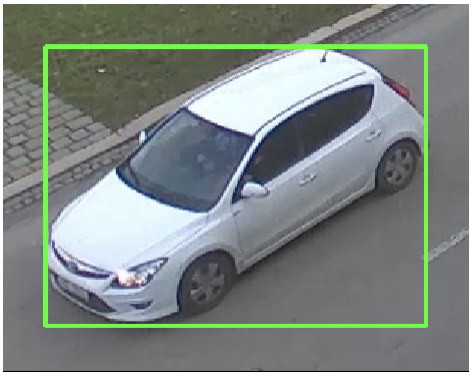
Táto dátová sada je mnou vytvorená sada, ktorá je len malý prídavok pre BoxCars116k. Zmyslom vytvorenia tejto sady bolo aby sa BoxCars116k zase o niečo zväčšila, keďže sa jedná o jednu z kľúčových vlastností pri trénovaní neurónovej siete ako už bolo spomenuté v 4 a zároveň aby som sám pochopil ako tvorba dátovej sady vyzerá.

Sada obsahuje 213 obrázkov rôznych vozidiel, ktoré boli zozbierané v okolí Brna v Českej republike. Tieto obrázky boli zhotovené z viacerých miest a uhlov a boli robené tak, aby napodobňovali zábery zhotovené pomocou kamier cestnej dopravy. Ďalšie informácie sú uvedené v tabuľke 4.3.

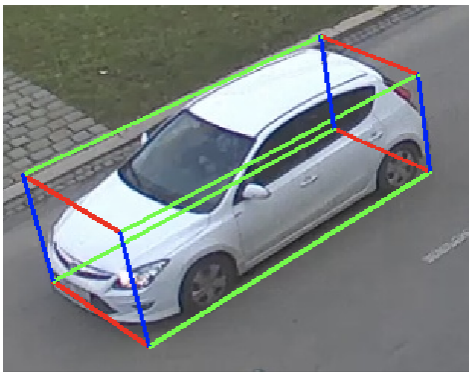
	počet
trasy	69
fotky	213
kamery	4
značky	10
značky&modely	29
značky&modely&generácie	38

Tabuľka 4.3: Tabuľka zobrazujúca základne údaje o mojej dátovej sade

[2]Pod trasou vozidla sa rozumie viacero fotiek počas jedného prejdenia vozidla pred kamerou(teda viacero fotiek môže pochádzať z jednej trasy vozidla)



Obr. 4.2: Vozidlo s 2D bounding boxom



Obr. 4.3: Vozidlo s 3D bounding boxom

4.3 Anotácia dátovej sady

Anotácia dát je dôležitá súčasť tvorby dátovej sady. Jej cieľom je priradiť v tomto prípade každému obrazu auta príslušné informácie. Tieto informácie pomáhajú neurónovej sieti na učenie, teda natrénovanie modelu a otestovanie správnosti a presnosti modelu.

4.3.1 Použitá anotácia

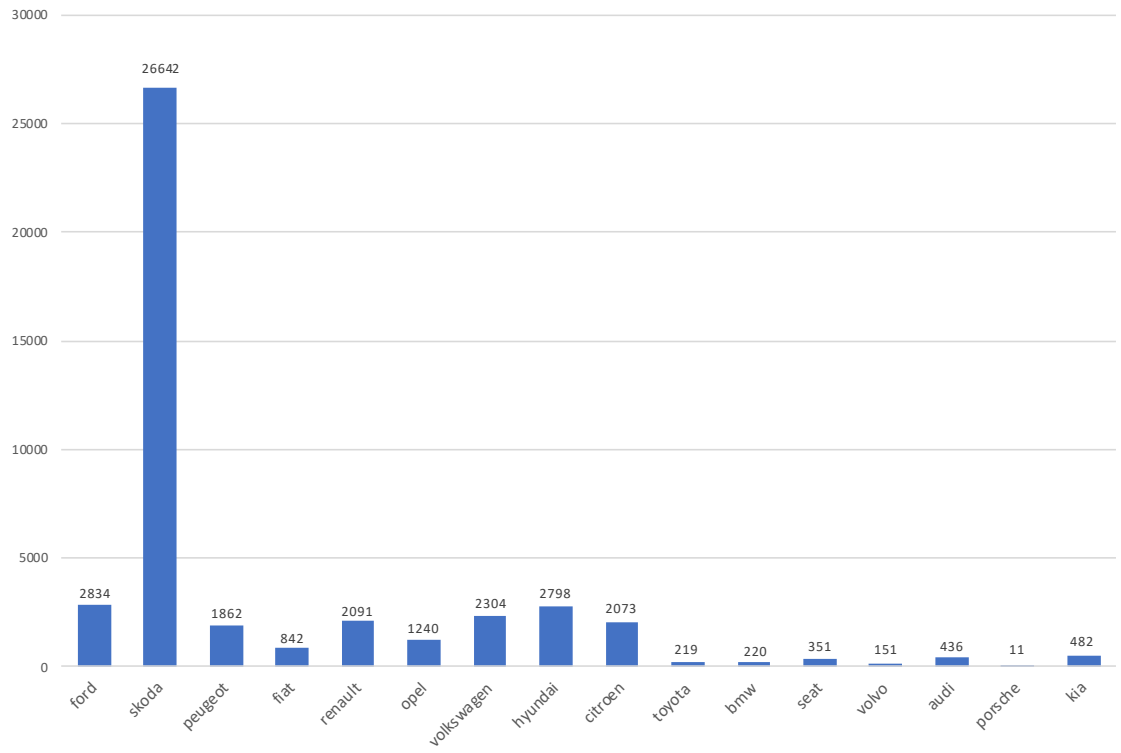
Rovnako ako v BoxCars116k, tak aj v mojej dátovej sade bola použitá rovnaká anotácia, aby bolo možné tieto dátové sady neskôr spojiť. Táto anotácia slúži na jasné určenie typu vozidla. Pri anotácií sú zároveň zapísané kalibračné údaje kamier, pomocou ktorých boli fotky zhotovené. Obidve dátové sady sú tzv. human-anotated, čiže anotované človekom. V anotácii je zapísaná vždy jedna trasa vozidla, ktorá obsahuje viacero fotiek toho vozidla. Štruktúra anotácie jednej trasy je nasledovná:

- annotation (celý názov vozidla)
- camera (názov kamery)
- flags (doplňajúce značky)
- id (identifikačné číslo)
- instances (inštancie)
  - 2DBB (súradníce 2D bounding boxu auta na fotke)
  - 3DBB (súradníce 3D bounding boxu auta na fotke)
  - instance_id (identifikačné číslo fotky)
  - path (obsahuje umiestnenie konkrétnej fotky)
- to_camera (informácia o tom, či smeruje vozidlo ku kamere alebo od kamery)

Získanie súradníc 2D bounding boxu

Nato, aby bolo možné získať súradnice 2D bounding boxu je v prvom rade nutné auto z obrazu detekovať. Na detekciu vozidiel z obrazu som v mnou vytvorenej časti dátovej sady použili detektor *YOLO*.

YOLO[3] (you only look once) je populárny algoritmus na detekovanie objektov z obrazu. Svoju popularitu si získal hlavne kvôli svojej vysokej presnosti a zároveň, je schopný fungovať s aplikáciami v reálnom čase. Tento algoritmus sa pozrie len raz (only look once) na obraz v tom zmysle, že cez sieť prejde len raz a vykoná predikciu. Na výstup sa tak dostane rozpoznané vozidlo aj s 2D bounding boxom.



Obr. 4.4: Graf zobrazujúci početné zastúpenie jednotlivých značiek áut v testovacej sade.

4.4 Problémy dátovej sady

Zhotoviť dokonalú sadu pre rozpoznávanie vozidiel v obraze je naozaj náročné až priam nemožné. Dátovú sadu využívanú v tejto práci teda BoxCars116k môžeme považovať za naozaj rozsiahlu. Ako bolo uvedené v tabuľke 4.2 obsahuje viac ako 116 000 fotiek rôznych vozidiel zo 107 tried. Problémom však je, že ani takéto veľké množstvo stále dostatočne nevyhovuje reálnej situácii, keďže sa na cestách vyskytuje ešte omnoho viac typov vozidiel, ako je obsiahnutých v BoxCars116k.

Ďalším problémom je, že automobilový trh sa hýbe závratnou rýchlosťou a každoročne sa na trh dostanú nové značky, modely a generácie vozidiel. Pre vyriešenie tejto situácie by bolo potrebné dátovú sadu priebežne dopĺňať a aktualizovať. Zároveň sa pri výrobe automobilov stretávame aj s tzv. faceliftom, teda malou kozmetickou zmenou vzhľadu, ktorá môže ale pôsobiť dosť výrazne a zmiasť tak neurónovú sieť a znížiť jej úspešnosť rozpoznávania.

Vzhľadom nato, že dátová sada bola zozbieraná z kamier cestnej premávky, obsahuje nemalé rozdiely medzi počtom vyhotovených obrázkov či už konkrétnych modelov, alebo aj značiek. Rozdiel v počte zastúpených značiek je možné vidieť na obrázku 4.4. To má

[3] https://pjreddie.com/darknet/yolo/

za následok, že sa neurónová sieť lepšie natrénuje na auto vyskytujúce sa v dátovej sade viackrát a autá, ktoré sa v sade nevyskytujú tak často nevie dobre rozoznať. Takýto konkrétny príklad zastáva napríklad model Porsche Cayenne, ktorý sa v celej dátovej sade vyskytuje len 22-krát a pri záverečnom ohodnotení ho model rozoznal s úspešnosťou 0 %. Ideálnym prípadom by bolo, ak by dátová sada obsahoval rovnaké množstvo z každého modelu. Žiaľ, zozbierať dátovú sadu, ktorá by disponovala rovnakým množstvom pre každú triedu a zároveň by bola dostatočne veľká, je veľmi obtiažne a časovo náročne.

Kapitola 5

Trénovanie modelu

Táto kapitola sa zaoberá potrebnými nastaveniami pred trénovaním, príprave dát, ich načítaniu z dátovej sady, popisom a ukážkami augmentácie. Ďalej je popísaný priebeh trénovania s ukážkami a opis vyhodnocovania natrénovaného modelu.

5.1 Konfigurácia a príprava na trénovanie

Pre trénovanie konvolučných neurónových sietí bolo na začiatku dôležité si vybrať vhodný jazyk. Rovnako dôležitý je tiež výber vhodných knižníc medzi ktoré patrí *TensorFlow*[1].

Tensorflow je open source knižnica, vytvorená firmou *Google* ako pre začiatočníkov, tak aj expertov na vytváranie modelov strojového učenia. Veľkou výhodou je, že táto knižnica spolupracuje výborne s knižnicou *Keras*[2].

Keras je vysokoúrovňová knižnica pre neurónové siete, napísaná v jazyku *Python*. Je vytvorená, aby umožňovala rýchlo a jednoducho experimentovať s hlbokými neurónovými sieťami. Disponuje širokou škálou bežne používaných stavebných blokov na tvorbu neurónových sietí ako sú optimalizátory, vrstvy, aktivačné funkcie. Okrem štandardných neurónových sietí má *Keras* podporu aj pre konvolučné a rekurentné neurónové siete. Zároveň aj architektúry už predprogramovaných verejne známych konvolučných neurónových sietí popísaných v kapitole 3. Práve tieto 2 knižnice tvoria základ celej mojej práce, ktorá bola napísaná v jazyku *Python 3.7*.

Aby bolo možné model natrénovať rýchlejšie (7 hodín), bolo potrebné mať nainštalovanú architektúru *CUDA*. Táto architektúra umožňuje aplikáciám bežať na grafických kartách od spoločnosti *NVIDIA*, čím prebiehajúce výpočty môžu byť spracované nielen na CPU ale aj GPU, čo výrazne ovplyvňuje dobu potrebnú na natrénovanie neurónovej siete.

Súčasne bolo nutné nainštalovať aj knižnicu *NVIDIA CUDA Deep Neural Network (cuDNN)*[3], ktorá je určená na akceleráciu výpočtov spojených s hlbokými neurónovými sieťami a poskytuje implementácie pre štandardné rutiny ako združovanie, normalizácia a aktivačné vrstvy.

Ďalšou veľmi dôležitou knižnicou v tejto práci bola knižnica *OpenCV*. Táto vysoko optimalizovaná knižnica sa zameriava na aplikácie v reálnom čase a spracovaním dát v počítačovom videní. Slúži hlavne na načítavanie obrázkov z dátovej sady, ich správnu konverziu kódovania a zobrazenie.

[1] https://www.tensorflow.org
[2] https://keras.io
[3] https://developer.nvidia.com/cudnn

5.2 Príprava dát

Príprava dát je dôležitou súčasťou príprav pred trénovaním modelu. Pre každé trénovanie je potrebný nejaký vstup, podľa ktorého sa následne neurónová sieť učí predpovedať výsledok. V tomto prípade sa jedná o obrázky vozidiel, ktoré sú uložené v dátovej sade. Tieto obrázky je nutné z dátovej sady vybrať, augmentovať a prispôsobiť veľkosti, ktorá je potrebná pre vstup podľa architektúry siete, ktorú chceme natrénovať.

Vzhľadom nato, že pri svojej práci používam dátovú sadu BoxCars116k, na načítavanie dat z dátovej sady používam knižnicu, ktorú pri svojej práci (spomenutá v kapitole 2.2.3) použili autori. Zároveň v nej využívam niektoré mnou vytvorené moduly, či už pre dátovú augmentáciu alebo vytvorenie ohraničovacieho rámca (2D bounding box).

Dáta v dátovej sade BoxCars116k sú obsiahnuté vo viacerých súboroch typu *pickle*. Tieto súbory sú vytvorené modulom *PICKLE*, čo je modul implementujúci binárny protokol pre serializáciu a deserializáciu objektov a štruktúr v jazyku Python. Zoznam súborov a ich popis:

- **atlas.pkl** – súbor obsahujúci obrázky v binárnej podobe
- **classification_splits.pkl** – súbor obsahujúci rozdelenie obrázkov do trénovacej, verifikačnej a testovacej sady a zároveň do úrovní: `hard` a `medium`.
- **dataset.pkl** – súbor obsahujúci anotáciu obrázkov, ktorá je popísaná v kapitole 4.3.1
- **verification_splits.pkl** – rozdelenie obrázkov pre verifikáciu

Na načítanie potrebných dát sa vytvorí trieda `BoxCarsDataset`. Táto trieda si pri svojej inicializácií načíta všetky potrebné informácie podľa toho, či majú slúžiť ako trénovacie, validačné alebo testovacie dáta a zároveň vyberá podľa zvolenej úrovne `hard` alebo `medium`. Tieto dáta si vyberá podľa rozdelenia v súbore classification_splits.pkl. Zároveň si pripraví aj súbory dataset.pkl a atlas.pkl, z ktorých bude pri trénovaní vyberať potrebné obrázky vozidiel aj s ich anotáciou pomocou vopred definovaných funkcií v triede `BoxCarsDataset`.

5.2.1 Načítavanie a generovanie dát

Pre vkladanie vstupu na trénovanie modelu, sa častokrát hlavne pri rozsiahlejších dátových sadách využíva aj generátor dát. Generátor je trieda, ktorá rozdelí dáta na dávky (batches) a postupne predkladá tieto dávky určené na trénovanie. Tento generátor zabezpečuje to, aby nebolo nutné si všetky obrázky uložiť do pamäte, ale aby sa ukladali po dávkach. Tak sa sprístupní veľká časť pamäte, ktorá sa môže použiť pri trénovaní. Generátor teda riadi celé predkladanie dát na trénovanie.

Postup generátora pri načítavaní a následnom vrátení dát pre neurónovú sieť:

1. Podľa identifikačného čísla vozidla, ktoré sa práve spracováva, si generátor uloží obrázok konkrétneho vozidla aj s jeho anotáciou
2. Pomocou metódy `get_vehicle_instance_data_2DBB` triedy `BoxCarsDataset` si uloží súradnice 2D bounding boxu vozidla z obrázku
3. Vykoná sa augmentácia dát 5.3.1
4. Za pomocí funkcie `cut_2DBB` sa vystrihne vozidlo z obrázka podľa 2D bounding boxu 5.3.2 a veľkosť obrázka sa upraví na požadovanú
5. Následne sa spracované dáta predávajú modelu na trénovanie

5.3 Augmentácia dát

Konvolučné neurónové siete potrebujú na svoje efektívne trénovanie a následne rozpoznávanie z obrazu veľké množstvo obrázkov a k tomu pomáha aj augmentácia dát. Augmentácia dát, v tomto prípade obrázkov, je veľmi nápomocná technika ako umelo vytvoriť väčšie a rôznorodejšie množstvo obrázkov. Augmentácia teda slúži na vytváranie väčšieho množstva rôznych obrázkov z dátovej sady, a tak ju umelo zväčšuje. Pomocou toho tak zvyšujeme schopnosť modelu lepšie rozpoznať správny objekt z obrazu.

Augmentácia sa vykonáva za použitia rôznych augmentačných techník ako sú napríklad: rotácia (zrotovanie obrázku o niekoľko stupňov), otočenie (vertikálne, horizontálne), priblíženie, zmena jasu, zmena kontrastu, pridanie šumu (Gaussovský, Laplaceov, Poissonov, ...), zaostrenie hrán a podobne.

Výhodou je, že tieto techniky je možné rôzne kombinovať, čím vzniká ešte väčšia variabilita pri zväčšovaní dátovej sady za použitia augmentácie.

Augmentácia môže byť vykonaná dvomi spôsobmi:

1. **Augmentácia offline alebo predbežne spracovaná** - tento typ augmentácie sa používa pred samotným začatím trénovania a slúži na zväčšenie malej dátovej sady. Obrázky, ktoré prejdú touto augmentáciou sa následne uložia medzi ostatné do dátovej sady, a tak ju umelo zväčšia. Takéto umelé zväčšenie dátovej sady môžeme považovať za užitočné, avšak pri väčšej dátovej sade musíme vziať do úvahy aj zabratie väčšieho miesta na disku.

2. **Augmentácia online alebo v reálnom čase** - tento typ augmentácie sa aplikuje v reálnom čase, čiže za behu pri trénovaní modelu. Jej zmysel je pri využití zväčša rozsiahlejších dátových sád a jej cieľom je vytvoriť väčšiu rozmanitosť medzi obrázkami, a tak neurónovú sieť lepšie pripraviť. Táto augmentácia sa aplikuje na jednotlivé obrázky alebo na dávku obrázkov. Väčšinou je aplikovaná za využitia náhodných hodnôt a tak ma za následok, že pri každej epoche model uvidí rôzne obrázky. To znamená, že sa dátová sada hneď niekoľko násobne umelo zväčší a to bez potreby uloženia augmentovaných dát, a teda šetrí miesto na disku.

Obr. 5.1: Ukážka augmentácie - horný rad zľava: Contrast, Multiply, PoissonNoise, Flip a dolný rad zľava - GaussianBlur, Scale, Solarize, Translate.

5.3.1 Použitá augmentácia

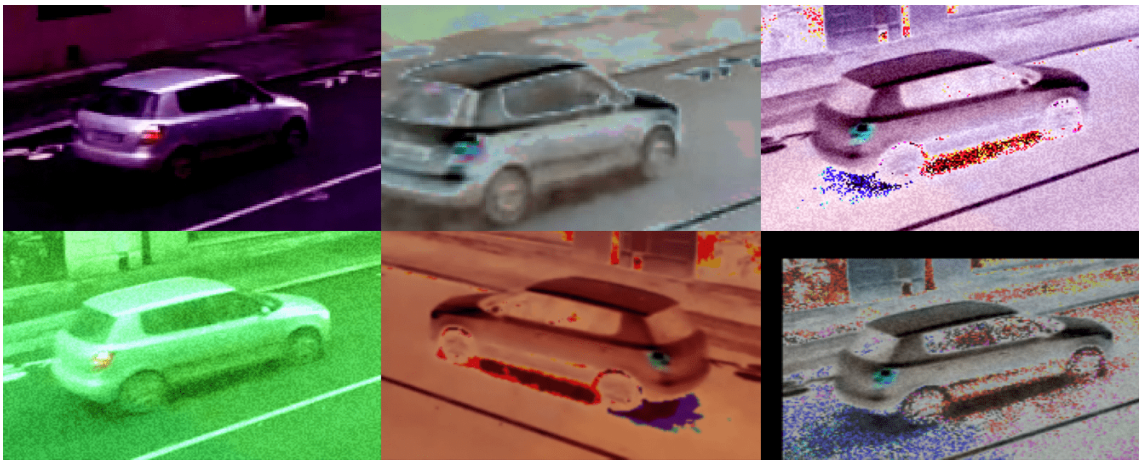
Vzhľadom na to, že dátovú sadu BoxCars116k môžeme považovať sa naozaj rozsiahlu, offline augmentácia by v tomto prípade nemala veľký zmysel, a tak som v tejto práci používal augmentáciu v reálnom čase.

Celá augmentácia je riadená generátorom, ktorý si načíta dávku obrázkov z dátovej sady a následne po jednom obrázky modifikuje. Na augmentovanie obrázkov som použil knižnicu *Imgaug*[4]. Knižnica *Imgaug* slúži na augmentáciu obrázkov v strojovom učení. Podporuje širokú škálu techník, ktorá ich umožňuje ľahko kombinovať a vyberať v náhodnom poradí. Táto knižnica dokáže nie len modifikovať obrázky, ale zároveň aj kľúčové a orientačné body, obaľovacie kvádre, tepelné mapy a segmentačné mapy.

Pri práci som použil rôznu augmentáciu. Je kombinovaná a vyberaná náhodne a z náhodných intervalov. V tejto práci bola použitá nasledujúca augmentácia (obrázok 5.1):

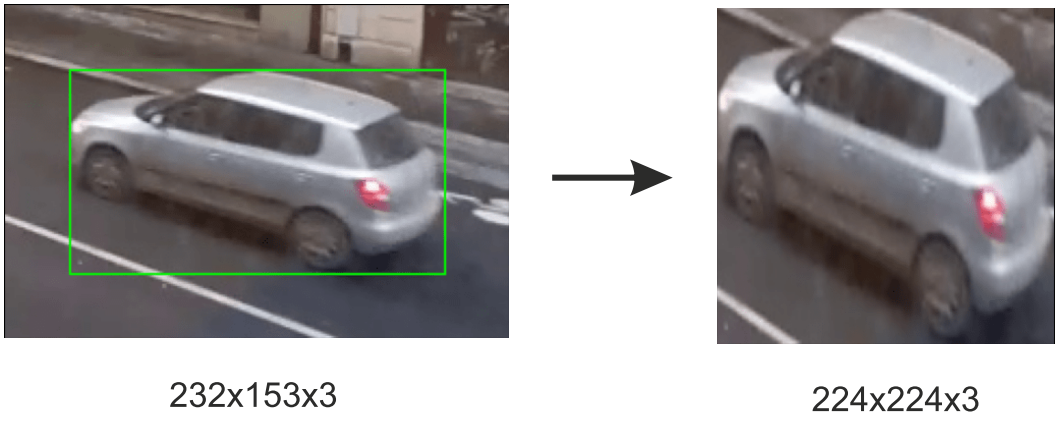
1. **PoissonNoise** – vzorkovaný šum z poissonových distribúcií
2. **Solarize** – invertovanie všetkých hodnôt pixelov v rozmedzí nastaveného intervalu
3. **Flip** – vertikálne otočenie
4. **Multiply** – vynásobenie všetkých pixelov obrázku špecifickou hodnotou
5. **Scale** – zväčšenie alebo zmenšenie obrazu
6. **Contrast** – zvýšenie alebo zníženie kontrastu
7. **Translate** – jemné posunutie obrazu do strán
8. **GaussianBlur** – rozmazanie obrazu pomocou gaussovského jadra

Každá jedna z tejto augmentácie má priradenú určitú mieru, akou ovplyvňuje obraz. Pri niektorých je táto miera určená intervalom, z ktorého sa vyberá sila danej zmeny obrazu, pri iných je zase percentuálne určené (napr. 50 % - každý druhý obrázok) ako často majú zmenu obrazu vykonať.



Obr. 5.2: Ukážka kombinovanej augmentácie

[4] https://imgaug.readthedocs.io/en/latest/index.html



Obr. 5.3: Vystrihnutie vozidla z obrázka a prispôsobenie veľkosti.

5.3.2 Vystrihnutie vozidla z obrázka

Každé vozidlo zachytené v dátovej sade je vystrihnuté zo záberu a tak predstavuje obraz aj so svojim okolím. Vzhľadom nato, že tieto vozidlá sú v obraze detekované, sú im priradené aj ich ohraničovacie rámce (2D bounding boxy). Tieto 2D bounding boxy majú za úlohu ohraničiť vozidlo od zvyšku obrazu. Pomocou tohto ohraničenia tak následne z obrázku vystrihneme len tú časť, kde sa vozidlo nachádza. To má potom za následok, že sa neurónová sieť pri učení zameriava len na tú časť (teda vozidlo), ktorá je potrebná.

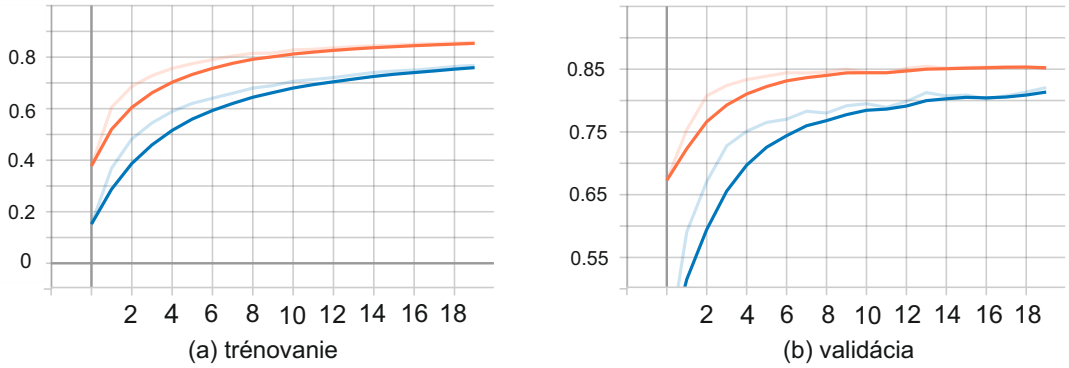
Na vystrihnutie vozidla z obrázka som v tejto práci použil funkciu `cut_2DBB`. Táto funkcia dostane na vstup súradnice 2D bounding boxu, pomocou ktorých vozidlo z obrazu vystrihne. Následne tak potom ešte veľkosť obrázka prispôsobí na požadované rozmery (väčšinou je to $224 \times 224 \times 3$).

5.4 Priebeh trénovania

Trénovanie (učenie) neurónovej siete, je veľmi časovo a zároveň aj výpočtovo náročná operácia. Na jej dĺžku má veľký vplyv hlavne veľkosť dátovej sady a výkonnosť GPU. Vzhľadom na to, že dátová sada BoxCars116k je dosť veľká, trénovanie si vyžadovalo výkonnú GPU. Veľkou výhodou pri tejto práci bolo, že som mal priamy prístup ku grafickej karte NVIDIA GeForce GTX 1080, ktorá sa dá považovať za jednu z výkonnejších grafických kariet. Táto grafická karta zabezpečovala, že trénovanie neurónovej siete tak trvalo relatívne v zvládnuteľnej dobe (približne 7 hodín).

5.4.1 Vytvorenie modelu

Ako prvé je však potrebné vytvoriť model určený na trénovanie. Jedna z možností je vytvoriť si vlastný, a to poskladaním vrstiev. Druhá možnosť je použiť už vytvorený model. V tejto práci som sa rozhodol pre využitie už vytvorených modelov, ktoré dosahujú vynikajúcu presnosť pri rozpoznávaní (popísané v kapitole 3.2.5). Zároveň bol tento model už predtrénovaný na dátovej sade ImageNet. Všetky použité modely poskytuje knižnica Keras. Po načítaní modelu z knižnice Keras je potrebné určiť základné nastavenia ako sú



Obr. 5.4: Priebeh trénovania a validácie (oranžová - ResNet50, modrá - VGG16)

napríklad: miera učenia (learning rate), metrika alebo počet tried. Po nastavení všetkého potrebného sa môže začať trénovať.

5.4.2 Trénovanie a validácia

Trénovanie prebieha v 20 epochách (okolo 20. sieť konverguje), kde v 1 epoche sieti na vstupe prídu všetky obrázky z trénovacej dátovej sady. V každej ďalšej epoche sa trénovacia sada opakuje, avšak vzhľadom nato, že je augmentácia nastavená náhodne, je augmentovaná stále inak. To poskytuje sieti náročnejšie podmienky, čo dopomáha k lepším výsledkom v úspešnosti. Sieti sú obrázky podávané v tzv. dávkach (batches), po ktorých si vždy sieť prestaví svoje váhy.

Na konci každej epochy prebehne validácia siete, ktorej sú podávané a vyhodnocované obrázky z validačnej sady. Tieto obrázky sú bez akejkoľvek augmentácie a ich vyhodnotením poskytujú prehľad o reálnej úspešnosti siete pri rozpoznaní.

5.5 Ohodnotenie trénovania a vizualizácia výsledkov

5.5.1 Ohodnotenie

Po natrénovaní sa model uloží a začína ohodnotenie (evaluation) siete. Na ohodnotenie je v dátovej sade vyčlenená časť, ktorá je iná od trénovacej a validačnej. Táto časť je poskytnutá modelu na vyhodnotenie bez akejkoľvek augmentácie. Sú to obrázky, ktoré približujú reálnu situáciu, pri ktorej má byť model použitý. Zmyslom ohodnotenia tak je poskytnúť reálne výsledky natrénovaného modelu.

Ohodnotenie modelu vykonáva funkcia `evaluate`, ktorá si z jednej trasy (track) vozidla načíta všetky obrázky a k trase si priradí svoje označenie (label). Postupne tak tieto obrázky predkladá modelu a porovnáva či klasifikácia prebehla správne alebo nie. Po vyhodnotení všetkých obrázkov z trasy vozidla určí úspešnosť a prechádza k trase druhého vozidla.

Funkcia `evaluate` poskytuje 2 typy vyhodnotenia:

1. Celková úspešnosť (accuracy) – úspešnosť pri vyhodnotení každého obrázka z testovacej sady

2. Úspešnosť jednotlivých trás (track accuracy) – úspešnosť kedy sa vyhodnotenie každého obrázka z trasy vozidla spriemeruje a podľa priemeru sa určí či vozidlo bolo správne rozpoznané

5.5.2 Vizualizácia výsledkov

Vizualizácia výsledkov tvorí veľkú súčasť pri trénovaní neurónových sietí. Poskytuje užívateľovi jednoduchý prehľad o tom, ako a či prebiehalo trénovanie správne. Poprípade ukazuje na možné chyby, ktoré mohli vzniknúť.

V tejto práci som k vizualizácií používal knižnice Matplotlib[5] na vykresľovanie rôznych grafov a zobrazenie obrázkov z dátovej sady. Za veľmi užitočnú považujem aj knižnicu Scikit-learn[6]. Táto knižnica obsahuje veľké množstvo užitočných nástrojov pre strojové učenie, tvorbu štatistických modelov a podobne. Jej využitie som našiel aj pri tvorbe matice zámen (confusion matrix).

[5] https://matplotlib.org
[6] https://scikit-learn.org/stable/

Kapitola 6

Experimenty a zhodnotenie výsledkov

Táto kapitola sa skladá z dvoch časti. Prvá časť popisuje dosiahnuté výsledky rozpoznania vozidiel z obrazu, s využitím ohraničovacieho rámca (2DBB) a s využitím augmentácie. Následne porovnáva tieto výsledky s prácou [20] využívajúcou obaľovací kváder (3D bounding box) okolo vozidla. Druhú časť tvoria rôzne experimenty, ktoré som vyskúšal pri tejto práci a popisuje ich vplyv na úspešnosť pri rozpoznaní vozidiel. Všetky merania a experimenty boli vykonávané na dátovej sade BoxCars116k popísanej v kapitole 4.2.

6.1 Využitie 2D bounding boxu

Pred samotným rozpoznaním vozidiel je potrebné vozidlo z obrazu ako prvé detekovať. Na detekciu sú využívané rôzne detektory (napr. SSD[1] alebo YOLO[2]). Tieto detektory detekujú vozidlo na obraze a vykreslia ohraničovací rámec (2D bounding box/2DBB) okolo vozidla. Myšlienkou je teda využiť tento 2D bounding box a vystrihnúť pomocou neho vozidlo z obrázku, ako je to popísané a znázornené v kapitole 5.3.2. Vplyvom toho tak je, že sieti budeme dávať na vstup priamo len obrázky obsahujúce vozidlá. Ďalšou výhodou by bolo, že ak by model dosahoval vyššie výsledky pri rozpoznaní s využitím 2D bounding boxu, bolo by možné obrázky podľa toho vystrihnúť natrvalo, a tak znížiť veľkosť dátovej sady.

Z tabuľky 6.1 jasne vidieť, že vystrihnutie obrázka pomocou 2DBB súradníc nemá pozitívny vplyv na správne rozpoznanie vozidla z obrazu, práve naopak prispieva k zhoršeniu výsledkov. Najmenší pokles v presnosti môžeme pozorovať pri sieti ResNet50, kde pri využití 2DBB došlo k poklesu presnosti pri rozpoznaní o približne 1 %. Najväčší pokles zase môžeme pozorovať pri sieti InceptionV3, kde pri využití 2DBB došlo k poklesu pri rozpoznaní o viac ako 3,5 %. Veľmi podobné prípady môžeme pozorovať aj pri úspešnosti trasy (track accuracy), a to súčasne pri porovnaní situácií kde za úspech považujeme keď vozidlo patrí do triedy s najvyššou pravdepodobnosťou ako situácií, kde za úspech považujeme keď vozidlo patrí do 1 triedy z 5 s najvyššou pravdepodobnosťou.

[1] https://arxiv.org/abs/1512.02325
[2] https://pjreddie.com/yolo/

model	top1-acc	top5-acc
VGG16	74,57/82,20	89,97/92.78
VGG16(2DBB)	72,06/81.02	87,36/91,01
VGG19	74,91/82,78	90,04/93,66
VGG19(2DBB)	71,90/79,95	87,43/92,00
ResNet50	**76,12/83,94**	**91,48/94,92**
ResNet50(2DBB)	75,08/83,69	90,80/95,55
InceptionV3	74,28/81,99	89,90/93,83
InceptionV3(2DBB)	70,78/80,09	87,90/93,47

Tabuľka 6.1: Tabuľka zobrazujúca úspešnosť modelov pri rozpoznaní vozidla z obrazu, kde je porovnávaný model využívajúci a nevyužívajúci 2DBB pri trénovaní a ohodnotení. Výsledky zobrazujú 2 typy presností: top1-acc – zobrazuje úspešnosť, kedy triede, ktorá patrí vozidlu na vstupe bola priradená najvyššia pravdepodobnosť a top5-acc – zobrazuje úspešnosť, kedy trieda, ktorá patrí vozidlu na vstupe patrí medzi 5 tried s najvyššou pravdepodobnosťou. Všetky hodnoty sú uvedené v percentách a zapísané ako – celková úspešnosť/úspešnosť trasy.

6.2 Vplyv augmentácie

Zatiaľ čo využitie 2DBB veľký úspech neprinieslo, augmentácia dát dopomohla k výraznejšiemu zlepšeniu. Augmentácia dát, ktorá je popísaná v kapitole 5.3.1 priniesla výraznejšie zmeny a to zvýšením presností. Najvýraznejší nárast je možné pozorovať u architektúry ResNet50, kde sa presnosť rozpoznania vozidiel zvýšila až o 8,15 %. Ďaleko nezaostáva ani sieť VGG16, kde sa augmentáciou podarilo navýšiť úspešnosť pri rozpoznaní o 7,4 %. Najmenší, no aj tak dosť veľky vplyv mala augmentácia na model VGG19, kde sa presnosť rozpoznania zvýšila o 6,64 %. Rovnakú postupnosť môžeme pozorovať aj pri úspešnosti z vyhodnotených trás vozidiel. Ak sa pozrieme na presnosť rozpoznania z 5 najlepších tried, najvyšší nárast môžeme pozorovať u sieti VGG16, zatiaľ čo najmenší pri sieti ResNet50. V konečnom dôsledku najvyššiu úspešnosť zo všetkých sieti s využitím augmentácie dosahuje sieť ResNet50 s úspešnosťou 84,27 % pri top1-acc a úspešnosťou 95,11 % pri top5-acc.

model	top1-acc	top5-acc
VGG16	74,57/82,20	89,97/92.78
VGG16(aug)	81,97/88,91	94,56/96,90
VGG19	74,91/82,78	90,04/93,66
VGG19(aug)	81,55/88,33	94,48/96,81
ResNet50	76,12/83,94	91,48/94,92
ResNet50(aug)	**84,27/90,68**	**95,11/97,23**
InceptionV3	74,28/81,99	89,90/93,83
InceptionV3(aug)	81,25/88,18	93,73/96,12

Tabuľka 6.2: Tabuľka zobrazujúca úspešnosť modelov pri rozpoznaní vozidla z obrazu, s využitím a bez využitia augmentácie. Výsledky zobrazujú 2 typy presností: top1-acc – zobrazuje úspešnosť, kedy triede, ktorá patrí vozidlu na vstupe bola priradená najvyššia pravdepodobnosť a top5-acc – zobrazuje úspešnosť, kedy trieda, ktorá patrí vozidlu na vstupe patrí medzi 5 tried s najvyššou pravdepodobnosťou. Všetky hodnoty sú uvedené v percentách a zapísané ako – celková úspešnosť/úspešnosť trasy.

model	obrázok+aug	2DBB+aug	3DBB+aug [20]	3DBB odhadom [20]
VGG16	81,97/88,91	80,75/88,75	**84,13/92,27**	80,60/90,69
VGG19	81,55/88,33	80,77/89,2	**84,12/92,00**	81,43/91,57
ResNet50	**84,27**/90,68	83,03/90,68	82,27/**90,79**	79,60/90,40

Tabuľka 6.3: Porovnanie úspešnosti pri rozpoznaní vozidiel na dátovej sade BoxCars116k. Modelom boli poskytnuté 4 druhy vstupných dát: 1. obrázok+aug (nemodifikované obrázky z dátovej sady, ktoré prešli navrhnutou augmentáciou), 2. 2DBB+aug (Obrázky z dátovej sady po vystrihnutí podľa súradníc 2DBB a aplikovanou navrhnutou augmentáciou), 3. 3DBB+aug (Využitie 3DBB s augmentáciou z práce [20]), 4. 3DBB odhadom (Využitie 3DBB odhadom z práce [20])

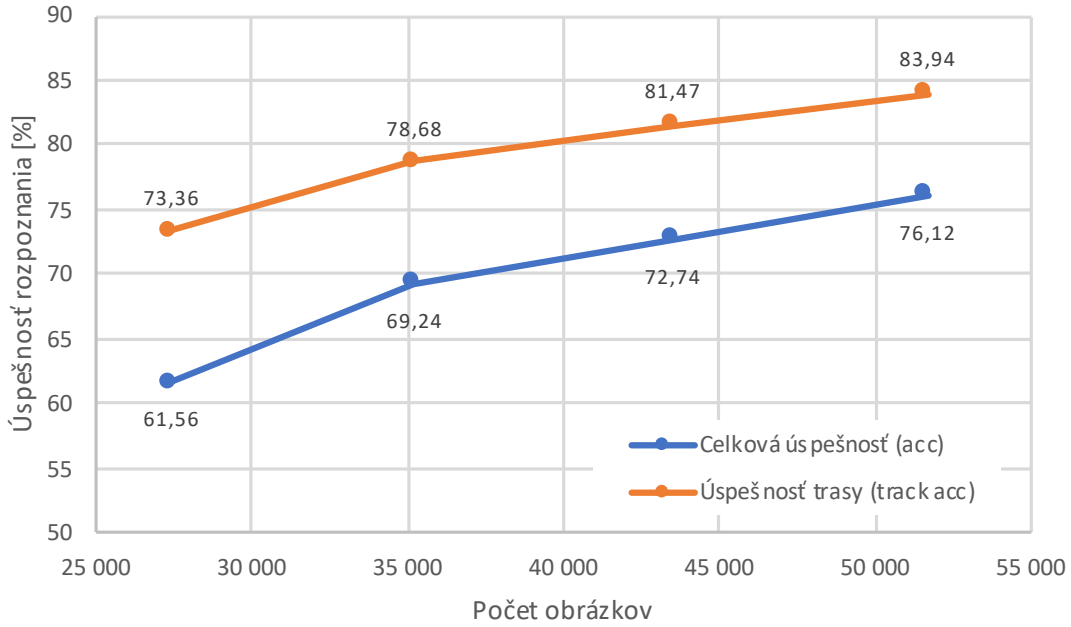
6.3 Porovnanie s 3D bounding boxom

Podľa práce [20] popísanej v kapitole 2.2.3 má využitie obaľovacieho kvádra (3D bounding boxu/3DBB) veľký vplyv na presnosť pri rozpoznávaní vozidiel a dokáže zvýšiť presnosť až o 11 %. Problémom však je, že vytváranie 3D bounding boxu je pomerne zložité a zaberá čas. Tento prístup dokáže presnosť pri rozpoznávaní zvýšiť, avšak jeho použitie v aplikáciách v reálnom čase by bolo problematické. Z tohto dôvodu som sa snažil len s využitím dátovej augmentácie a ohraničovacieho rámca (2D bounding box/2DBB), priblížiť k výsledkom, ktoré boli dosiahnuté s 3DBB a augmentáciou, ktorú pri tejto práci autori využívali. Zároveň porovnávam dosiahnuté výsledky s výsledkami kde bol využitý tzv. 3DBB odhadom (estimated 3DBB), čiže nevzniká dlhým pozorovaním videa, ale jednoduchým odhadom.

V tabuľke 6.3 môžeme vidieť, že v porovnaní s 3DBB sa mi pomocou augmentácie podarilo pri modeloch VGG16 a VGG19 priblížiť k presnosti rozpoznania s rozdielom 2 %. Pokiaľ sa jedná o porovnanie s využitím 3DBB odhadom, je jasne vidieť, že VGG16 sa mi podarili pomocou augmentácie prekonať presnosť o takmer 1,5 %, zatiaľ čo pri VGG19 sa mi pomocou augmentácie podarilo dosiahnúť výsledok, ktorý poskytla sieť s využitím 3DBB odhadom. Skutočný rozdiel je však možné pozorovať pri sieti ResNet50, kde sa mi za pomoci augmentácie podarilo v presnosti dokonca prevýšiť všetky modeli, či už tie ktoré využívajú klasický 3DBB s augmentáciou alebo 3DBB odhadom. Konkrétne sieť ResNet50 dosahuje o 2 % vyššiu úspešnosť oproti modelu s 3DBB a takmer o 4 % s 3DBB odhadom. Čo si ale môžeme všimnúť je, že presnosť pri trase vozidla (track accuracy) sa mi v žiadnom prípade nepodarilo augmentáciou prekonať ani za využitia 2DBB, ani bez.

6.4 Experimenty

Nasledujúca podkapitola popisuje vykonané experimenty na dátovej sade BoxCars116k pomocou modelu ResNet50 pri vyhodnotení výsledkov. Experimenty majú za úlohu zamerať sa na určitú situáciu a sledovať aký vplyv má na úspešnosť modelu pri rozpoznaní. Tieto experimenty tak majú priblížiť, čo a ako môže ovplyvniť úspech rozpoznania. Pomocou toho je tak možné pozrieť sa na to, čo by bolo možné ešte zlepšiť a tak presnosť rozpoznania ešte zvýšiť.



Obr. 6.1: Graf znázorňujúci vplyv veľkosti dátovej sady na úspešnosť rozpoznania.

6.4.1 Vplyv veľkosti dátovej sady

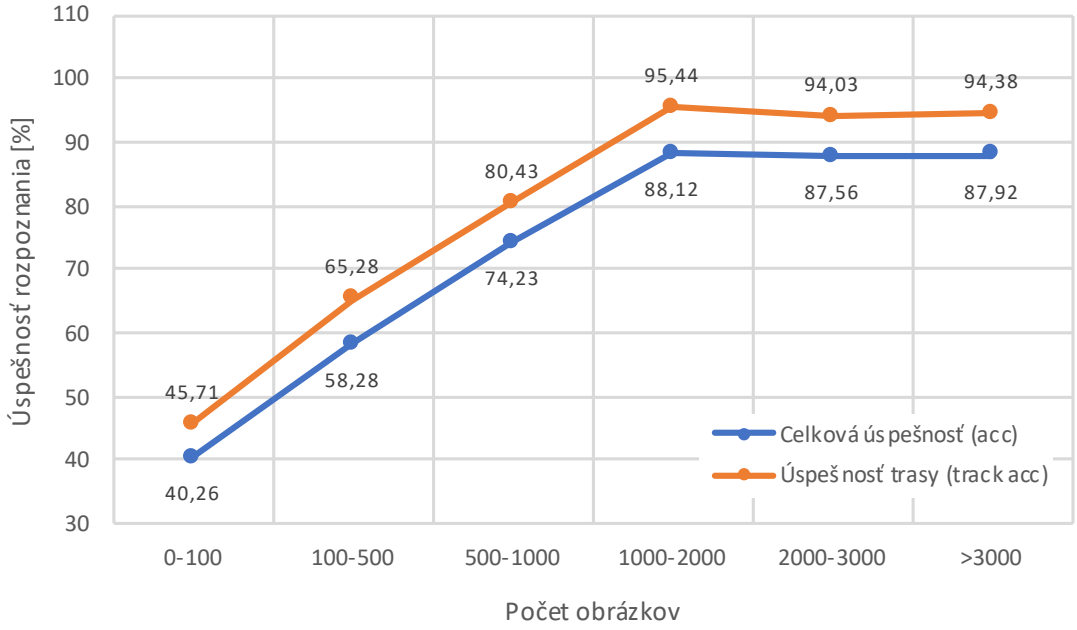
Dátová sada, ako je popísané v kapitole 4, je dôležitou súčasťou počítačového videnia. V tomto experimente sa zameriavam na preukázanie vplyvu veľkosti dátovej sady na úspešnosť modelu pri rozpoznaní vozidiel.

Základná veľkosť trénovacej sady pri rozdelení hard je 51 691 obrázkov z 11 653 vozidiel (rozdelenie je popísané v kapitole 4.2). Postupom tohto experimentu bolo znížiť počet vozidiel o 1000 a porovnať tak vplyv na presnosť rozpoznania. Počet vozidiel som tak znížil celkovo 3-krát o 1000, čím som sa dostal približne na polovičnú veľkosť trénovacej sady.

Ako je možné vidieť z obrázku 6.1, veľkosť dátovej sady naozaj vplýva na úspešnosť rozpoznania. Zatiaľ čo pri počte obrázkov 51 691(11 653 vozidiel) sa úspešnosť pohybuje na úrovni 76 %, veľký rozdiel môžeme pozorovať každým znížením počtu vozidiel a ich obrázkov. Pri poslednom znížení množstva vozidiel na počet 8653, a teda zníženiu počtu obrázkov na 27 428, čo predstavuje takmer 50 % zníženie zo základnej dátovej sady, je možné pozorovať úspešnosť rozpoznania na úrovni 62 %, čo predstavuje pokles v úspešnosti o viac ako 14 %. Celý prehľad je možné vidieť aj v tabuľke 6.4.

počet obrázkov	**počet vozidiel**	**úspešnosť [%]**	**rozdiel v úspešnosti [%]**
51 691	11 653	76,12/83,94	0/0
43 512	10 653	72,74/81,47	3,38/2,47
35 227	9653	69,24/78,68	6,88/5,26
27 428	8653	61,56/73,36	14,56/10,58

Tabuľka 6.4: Prehľad zníženia počtu obrázkov dátovej sady.



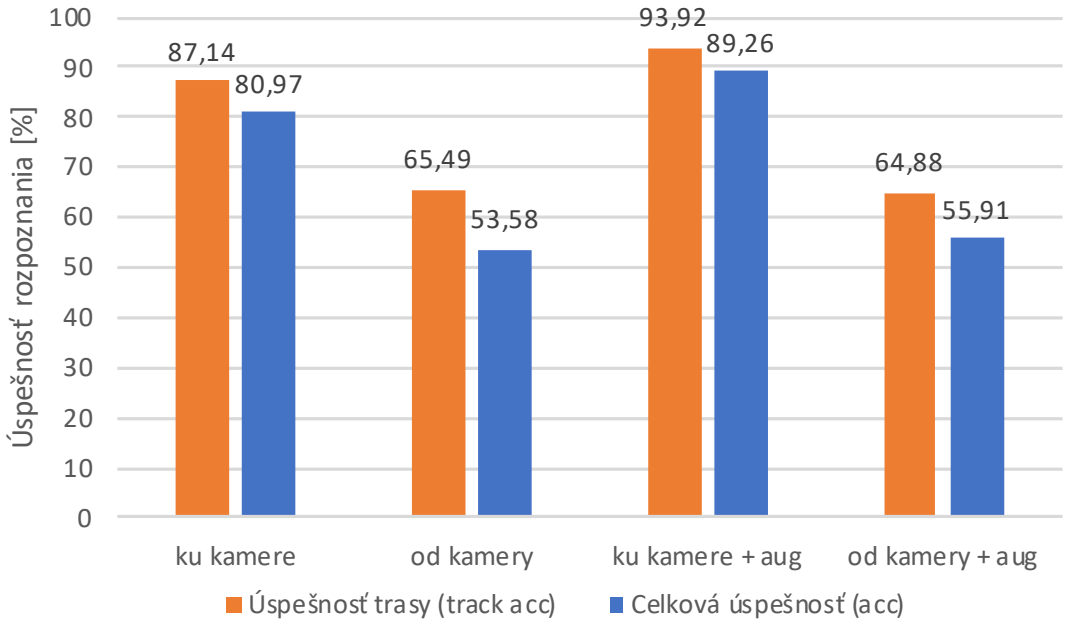
Obr. 6.2: Graf znázorňujúci vplyv počtu obrázkov jednej triedy na jej úspešnosť pri rozpoznaní vozidiel.

6.4.2 Vplyv počtu obrázkov jednej triedy

Ako je možné pozorovať v prvom experimente, zmenšenie dátovej sady naozaj ovplyvňuje úspešnosť rozpoznania. Tento experiment je zameraný na vplyv počtu obrázkov pri konkrétnom vozidle/triede.

V tomto experimente som rozdelil triedy do 6 skupín podľa ich počtu obrázkov z trénovacej sady, na ktorých sa môže model trénovať. Tieto skupiny predstavujú intervaly počtu obrázkov z tried (napríklad interval 0-100 tvoria triedy, ktoré majú v trénovacej sade počet obrázkov do 100). Z týchto obrázkov som tak vypočítal priemernú úspešnosť rozpoznania. V tomto experimente chcem dokázať, že malý počet obrázkov pri jednej triede má negatívny vplyv na úspešnosť rozpoznania vozidla danej triedy a teda zväčšovanie dátovej sady nemá zmysel, ak dôjde k zvýšeniu počtu obrázkov triedy, ktoré disponujú vysokým počtom obrázkov.

Ako je možno z obrázka 6.2 spozorovať, počet obrázkov v danej triede má vplyv na jej úspešnosť pri rozpoznaní. Z obrázku jasne vyplýva, že nízky počet obrázkov danej triedy spôsobuje nižšiu úspešnosť. Úspešnosť rozpoznania výrazne rastie s počtom obrázkov až do intervalu 1000-2000. Od tohto intervalu je možné pozorovať, že úspešnosť už ďalej nestúpa, a tak je jasné že zväčšovanie dátovej sady pridávaním obrázkov do týchto tried by vyššiu úspešnosť už neprinieslo. Experimentom som teda zistil, že úspešnosť rozpoznania by sa dala výraznejšie zvýšiť zväčšením dátovej sady, ale hlavne pre triedy, ktoré nedisponujú vysokým počtom obrázkov.



Obr. 6.3: Graf znázorňujúci vplyv smeru vozidla na úspešnosť pri rozpoznaní.

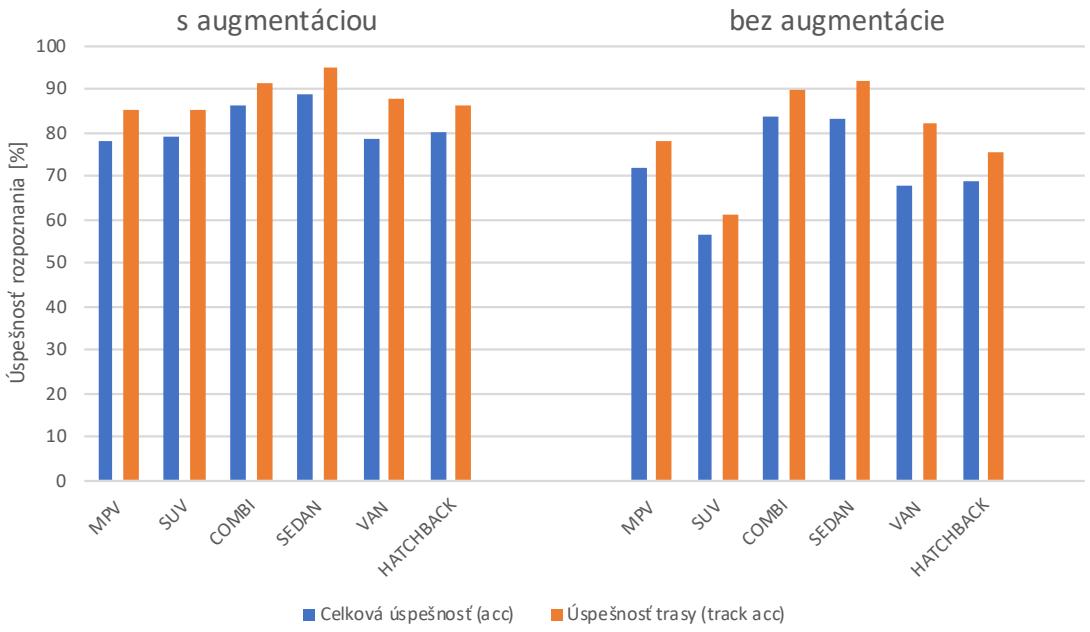
6.4.3 Vplyv smeru vozidla

Vozidla sú v dátovej sade uložené či už smerujúce ku kamere ktorou boli zachytené, alebo naopak od kamery. Toto označenie nájdeme aj v anotácií pri každom vozidle, kde parameter `to_camera` naznačuje smer vozidla.

V tomto experimente som sa zameral na vplyv smeru vozidla na presnosť rozpoznania. Tento experiment teda dopomôže k zisteniu, či má vplyv to, akou stranou je auto snímané respektíve, či dokáže predná alebo zadná časť vozidla lepšie dopomôcť k jeho rozpoznaniu. Meranie prebiehalo, ako na dátovej sade bez úprav, tak aj na dátovej sade s augmentáciou.

Ako je možné z obrázku 6.3 vidieť, úspešnosť medzi tým či vozidlo smeruje ku kamere a teda ho vidíme spredu alebo smeruje od kamery a vidíme ho zozadu je veľmi rozdielna. Ak by sme obmedzili rozpoznanie len na vozidlá snímané spredu, úspešnosť by dosahovala viac než 80 % a s využitím augmentácie takmer 90 %. Úspešnosť trasy by sa tak pohybovala na úrovni 87 % a s augmentáciou až takmer 94 %. Úplne iný prípad je však pri snímaní vozidiel zozadu. Úspešnosť dosahuje len niečo viac ako 53 % a s využitím augmentácie je to len takmer 56 %. Ešte menší rozdiel môžeme pozorovať pri úspešnosti trasy, kde dokonca dochádza poklesu úspešnosti pri využití augmentácie. Obrázky bez augmentácie dosahovali úspešnosť 65,5 %, zatiaľ čo s augmentáciou je to len 64,9 %.

Vzhľadom na tieto výsledky som porovnal počet vozidiel nasnímaných spredu a zozadu z trénovacej sady a zistil som, že je medzi nimi veľký rozdiel. Počet vozidiel, ktoré smerujú ku kamere je 43 775, avšak počet vozidiel smerujúcich od kamery je 7916. Práve tento rozdiel je hlavnou príčinou, prečo je rozpoznanie vozidiel spredu o toľko úspešnejšie ako zozadu. Je tak jasné, že dátovú sadu by bolo ideálne zväčšiť hlavne o vozidlá smerujúce od kamery, a tým tak poskytnúť sieti viac obrázkov na trénovanie, čo by pravdepodobne dopomohlo k vyššej úspešnosti pri vozidlách nasnímaných zozadu a teda aj k celkovej úspešnosti.



Obr. 6.4: Graf znázorňujúci úspešnosť pri rozpoznaní podľa typu vozidla.

6.4.4 Vplyv typu vozidla

Vozidlá je možné podľa ich tvaru rozdeliť na niekoľko typov. V dátovej sade BoxCars116k je 6 takýchto typov a to: MPV, SUV, VAN, combi, sedan a hatchback. Každý typ sa od iného niečím odlišuje, a tak som sa v tomto experimente zameral na to, či má typ vozidla vplyv na úspešnosť pri rozpoznaní.

Z obrázku 6.4, ktorý znázorňuje graf na ktorom je možné vidieť úspešnosť rozpoznania pre každý typ vozidla je jasné vidieť, že v druhej polovici grafu (bez využitia augmentácie) je rozdiel pozorovateľný medzi jednotlivými typmi. Zatiaľ čo vozidlá typu combi a sedan dosahovali pri rozpoznaní celkovú úspešnosť až viac ako 83 %, vozidlo typu VAN dosahovalo úspešnosť 67 % a SUV dokonca len 56 %. Veľmi podobnú situáciu môžeme pozorovať aj pri úspešnosti trasy. Rozdiel medzi typom s najvyššou percentuálnou úspešnosťou rozpoznania a typom s najnižšou predstavuje viac ako 27 %.

Ak sa však pozrieme na prvú polovicu grafu, teda s využitím augmentácie, je možné si všimnúť, že rozdiely medzi jednotlivými typmi sa výraznejšie dorovnali. Typ vozidla sedan, čiže typ s najvyššou úspešnosťou rozpoznania dosahuje úspešnosť až takmer 89 %. Veľmi podobne je na tom aj typ combi, ktorého úspešnosť je viac ako 86 %. Pri porovnaní s rozpoznávaním, ktoré pri trénovaní nevyužilo augmentáciu môžeme vidieť, že úspešnosť typu sedan sa zvýšila o približne 5 % a pri type combi o približne 3 %. O niečo menší nárast môžeme pozorovať pri úspešnosti trasy. Pri pohľade na úspešnosť typov SUV a VAN, ktoré bez použitia augmentácie dosahovali výrazne nižšiu úspešnosť, s využitím augmentácie sa typ SUV dotiahol až na 79 % čo predstavuje zvýšenie celkovej úspešnosti až o 23 % a typ VAN sa dotiahol na celkovú úspešnosť až takmer 79 %, čo predstavuje zlepšenie v rozpoznaní o takmer 11 %.

typ vozidla	**počet**	**úspešnosť [%]**	**úspešnosť+aug [%]**	**rozdiel [%]**
MPV	2038	71,65/78,10	78,20/85,03	6,55/6,93
SUV	1855	56,26/60,94	79,12/85,19	22,86/24,25
COMBI	20626	83,73/89,97	86,25/91,31	2,52/1,34
SEDAN	7796	83,11/92,05	88,95/95,06	5,84/3,01
VAN	6339	67,60/81,86	78,51/87,76	10,91/5,9
HATCHBACK	13037	68,63/75,34	79,90/86,40	11,27/11,06

Tabuľka 6.5: Prehľad úspešnosti rozpoznania podľa typu vozidla, kde počet - je počet obrázkov daného typu v trénovacej sade, úspešnosť - predstavuje úspešnosť rozpoznania na testovacej sade, úspešnosť+aug - je úspešnosť rozpoznania natrénovaného modelu na augmentovaných dátach trénovacej sady a rozdiel - predstavuje rozdiel úspešnosti s agumentáciou a úspešnosti bez augmentácie.

Pri pohľade na typy vozidiel a ich úspešnosť za použitia augmentácie je možné vidieť, že augmentácia veľmi pozitívne vplýva na vozidlá, ktoré dosahovali nižšiu úspešnosť v porovnaní s ostatnými. Napomáha k ich úspešnému rozpoznaniu, a tak znižuje rozdiel v úspešnosti medzi jednotlivými typmi vozidiel.

Kapitola 7

Záver

Ako základ pre riešenie tejto práce som si naštudoval problematiku rozpoznania vozidiel v obraze. V práci som tak popísal aktuálne metódy, ktoré pristupujú k rozpoznaniu vozidiel. Ako súčasť štúdie bolo potrebné pochopiť fungovaniu a architektúre konvolučnej neurónovej siete, zoznámiť sa s najznámejšími architektúrami a porovnať ich úspešnosť pri rozpoznaní. Súčasne bolo potrebné pozrieť sa na voľne dostupné dátové sady, ich výhody a nedostatky.

Cieľom tejto práce bolo pristúpiť k riešeniu rozpoznania vozidiel v obraze bez použitia či už 3D modelov alebo obaľovacieho kvádra, ktorých vytvorenie má vplyv na dĺžku pri rozpoznaní, a tak je ich využitie v aplikáciách v reálnom čase problematické. Zameral som sa na porovnanie úspešnosti rozpoznania s využitím a bez využitia ohraničovacieho rámca (2D bounding box). Z výsledkov jasne vyplýva, že použitie ohraničovacieho rámca jasne zhoršuje správne rozpoznanie vozidiel, a to v niektorých prípadov aj o viac ako 3 %. Následne som skúsil aplikovať dosť výraznú augmentáciu na vstupné dáta. Tento postup sa ukázal ako veľmi užitočný, kedy som dokázal zvýšiť presnosť pri rozpoznaní vozidiel o viac ako 7 %. Tieto výsledky som porovnal s prácou využívajúcou obaľovací kváder (3D bounding box) a poukázal som na to, že s augmentáciou sa mi podarilo výrazne priblížiť k výsledkom z práce s využitím obaľovacieho kvádra a dokonca pri sieti ResNet50 som úspešnosť prekonal o 2 %.

Súčasťou práce bolo aj vykonanie rôznych experimentov na dátovej sade BoxCars116k a ich vyhodnotenie. Tieto experimenty poukazujú na vplyv rôznych činiteľov dátovej sady na úspešnosť rozpoznania. Zároveň som vyhotovil vlastnú dátovú sadu, ktorá je síce malá ale vhodne anotovaná a môže doplniť iné väčšie dátové sady.

Táto práca sa zameriavala na to, či a aký vplyv má na presnosť pri rozpoznaní vozidiel ohraničovací rámec a dosť výrazná augmentácia dát. Celková úspešnosť rozpoznania vozidiel z obrazu by sa pravdepodobne dala ešte zvýšiť, a to buď využitím novších architektúr konvolučných neurónových sietí alebo iným prístupom, ako napríklad poskytnúť sieti naraz 2 vstupy: jeden s využitím obaľovacieho kvádra a jeden s obrázkom bez modifikácie poprípade s augmentáciou. Zároveň by bolo vhodné doplniť dátovú sadu BoxCars116k o vozidlá, ktorých je v tejto sade málo, čím by mohlo dôjsť k zvýšeniu úspešnosti rozpoznania.

Literatúra

[1] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q. et al. *ShapeNet: An Information-Rich 3D Model Repository.* arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[2] Dataman, D. What Is Image Recognition? *Medium* [online]. 2018. Dostupné z: https://towardsdatascience.com/module-6-image-recognition-for-insurance-claim-handling-part-i-a338d16c9de0.

[3] Dua, D. a Graff, C. *UCI Machine Learning Repository.* 2017. Dostupné z: http://archive.ics.uci.edu/ml.

[4] Felzenszwalb, P. F., Girshick, R. B., McAllester, D. a Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2010, zv. 32, č. 9, s. 1627–1645.

[5] Fung, V. An Overview of ResNet and its Variants. *Medium* [online]. 2017. Dostupné z: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035.

[6] Ge, Z., Mccool, C., Sanderson, C., Bewley, A., Chen, Z. et al. Fine-grained bird species recognition via hierarchical subset learning. In:. September 2015, s. 561–565. DOI: 10.1109/ICIP.2015.7350861.

[7] He, K., Zhang, X., Ren, S. a Sun, J. Deep Residual Learning for Image Recognition. *CoRR.* 2015, abs/1512.03385. Dostupné z: http://arxiv.org/abs/1512.03385.

[8] HU, L.-j., DENG, J.-h., LUO, X.-c. a HU, J. A Large-Scale Car Recognition Dataset Based on Deep Learning. In: World Scientific. *Artificial Intelligence Science And Technology-Proceedings Of The 2016 International Conference (Aist2016).* 2017, s. 26.

[9] Karim, R. Illustrated: 10 CNN Architectures. *Medium* [online]. 2019. Dostupné z: https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#e4b1.

[10] Khosla, A., Jayadevaprakash, N., Yao, B. a Li, F.-F. Novel dataset for fine-grained image categorization: Stanford dogs. In: *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC).* 2011, sv. 2, č. 1.

[11] Krause, J., Stark, M., Deng, J. a Fei Fei, L. 3D Object Representations for Fine-Grained Categorization. In: *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia: [b.n.], 2013.

[12] Krizhevsky, A., Sutskever, I. a Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*. Január 2012, zv. 25. DOI: 10.1145/3065386.

[13] Lin, Y.-L., Morariu, V. I., Hsu, W. a Davis, L. S. Jointly Optimizing 3D Model Fitting and Fine-Grained Classification. In: Fleet, D., Pajdla, T., Schiele, B. a Tuytelaars, T., ed. *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, s. 466–480. ISBN 978-3-319-10593-2.

[14] Ma, Z., Chang, D., Xie, J., Ding, Y., Wen, S. et al. Fine-grained vehicle classification with channel max pooling modified CNNs. *IEEE Transactions on Vehicular Technology*. IEEE. 2019, zv. 68, č. 4, s. 3224–3233.

[15] Nash, W., Drummond, T. a Birbilis, N. A review of deep learning in the study of materials degradation. *Npj Materials Degradation*. December 2018, zv. 2. DOI: 10.1038/s41529-018-0058-x.

[16] Nayak, S. *Understanding AlexNet*. 2018. Dostupné z: https://www.learnopencv.com/understanding-alexnet/.

[17] Saha, S. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. *Medium* [online]. 2018. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[18] Simonyan, K. a Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv 1409.1556*. September 2014.

[19] Smeda, K. Understand the architecture of CNN. *Medium* [online]. 2019. Dostupné z: https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7.

[20] Sochor, J., Špaňhel, J. a Herout, A. BoxCars: Improving Fine-Grained Recognition of Vehicles Using 3-D Bounding Boxes in Traffic Surveillance. *IEEE Transactions on Intelligent Transportation Systems*. 2018, PP, č. 99, s. 1–12. DOI: 10.1109/TITS.2018.2799228. ISSN 1524-9050.

[21] Sochor, J., Herout, A. a Havel, J. BoxCars: 3D Boxes as CNN Input for Improved Fine-Grained Vehicle Recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[22] Sorokina, K. Image Classification with Convolutional Neural Networks. *Medium* [online]. 2017. Dostupné z: https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8.

[23] Šulc, M. a Matas, J. Fine-grained recognition of plants from images. *Plant Methods*. Springer. 2017, zv. 13, č. 1, s. 115.

[24] Szegedy, C., Ioffe, S. a Vanhoucke, V. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *CoRR*. 2016, abs/1602.07261. Dostupné z: http://arxiv.org/abs/1602.07261.

[25] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E. et al. Going Deeper with Convolutions. *CoRR*. 2014, abs/1409.4842. Dostupné z: http://arxiv.org/abs/1409.4842.

[26] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. a Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *CoRR*. 2015, abs/1512.00567. Dostupné z: http://arxiv.org/abs/1512.00567.

[27] Tafazzoli, F., Nishiyama, K. a Frigui, H. A Large and Diverse Dataset for Improved Vehicle Make and Model Recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. July 2017.

[28] Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J. et al. Building a Bird Recognition App and Large Scale Dataset With Citizen Scientists: The Fine Print in Fine-Grained Dataset Collection. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[29] Wang, Q., Wang, Z., Xiao, J., Xiao, J. a Li, W. Fine-Grained Vehicle Recognition in Traffic Surveillance. In: Chen, E., Gong, Y. a Tie, Y., ed. *Advances in Multimedia Information Processing - PCM 2016*. Cham: Springer International Publishing, 2016, s. 285–295. ISBN 978-3-319-48890-5.

[30] Wang, Y., Tan, X., Yang, Y., Liu, X., Ding, E. et al. 3D Pose Estimation for Fine-Grained Object Categories. In: Leal Taixé, L. a Roth, S., ed. *Computer Vision – ECCV 2018 Workshops*. Cham: Springer International Publishing, 2019, s. 619–632. ISBN 978-3-030-11009-3.

[31] Wei, X., Xie, C. a Wu, J. Mask-CNN: Localizing Parts and Selecting Descriptors for Fine-Grained Image Recognition. *CoRR*. 2016, abs/1605.06878. Dostupné z: http://arxiv.org/abs/1605.06878.

[32] Xiang, Y., Fu, Y. a Huang, H. Global Topology Constraint Network for Fine-Grained Vehicle Recognition. *IEEE Transactions on Intelligent Transportation Systems*. IEEE. 2019.

Príloha A

Obsah priloženého DVD

Priložené DVD obsahuje:

- dataset/ – priečinok obsahujúci moju časť dátovej sady
- lib/ – knižnica, k dátovej sade BoxCars116k
- saved_models/ – priečinok s natrénovanými modelmi
- src/ – priečinok so zdrojovými kódmi
- README.md – súbor s popisom inštalácie a použitia
- requirements.txt – súbor s potrebnými balíkmi k inštalácií
- poster.pdf – plagát mojej práce
- video_present.mp4 – video-prezentácia
- tex/ – priečinok so zdrojovými súbormi textovej časti práce
- bp.pdf – textová časť práce

Príloha B

Plagát

Obr. B.1: Plagát