

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ANONYMIZACE SPZ VOZIDEL

CAR LICENCE PLATE ANONYMIZATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. BARBORA SKŘIVÁNKOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MICHAELA DRAHOŠOVÁ

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů

Akademický rok 2015/2016

Zadání diplomové práce

Řešitel: **Skřivánková Barbora, Bc.**

Obor: Bioinformatika a biocomputing

Téma: **Anonymizace SPZ vozidel
Car Licence Plate Anonymization**

Kategorie: Umělá inteligence

Pokyny:

1. Prostudujte principy a metody používané pro detekci a anonymizaci SPZ vozidel zachycených na fotografiích a existující nástroje.
2. Analyzujte obrazový materiál ze služby Panorama dodaný společností Seznam.cz.
3. Navrhněte program umožňující provádět automatizovanou detekci a anonymizaci SPZ v obrazovém materiálu ze služby Panorama s využitím metody založené na hlubokých neuronových sítích.
4. Program z bodu 3 implementujte.
5. Ověřte funkčnost programu pro obrazový materiál ze služby Panorama.
6. Zhodnoťte dosažené výsledky a diskutujte přínos práce.

Literatura:

- Dle pokynů vedoucí práce.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

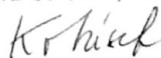
Vedoucí: **Šikulová Michaela, Ing., UPSY FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 25. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií
Ústav počítačových systémů a sítí
602 00 Brno, Božetěchova 2



doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

Abstrakt

Při procházení online map jsou dnes dostupné i souvislé snímky prohlížených míst. Pokud jsou však snímky pořizovány ve veřejném prostoru, nacházejí se na nich také kolemjdoucí lidé a kolemjdoucí vozidla. Cílem této práce je vytvoření automatizovaného systému pro anonymizaci SPZ vozidel ve fotografiích optimalizovaného pro službu Panorama společnosti Seznam.cz a.s.. Problém anonymizace SPZ vozidel je v této práci rozdělen do dvou fází. V první fázi jsou ze snímku pořízeného na veřejném prostranství vybrána místa, na kterých se nachází vozidla s SPZ, ve druhé fázi je potom ve vybrané části obrázku s vozidlem lokalizována konkrétní oblast, na které se SPZ vozidla nachází. Detekce automobilů je prováděna pomocí hluboké neuronové sítě. Následná lokalizace SPZ vozidel je zpracovávána pomocí dopředné plně propojené neuronové sítě. Cílem této práce je překonat problémy, na kterých selhávají aktuální komerční řešení. Jedná se zejména o častou falešně pozitivní detekci SPZ a velkou výpočetní náročnost. Výsledky této práce však nejsou uspokojivé, což mohlo být způsobeno použitou trénovací sadou pro neuronové sítě. Ačkoliv se tato sada dodaná společností Seznam.cz a.s. zdála býti dostatečně robustní, nakonec se ukázala jako nedostačující pro naučení složitější neuronové sítě.

Abstract

While browsing an online map server, continuous photographs of certain places can be browsed as well. When the map service takes pictures of a public space, there are some personal data captured as well (i.e. faces, car licence plates). The goal of this thesis is the design of automated car licence plates anonymization system, optimized for the Panorama service provided by the Seznam.cz a.s. corporation. In this thesis, the process of car licence plate anonymization is divided into two parts: the first one solves a detection of cars and the second solves a car licence plate localization in the selected image. The car detection is based on the deep neural network approach, the car licence plate localization is solved by using a fully connected neural network performing a regression task. The goal of this thesis is to get over the disadvantages of commercial solution used nowadays. These are false positive results and high computational complexity. Results of this thesis are not as good as expected. The reason could be a dataset provided by Seznam.cz a.s. corporation, which seemed to be robust enough in the beginning, but in the end it showed up to be not suffice enough to train the neural network.

Klíčová slova

Detekce, anonymizace, státní poznávací značka, registrační značka, neuronová síť, hluboká neuronová síť, zpracování osobních údajů.

Keywords

Detection, anonymization, car licence plate, neural networks, deep neural networks, personal data processing.

Citace

Barbora Skřivánková: Anonymizace SPZ vozidel, diplomová práce, Brno, FIT VUT v Brně, 2016

Anonymizace SPZ vozidel

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pod vedením Ing. Michaely Drahošové a uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala. Další informace mi poskytli Ing. Adam Kolář, Ing. Miroslav Talášek a Mgr. Jan Havlíček ze společnosti Seznam.cz.

.....
Barbora Skřivánková
26. května 2016

Poděkování

Na tomto místě bych ráda poděkovala svojí vedoucí, paní Ing. Michaely Drahošové, za trpělivé vedení a vstřícný přístup na konzultacích. Dále bych chtěla poděkovat panu Ing. Adamu Kolářovi za poskytnutí cenných rad a podpory při zpracovávání praktické části diplomové práce. Velký dík patří také celému týmu Mapy.cz v čele s Mgr. Janem Havlíčkem a Ing. Miroslavem Taláškem, který mi v průběhu vypracovávání praktické části práce poskytoval perfektní zázemí. A nakonec skupině Prodigy za skvělou motivační hudbu, která člověku nedovolí přestat.

© Barbora Skřivánková, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Osobní údaje a jejich ochrana	3
2.1	Základy ochrany osobních údajů v ČR	3
2.2	Anonymizace	4
2.3	Metody využívané v současnosti	4
2.3.1	Manuální anonymizace	6
2.3.2	Automatizovaná anonymizace	6
2.4	Obrazová data pro službu Panorama	7
2.4.1	Snímání fotografií	7
2.4.2	Ukládání fotografií	8
2.4.3	Sešívání fotografií	8
2.4.4	Anonymizace	9
2.4.5	Zobrazení uživateli	9
3	Neuronové sítě	10
3.1	Biologický základ neuronových sítí	10
3.2	Umělý neuron	11
3.3	Vícevrstvé neuronové sítě	14
3.3.1	Topologie sítí	15
3.3.2	Učení neuronových sítí	15
3.4	Konvoluční neuronové sítě	19
3.4.1	Vstupní vrstva	20
3.4.2	Konvoluční vrstva	20
3.4.3	Pooling vrstva	21
3.4.4	Konvoluční síť	22
3.5	Finetuning	23
4	Návrh	24
4.1	Předzpracování dat	25
4.2	Detekce automobilů	26
4.3	Lokalizace SPZ	27
4.4	Zatemnění SPZ	29
5	Implementace	30
5.1	Výsledný program	30
5.1.1	Předzpracování dat	30
5.1.2	Detekce automobilů	30

5.1.3	Lokalizace SPZ	32
5.1.4	Zatemnění SPZ	33
5.2	Pomocné skripty	33
5.2.1	Ruční klasifikace	33
5.2.2	Zpracování výsledků	33
6	Ověření funkčnosti	35
6.1	Data	35
6.2	Metoda testování	37
6.2.1	Spouštění	37
6.3	Dosažené výsledky	37
6.3.1	Detekce vozidel	37
6.3.2	Lokalizace SPZ	38
7	Závěr	43
A	Obsah CD	47
A.1	Technická zpráva	47
A.2	Praktická část	47
B	Spuštění programu	48
B.1	Závislosti	48
B.1.1	Caffe	48
B.1.2	ImageMagick	48
B.1.3	Keras	48
B.1.4	CUDA	48

Kapitola 1

Úvod

V průběhu vývoje internetových služeb se stále zvyšuje objem dat, která je možno uživateli v reálném čase předat. Výpočetní technika se dostala tak daleko, že je možné uživateli ukázat reálnou podobu míst, která jsou od něj vzdálena tisíce kilometrů. S nespornými výhodami plynoucími z vývoje tímto směrem však přicházejí i mnohé problémy. Jde převážně o problémy rázu prozrazení osobních údajů, kterým je třeba se jak z legislativních, tak z etických důvodů vyvarovat. V současné době se tato problematika řeší pomocí anonymizace zachycených dat. Právě tomuto tématu se věnuje tato diplomová práce - anonymizaci SPZ vozidel zachycených na snímcích veřejného prostoru.

V současné době máme díky neustálému zlepšování hardwaru k dispozici obrovský výpočetní výkon. Díky tomuto výkonu si můžeme troufnout i do řešení tak obtížné úlohy jako je učení neuronových sítí na úloze automatizované detekce a anonymizace obrazových dat.

Tato práce je členěna do sedmi kapitol. Po této úvodní kapitole následuje kapitola 2, která se v první polovině – 2.1 – zabývá platnou legislativou definující, jakým způsobem jsou osobní údaje v České republice definovány a jak je třeba s nimi nakládat. Ve druhé polovině této kapitoly – 2.2 – je potom rozebrána anonymizace nejen obrazových dat a její metody využívané v současnosti.

V následující kapitole 3 je popsána návaznost umělých neuronových sítí na biologický neuron, dále pak neuronové sítě a jejich použití v praxi.

V kapitole 4 je na základě předchozích kapitol navržen program, který bude řešit automatizovanou anonymizaci obrazových dat s využitím neuronových sítí a detektoru hran.

Následující kapitola 5 popisuje technologie, které byly využity k implementaci programu, který automatickou anonymizaci zprostředkovává.

V kapitole 6 jsou potom popsány výsledky, kterých bylo s implementovaným programem dosaženo a optimalizace, které v rámci řešení této práce byly nasazeny.

V kapitole 7 jsou potom shrnuty výsledky této diplomové práce a nastíněno její možné pokračování do budoucna.

Kapitola 2

Osobní údaje a jejich ochrana

Anonymizace dat je v dnešní době velmi aktuálním tématem. Intenzivně se na ní pracuje jak na úrovni datové, tak na úrovni anonymizace obrazových dat, která je také předmětem této práce. Obchodní společnost, která je schopna zajistit kvalitní anonymizaci dat, která spravuje, tím nejen plní svoje zákonné povinnosti, může si tím také vylepšit svoje jméno v očích zákazníků a společností, se kterými spolupracuje.

Tato kapitola se blíže zabývá legislativním podtextem ochrany osobních údajů v části 2.1, dále potom v části 2.2 popisem pojmu anonymizace dat, v části 2.3 jsou popsány metody, které se k anonymizaci dat používají v současnosti a nakonec v části 2.4 je dokumentován postup snímání dat pro službu Panorama společnosti Seznam.cz.

2.1 Základy ochrany osobních údajů v ČR

Ochrana osobních údajů v České republice představuje soubor práv a povinností, které se vztahují k uchování a zacházení s daty přímo se týkajícími dané fyzické osoby. Osobní údaje nemusí být vždy přímo identifikující danou osobu (jako je například rodné číslo, jméno a příjmení, adresa trvalého bydliště apod.). Za osobní údaje jsou považovány jakékoliv údaje přímo související se životem konkrétní nebo určitelné osoby. Může jít například o provozování koníčků, členství v politických stranách, účast na specifických událostech, návštěvy různých míst, velikost oblečení a podobně. Ochrana osobních údajů (ochrana osobních dat) je v České republice regulována zákonem č. 101/2001 Sb. [2], o ochraně osobních údajů a dalšími právními předpisy.

Zákon rozlišuje osoby, které zpracovávají osobní údaje (*správce osobních údajů*) a osoby, jejichž data správci zpracovávají (*subjekt osobních údajů*). Správcům jsou zákonem ukládány především povinnosti, zatímco subjektům jsou dána práva. Dodržování práv a povinností obou stran je v České republice kontrolováno prostřednictvím Úřadu na ochranu osobních údajů.

Základní ideou ochrany osobních údajů je to, že osobní údaje jsou shromažďovány a zpracovávány za nějakým účelem. Každý účel vyžaduje jiný typ osobních údajů a také jiné množství takto zpracovávaných osobních údajů. Ve výše uvedeném zákoně byly pro bližší specifikaci těchto účelů zavedeny pojmy *účel zpracování*, *prostředky zpracování*, *způsob zpracování*, *kategorie osobních údajů*, *kategorie subjektů údajů* a *kategorie příjemců*. V těchto termínech je třeba zpracování osobních údajů vždy předem popsat.¹

¹Výklad zákona 101/2001 Sb. dle <http://www.oou.cz>

V této práci se budu věnovat zpracovávání obrazového materiálu nasbíraného na veřejném prostranství. Vzhledem k tomu, že veřejné prostranství je vždy snímáno za běžného dne, spolu s terénem jsou nasnímaní i lidé a další objekty. Jak bylo řečeno dříve v této kapitole, veškeré informace, které mohou prozradit jakékoliv údaje přímo související s životem dané osoby jsou považovány za osobní údaje. Z toho vyplývá, že i obrazový materiál nasnímaný na veřejném místě obsahující cizí osoby nebo automobily obsahuje osobní údaje a je třeba se věnovat způsobu jejich zpracování.

V dnešní digitální době máme k dispozici velké množství obrazových dat. V této práci se budu věnovat datům zachytávaným pro účely vizualizace prostoru z mapy nasbírané v rámci služby *Panorama* společnosti *Seznam.cz*. Tato data jsou sbírána prostřednictvím speciálně upravené panoramatické kamery, která je umístěna na střeše vozidla. Toto vozidlo při průjezdu zpracovávanou lokalitou snímá svoje okolí a pořízený obrazový materiál se poté zpřístupňuje široké veřejnosti. Publikování obrazového materiálu, na kterém se vyskytují osobní údaje jako jsou státní poznávací značky (SPZ)² vozidel, obličejů kolemjdoucích a podobně je velmi citlivou záležitostí. Aby bylo minimalizováno riziko zneužití těchto údajů, byla zvolena možnost anonymizace.

2.2 Anonymizace

Anonymizovaná data jsou taková data, která nemohou vést přímo k identifikaci konkrétního člověka. Pokud jsou data obsahující osobní údaje anonymizována, přestávají podléhat přísným regulacím Zákona o ochraně osobních údajů a lze s nimi nakládat volněji.

Hranici, kdy jsou data už anonymizována nelze zcela jednoznačně definovat a anonymizaci je třeba pro každou specifickou skupinu dat posuzovat individuálně. Pouhé začernění jména v dokumentech malé firmy například nemusí vést k tomu, že konkrétní osoba nebude dle dalších vodítek v dokumentech identifikovatelná. Příklad toho, že stejná úroveň anonymizace nemusí mít vždy stejné výsledky lze sledovat porovnáním obrázků 2.1 a 2.2. Oba obrázky jsou zpracovány stejným rozostřovacím filtrem. U SPZ na obrázku 2.1 by bylo dekodování původní poznávací značky automobilu obtížné. U fotografie 2.2 by však v případě dodatečné informace, že se jedná například o některého ze zaměstnanců konkrétního středního podniku byla identifikace s velkou pravděpodobností možná.

2.3 Metody využívané v současnosti

Datová anonymizace dosáhla v současné době uspokojivé úrovně. Úspěšně anonymizována jsou jak data v pdf dokumentech, tak v databázích. Datovou anonymizaci lze v dnešní době provádět plně automatizovaně prostřednictvím mnoha nástrojů jak komerčních, tak volně dostupných.

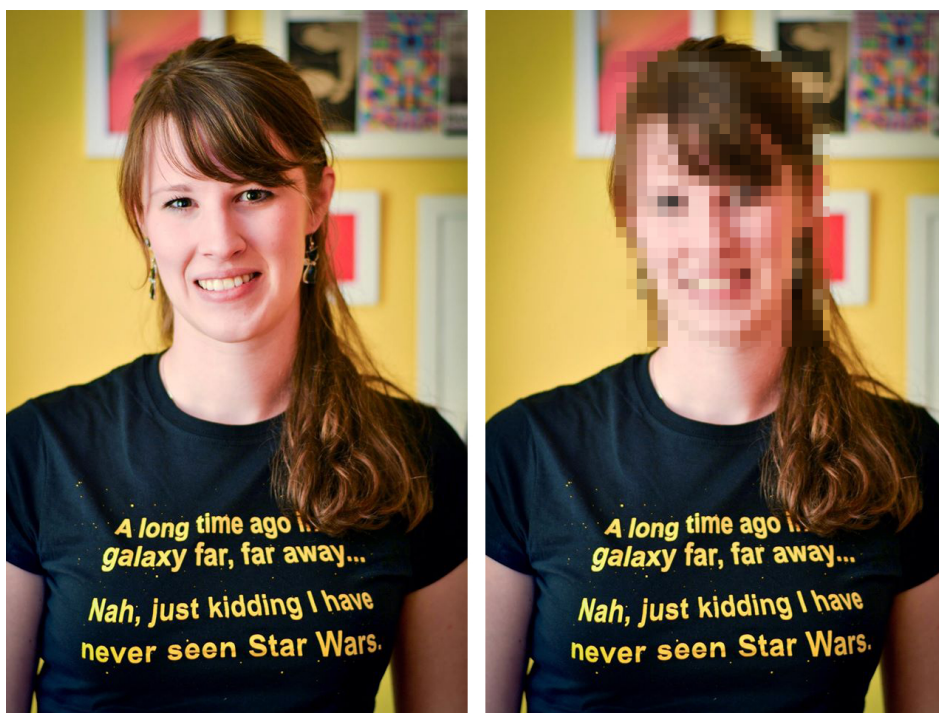
Anonymizace v obrazových datech zatím nedosáhla tak vysoké úrovně a to hned z několika důvodů:

1. Anonymizace v obrazových datech je velmi aplikačně specifická - není proto tak velký tlak na její vývoj.
2. Anonymizace obrazových dat je výpočetně náročnější než anonymizace na úrovni textové.

²Též registrační značky (RZ). Pojmy jsou dle Předpisu č. 56/2001 Sb. §4 ekvivalentní.



Obrázek 2.1: Anonymizovaná SPZ vozidla.



Obrázek 2.2: Anonymizovaný obličej.



Obrázek 2.3: Problémy stávajícího řešení anonymizace SPZ v Panoramě. Chybně anonymizované nápisy nad obchody jsou na obrázku vyznačeny červenou barvou³.

Anonymizace obrazových dat sestává ze dvou částí: *detekce* části obrázku určené k anonymizaci a vlastního procesu *anonymizace*. Je zřejmé, že proces anonymizace už není výpočetně nijak náročný. Jedná se zpravidla o aplikaci rozostření, případně o přidání vrstvy, která plně překrývá anonymizované části obrázku. Problematická však je detekce části obrázku určené k anonymizaci.

2.3.1 Manuální anonymizace

Po dlouhou dobu byla anonymizace obrazových dat potřeba většinou na malém objemu dat (například ve zpravodajství a podobně). V takovém případě je naprosto dostačující provádět výběr části obrázku, která je určena k anonymizaci, manuálně.

Kvalita anonymizace je velmi dobrá, protože člověk přímo vidí výsledek a dokáže korigovat rozsah provedené úpravy tak, aby výsledný obrázek byl skutečně anonymní. Na svoje limity tato metoda však naráží při nárůstu množství anonymizovaných dat.

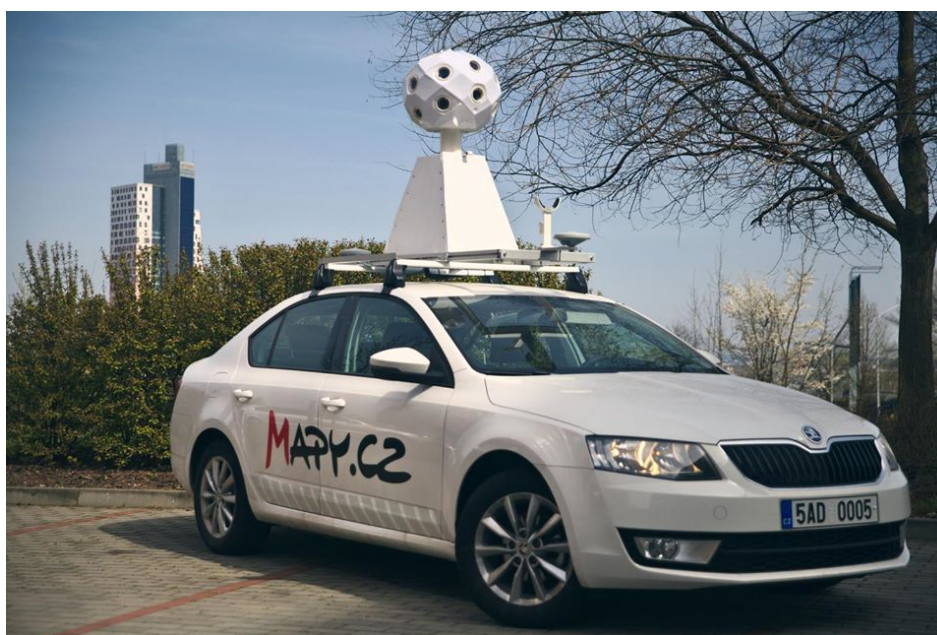
Pro představu, při zpracovávání dat pro službu Panorama vzniknou každý snímání den 4TB obrazového materiálu z každého snímajícího vozidla. Takové množství dat už není v rozumném čase možné zpracovávat manuálně a tak je třeba vytvořit automatizovanou variantu.

2.3.2 Automatizovaná anonymizace

Automatizovaná anonymizace obrazových dat je zatím na počátku svého vývoje. Zásadním problémem automatizované anonymizace, hlavně při anonymizaci SPZ vozidel, jsou chyby vzniklé při detekci SPZ. Ty jsou totiž často detekovány i na místech, na kterých se sice na fotografii vyskytuje text, nikoliv však SPZ, viz obrázek 2.3.

Při použití anonymizovaných dat pro službu Panorama je tento druh chyb nepřijatelný. Jde totiž o omezení jednoho z důležitých účelů služby Panorama. Uživatelům slouží hlavně pro lepší orientaci v terénu, který neznají. Pokud chce uživatel službu využít k rychlejšímu nalezení nějaké prodejny, je naprosto nežádoucí, aby v rámci automatické anonymizace SPZ vozidel bylo anonymizováno označení nad vstupem této prodejny. V aktuálním stavu automatizované anonymizace je právě tato závada velmi častá. V této práci bude ukázáno, jakým způsobem se těmito chybám vyvarovat.

³Obrázek pochází z automatizovaného anonymizéru, který je aktuálně používán ve společnosti Seznam.cz.



Obrázek 2.4: Automobil s 18 kamerami umístěnými na snímací hlavě na střeše⁴.

2.4 Obrazová data pro službu Panorama

Obrazová data pro tuto práci byla dodána společností Seznam.cz a.s. Většina informací pro tuto část práce byla získána v rámci několika konzultací s jejími zaměstnanci, Ing. Adamem Kolářem a Ing. Miroslavem Taláškem. Ze společnosti Seznam.cz byla dále poskytnuta neveřejná prezentace o snímání obrazových dat pro službu Panorama, ze které bylo při přípravě tohoto textu také vycházeno [13].

V této sekci bude popsán způsob, jakým jsou obrazová data pro službu Panorama snímána. Dále bude popsáno, jak jsou data dále zpracovávána, než se dostanou ke koncovému uživateli. Anonymizér, který je praktickým výstupem této práce je jen jedním článkem v sekvenci operací, která je nad každým zachyceným obrázkem provedena.

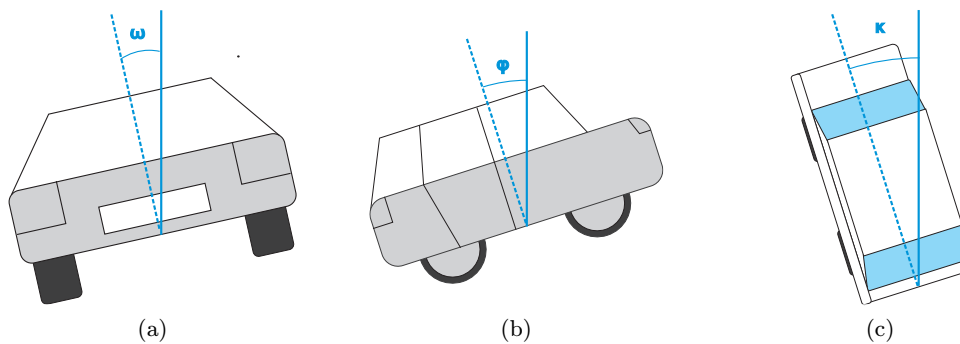
2.4.1 Snímání fotografií

Fotografie jsou snímány speciálně upraveným vozem s 18 kamerami rozmístěnými na střeše. Tyto kamery zabírají 360° okolí snímacího automobilu. Způsob, jakým jsou kamery na střeše automobilu umístěny, je znázorněn na obrázku 2.4. Kamery jsou na snímací hlavě umístěny ve dvou řadách. První snímá dolní část výsledné fotografie - v té je umístěno 8 kamer. V druhé řadě jsou umístěny kamery snímající střední a horní část fotografie. Na vrcholu záchytného zařízení pro kamery je umístěna ještě jedna kamera mířící přímo vzhůru.

Při snímání se auto může pohybovat rychlostí až 70 km/h. Snímání se provádí na všech 18 kamerách současně každých 5 metrů. Aby bylo možné později fotografii co nejvíce přiblížit tomu, jak prostředí vnímá člověk, který se v něm pohybuje, při každém snímku je zaznamenána ještě sada metadat. Ta zahrnují:

- Číslo snímku,

⁴Fotografie převzatá z Facebooku Mapy.cz.



Obrázek 2.5: Zaznamenávané úhly při snímání fotomateriálu.

- celek (sada 18 fotografií), do kterého snímek patří a poloha v něm,
- datum snímání,
- zeměpisná poloha snímacího auta v souřadnicích UTM 33,
- ω – úhel náklonu automobilu v pravolevém směru (viz 2.5a),
- φ – úhel náklonu automobilu v předozadním směru (viz 2.5b),
- κ – odklon směru jízdy automobilu od severního směru, dále označováno pojmem *heading* (2.5c).

Pro výpočet jednotlivých úhlů jsou k dispozici speciální senzory. Pro zaznamenávání *headingu* jsou na autě dvě vysoce přesné GPS jednotky, které na základě rozdílu svých GPS poloh dokáží přesně určit směr, kterým se snímací automobil pohybuje. Náklony ve všech směrech jsou měřeny pomocí *inerciální jednotky a odometru*.

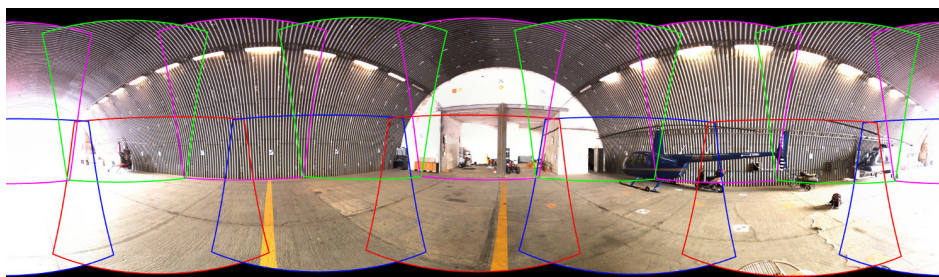
2.4.2 Ukládání fotografií

Při snímání každých 5 metrů v rychlosti 70 km/h musí kamery pořizovat fotografie v plném rozlišení v rychlosti 3,88 fps. V tomto okamžiku tedy vzniká problém obrovského množství dat, která je potřeba okamžitě zpracovávat. Ještě předtím, než jsou fotografie ve dvou kopiích uloženy na jeden ze čtyř 800GB SSD disků je třeba je převést z formátu *raw*, do kterého jsou pořizovány, do formátu *jpeg*, ve kterém jsou ukládány a dále zpracovávány. Tento převod je prováděn na výkonné grafické kartě umístěné v zavazadlovém prostoru automobilu pomocí technologie *CUDA*.

Veškeré operace nad daty jsou prováděny v zavazadlovém prostoru snímacího automobilu, kam data ze střechy putují přes vysokorychlostní datové kabely. Každá z 18 kamer je s počítačem spojena pomocí svojí vlastní gigabitové linky.

2.4.3 Sešívání fotografií

Jakmile se data dostanou z terénu, začíná rozsáhlá množina operací, která je nad nimi prováděna. Jako první přijde na řadu sešívání snímků (znázorněno na obrázku 2.6). Pomocí nástroje navrženého ve společnosti Seznam.cz je z 18 fotografií zachycených v jednom okamžiku vytvořena jedinná širokoúhlá fotografie, nad kterou jsou prováděny další operace. V rámci sešívání fotografií jsou také odstraněny části snímacího automobilu, které se dostaly do záběru.



Obrázek 2.6: Sešívání 17 snímků s překryvem do jedné panoramatické fotografie. Poslední snímek z horní kamery slouží k doplnění volných míst v horní části fotografie.⁵

2.4.4 Anonymizace

Anonymizace, která je tématem této práce, probíhá nad materiálem, který je již téměř připraven k zobrazení uživateli. Anonymizovány nejsou jen SPZ vozidel, o kterých je v této práci řeč, ale také obličeje chodců a řidičů zachycených na snímcích.

Aktuálně je pro anonymizaci využíváno komerčního řešení vyvinutého na míru pro společnost Seznam.cz.

2.4.5 Zobrazení uživateli

Výsledný obrázek je následně namapován na kouli, na které už je zobrazován koncovému uživateli. Aby byla přesně simulována situace při snímání prostředí, pomocí ekvidistantní projekce (viz dále) jsou vypočteny souřadnice na vnitřním povrchu koule, v jejímž středu je situován uživatel.

Natočení této koule v prostoru potom dokáže dorovnat zakřivení, které vznikne vlivem snímání pod úhlem a sešívání obrazu. V tomto okamžiku jsou využity zaznamenané úhly ω a φ , jejichž aplikace zamezí uživatelskému pocitu tzv. *padání obrazu*.

Před zobrazením uživateli je fotografie rozdělena na menší podčásti o rozměru $512 \times 512px$, které jsou ze serveru stahovány až na vyžádání. Tímto se dosahuje vyšší rychlosti zobrazování služby Panorama koncovému uživateli, protože konkrétní části snímku se stahují až v okamžiku, kdy se dostanou do zorného pole uživatele a tedy nikdy nejsou stahovány části fotografie, které nebudou zobrazeny.

Ekvirektangulární projekce

Pro namapování obdélníkového obrazu na kouli je využita ekvirektangulární projekce (v literatuře též označována jako ekvidistantní projekce) [11]. Tato projekce využívá přepočtení souřadnic dle vztahů:

$$x = \lambda * \cos(\varphi_1), \quad (2.1)$$

$$y = \varphi, \quad (2.2)$$

kde x a y jsou souřadnice na fotografii v rovině, λ značí zeměpisnou délku od centrálního poledníku v daném zobrazení, φ značí zeměpisnou šířku a φ_1 je parametr rovnoběžek, který určuje rozsah skutečné projekce.

⁵Fotografie převzatá z archivu firmy Seznam.cz

Kapitola 3

Neuronové sítě

Neuronové sítě jsou jednou ze tří hlavních podskupin vědního oboru soft computing [10]. Dalšími problematikami, které tato vědní disciplína zkoumá jsou fuzzy systémy a evoluční algoritmy. Tato práce se věnuje hlubokým neuronovým sítím pracujícím nad obrázkovými daty. Neuronové sítě jsou uvedeny do širšího kontextu v této kapitole, která je rozdělena do několika částí.

V části 3.1 je zjednodušeně popsán biologický neuron, na jehož funkci je celá teorie neuronových sítí založena. V sekci 3.2 je potom popsán způsob, jakým je v umělé inteligenci funkčnost biologického neuronu napodobována. Sekce 3.3 shrnuje postupy, kterými jsou umělé neurony spojovány do neuronových sítí a jakých výhod je tím dosaženo. V této části je důležitá podkapitola 3.3.2, kde je podrobně popsán jeden ze základních algoritmů pro učení neuronových sítí: *backpropagation*. V následující části 3.4 jsou popsány složitější koncepty hlubokých neuronových sítí, které budou využity v této diplomové práci. Nakonec je v krátké kapitole 3.5 shrnuta metoda finetuningu, který v této práci hraje významnou roli.

3.1 Biologický základ neuronových sítí

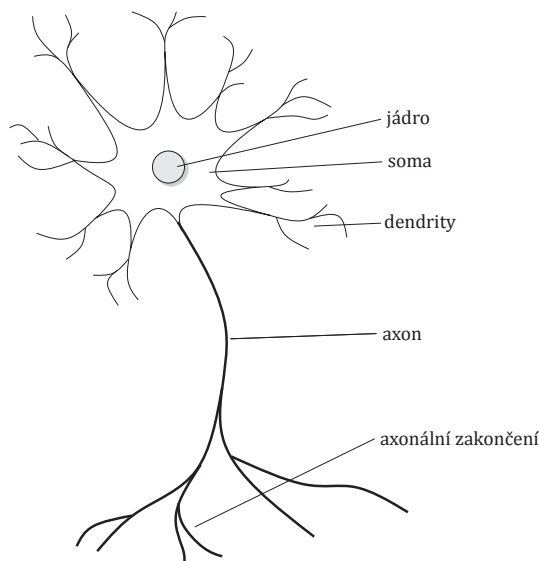
Při studiích v oborech biologie a biofyziky bylo během posledních několika dekad odhaleno mnoho principů, díky kterým se můžeme přiblížit pochopení funkce naší nervové soustavy a mozku. Základní stavební jednotkou lidského těla je buňka, základní stavební jednotkou nervové soustavy potom je *neuron*. Těchto neuronů se v lidské nervové soustavě nachází cca 10^{11} [4]. Hlavní součásti neuronu jsou znázorněny na obrázku 3.1 a jsou to tyto:

Dendrit – Vstupní bod neuronu. Přes dendrity se do těla neuronu dostávají elektricko-chemické vzruchy, jejichž přenos je hlavní činností neuronu.

Axon – Část neuronu, která zajišťuje přenos vzruchů. Je ukončena *axonálním zakončením*, které je výstupním bodem neuronu a nachází se velmi blízko dendritu jiného neuronu.

Synapse – Spojení mezi dvěma neurony - přesněji mezi dendritem a axonálním zakončením. V dnešní době není aktivita mozku měřena v počtu neuronů, nýbrž právě v počtu synapsí.

Neurotransmitter – chemická látka, která zajišťuje přenos informací v mozku. V této práci budu mluvit jen o přenosu elektricko-chemického vzruchu v synapsi, ale neurotransmitery mají v nervové soustavě mnohem širší funkci.



Obrázek 3.1: Schéma biologického neuronu.

Soma – tělo buňky. Sleduje, zda-li suma velikosti nervových vzruchů přicházejících z jednotlivých dendritů nepřesáhla daný práh a pokud se tak stane, vygeneruje nový vzruch.

V okamžiku, kdy se v těle neuronu nashromáždí dostatečně velký potenciál, začne se šířit nervovým vláknem (axonem) až k axonálnímu zakončení. Tady přejde do dalšího neuronu. Velikost signálu přenášená z dendritu do těla neuronu závisí na síle impulsu sousedního neuronu, kterého se dendrit dotýká a také na *synaptické váze*. V těle neuronu se sečtou hodnoty signálů od všech aktivních dendritů. Pokud součet hodnot překročí jistý práh, soma vygeneruje nový impuls a tím se vzruch šíří do dalších neuronů. Bezprostředně po vytvoření pulzu se neuron stane na chvíli necitlivým. To má za následek generování nespojitého signálu, jehož frekvence se může pohybovat v rozmezí 0,1-100 Hz.

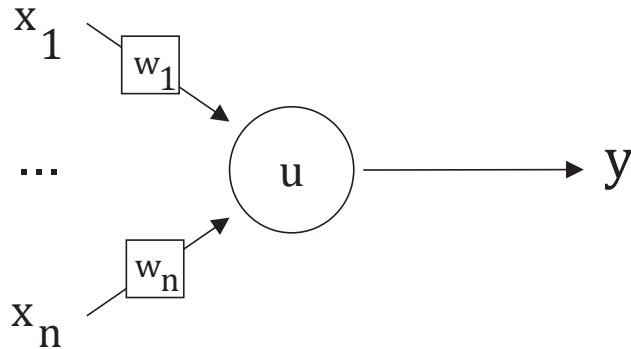
3.2 Umělý neuron

Princip biologického neuronu je napodobován umělým neuronem. Základní model umělého neuronu je zobrazen na obrázku 3.2, kde

x_i je vstupní hodnota, která může být napojena na primární vstup nebo na výstup jiného neuronu,

w_i je váha propojení. Určuje jak moc bude konkrétní vstup i ovlivňovat výstup neuronu. Váhy se v průběhu výpočtu mění - probíhá *učení*, případně přes neměnnou váhu může být připojen tzv. *bias*. Ten není primárním vstupem ani výstupem jiného neuronu - jedná se o konstantu vždy vstupující do neuronu,

u je hodnota vnitřního potenciálu neuronu.



Obrázek 3.2: Obecné schéma umělého neuronu, kde $\vec{x} = (x_1, x_2, \dots, x_n)$ je vstupní vektor, u je vnitřní potenciál a y je výstupní hodnota.

Jak je vidět na obrázku 3.2, na vstup umělého neuronu se přivádí vektor čísel, ty jsou pronásobena vektorem vah, které mají zpravidla malé hodnoty (většinou v intervalu $\langle -1; 1 \rangle$, případně $\langle -2; 2 \rangle$), dále jsou nad nimi provedeny básová a aktivační funkce (viz dále) a výstupem je jediné číslo. Umělý neuron je tedy matematická struktura, která zpracovává čísla a na výstupu podává jiná čísla [5].

Bázová funkce - určuje způsob výpočtu vnitřního potenciálu u neuronu. Nejčastěji využívána je lineární básová funkce [15]. Její vztah pro výpočet vnitřního potenciálu je následující:

$$u = \sum_{i=1}^n w_i \times x_i. \quad (3.1)$$

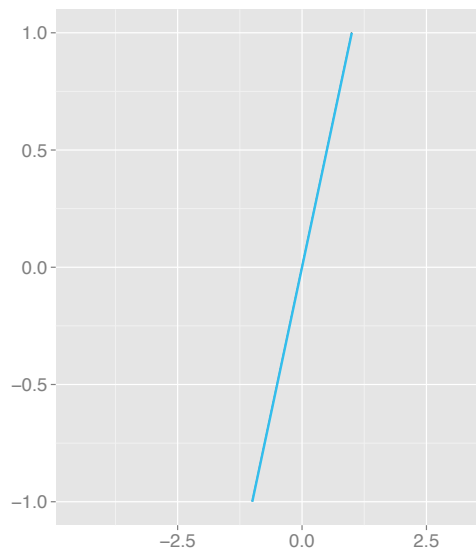
Další z využívaných básových funkcí je radiální básová funkce:

$$u = \sqrt{\sum_{i=1}^n (w_i \times x_i)^2}. \quad (3.2)$$

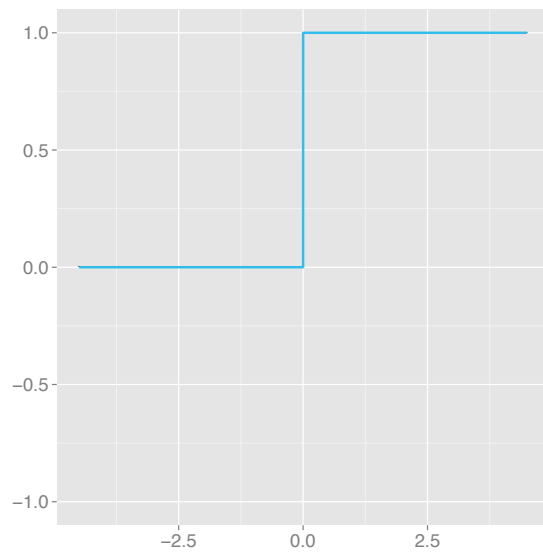
Aktivační funkce - též nazývaná *přenosová funkce*, využívá se pro ni symbol σ . Určuje způsob výpočtu výstupu neuronu z jeho vnitřního potenciálu. Čtyři nejčastěji používané aktivační funkce jsou znázorněny na obrázku 3.3. Na ose x je vždy hodnota vnitřního potenciálu neuronu u , na ose y je potom výstupní hodnota neuronu y .

Jednoduchý umělý neuron s jedním biasem a skokovou aktivační funkcí je v literatuře označován jako *perceptron*. Perceptron byl poprvé představen Frankem Rosenblattem v roce 1958, matematický model neuronu však byl známý už od roku 1943 a princip činnosti biologického neuronu dokonce už od konce 19. století.

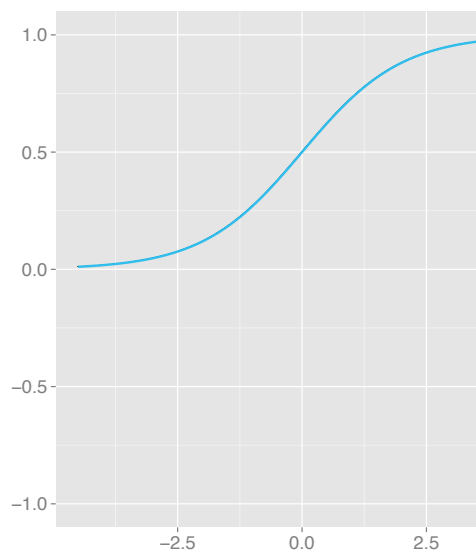
Pokud bychom chtěli perceptron využít ke klasifikaci vzorků do tříd, narazíme na jeho základní problém. Dokáže klasifikovat vzorky příslušící pouze dvěma třídám a tyto třídy musí být *lineárně separovatelné*. Váhy, které se upravují v průběhu učení perceptronu lze matematicky převést na vztah pro přímku v n -rozměrném prostoru. V rovině lze tedy lineární separovatelnost ilustrovat rozdělením na dvě poloroviny, kdy v jedné se nacházejí vzorky jedné třídy a ve druhé se nacházejí vzorky třídy druhé. Rozdíl mezi lineárně separovatelnými a lineárně neseparovatelnými problémy je znázorněn na obrázku 3.4. Mezi



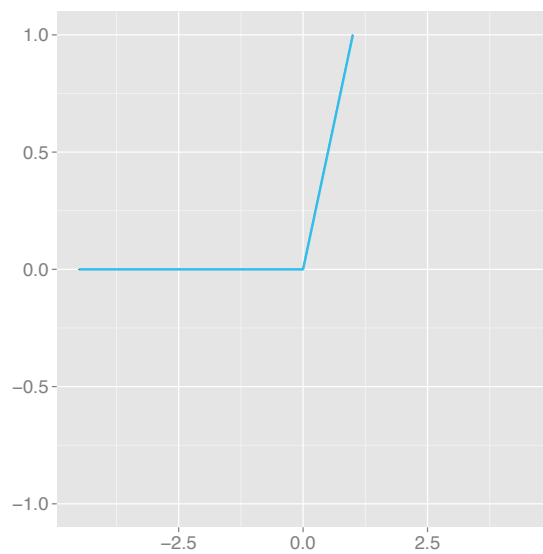
(a) Lineární funkce.



(b) Skoková funkce.

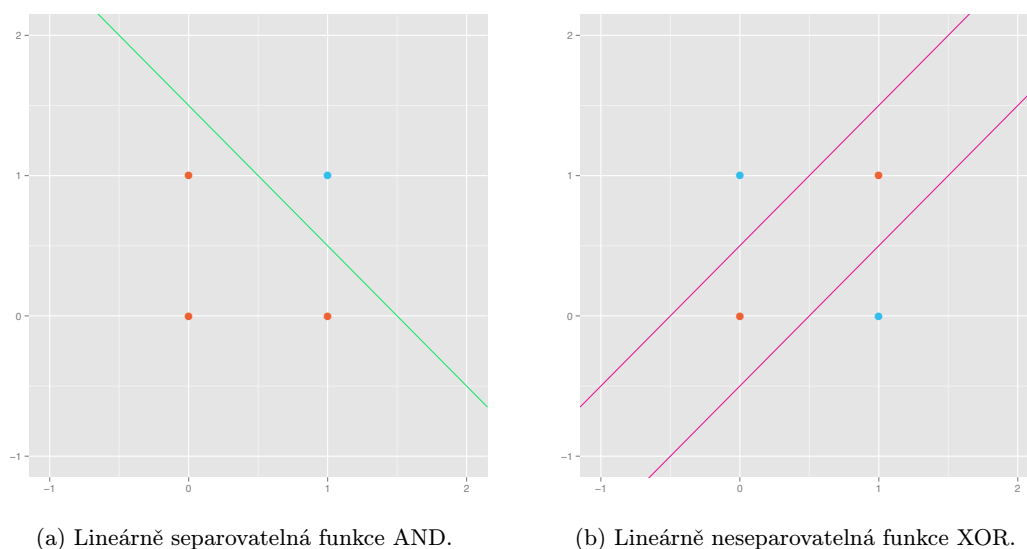


(c) Sigmoida.



(d) Lineární usměrňovač.

Obrázek 3.3: Nejčastěji využívané aktivační funkce.



Obrázek 3.4: Problém lineární separovatelnosti.

lineárně neseparovatelné problémy se řadí už i tak jednoduchý problém jako je XOR dvou hodnot (viz 3.4b). To značně omezuje obecnou využitelnost perceptronu a vzniká tak potřeba sdružování neuronů do neuronových sítí.

3.3 Vícevrstvé neuronové sítě

Neuronová síť je výpočetním systémem, o kterém může být uvažováno jako o *černé skříňce*, která přijímá vstupy a produkuje výstupy. Neuronová síť se skládá z několika vrstev umělých neuronů a hlavní zlepšení, které neuronové sítě přinášejí oproti samotným neuronům je možnost klasifikace lineárně neseparovatelných problémů.

Neuronové sítě jsou v praxi často nasazovány na některé z komplikovaných problémů [3], jako například:

Klasifikace - na vstup je přiložen vektor a na výstupu je získáno zařazení daného vektoru do některé z předem definovaných tříd.

Rozpoznávání vzorů - na vstup je přiložen vektor a na výstupu je vygenerován vzor, který nejlépe popisuje vektor poskytnutý na vstupu.

Porovnávání vzorů - na vstup je přiložen vektor znázorňující část hledaného vzoru. Na výstup jsou potom dotvořeny chybějící části vzoru.

Odstraňování šumu - na vstup je přiložen zašuměný obrázek a na výstupu je získán obrázek s odstraněným šumem.

Optimalizace - na vstup je přiložen systém, který správně řeší danou optimalizační úlohu a na výstupu je získáno nové funkční řešení, které přineslo zlepšení v některém ze sledovaných parametrů (např. rychlost, spotřeba energie apod.).

Řízení - na vstup je přiložen aktuální stav řízení a stav, do kterého chceme, aby se řízení dostalo a na výstupu získáme sekvence kroků, která tento přechod zajistí.

Simulace - na vstup je přiložen aktuální stav, případně i sekvence předchozích stavů řízení a na výstupu postupně získáváme stavy, které odpovídají vývoji reálného systému z těchto počátečních podmínek v čase.

3.3.1 Topologie sítí

Aby bylo možné pomocí neuronových sítí řešit i lineárně neseparovatelné problémy, je třeba podstatně zesložitit jejich architekturu. To se provádí vkládáním dalších vrstev neuronů do sítí. Důležitým faktorem přitom je, aby jednotlivé vrstvy neuronové sítě měly nelineární aktivační funkci. Pokud by totiž všechny vrstvy měly lineární aktivační funkci, složením těchto vrstev by pouze vznikla jiná lineární funkce a řešení neuronové sítě by tedy stále bylo ekvivalentem hledání rovnice přímky, která prostor řešení rozdělí na dva podprostory.

Dalším z důležitých faktorů při návrhu neuronových sítí jsou jednotlivá propojení. V průběhu vývoje se experimentovalo s mnohými nastaveními propojení. Některá používaná propojení jsou zobrazena na obrázku 3.5. První experimenty proběhly nad plně propojenou sítí, ve které je každý neuron propojen se všemi neurony svojí vrstvy, vrstev předchozích i následujících. Ta se postupně vyvinula ve vrstvou síť (viz 3.5a). Ve vrstvou síti lze výstup každého neuronu využít jako vstup některého z neuronů ve stejné vrstvě, případně jako vstup kteréhokoliv neuronu v některé z vrstev následujících. Odstraněním zpětných vazeb v rámci jednotlivých vrstev vzniká acyklická síť (viz 3.5b). V acyklické síti může být výstup každého neuronu připojen na vstup kteréhokoliv neuronu v některé z následujících vrstev. V této práci se však budu věnovat výhradně výzkumu na dopředných sítích. Dopředná síť (viz 3.5c) vzniká omezením propojení acyklických sítí. V dopředné síti může být výstup každého neuronu připojen pouze na vstup kteréhokoliv neuronu v logicky následující vrstvě neuronů [8].

Vícevrstvé sítě

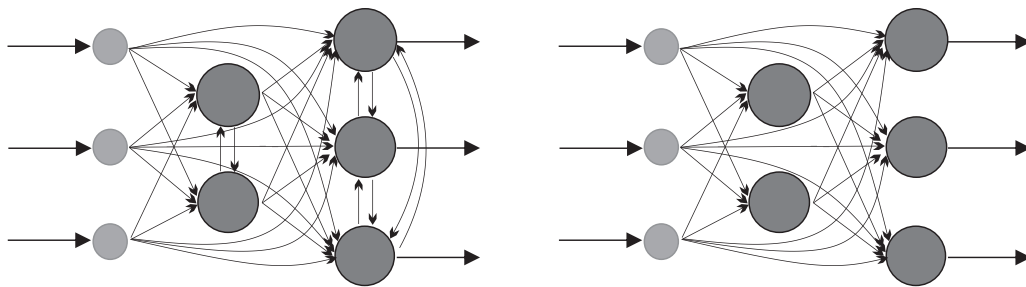
Vícevrstvá neuronová síť má více než dvě vrstvy neuronů. První vrstva se nazývá *vstupní vrstva* a nad načtenými vstupy neprovádí žádnou operaci. Do celkového počtu vrstev se vstupní vrstva v literatuře většinou nezapočítává. Poslední vrstva se nazývá *výstupní vrstva*. Tato vrstva se do celkového počtu vrstev započítává, protože pracuje jako obyčejná vrstva neuronové sítě. Až její výstupy jsou namapovány přímo na primární výstupy celé neuronové sítě. Všechny další vrstvy mezi vstupní a výstupní vrstvou se nazývají *skryté vrstvy* a počítají se do celkového počtu vrstev.

V dopředné neuronové síti je výstup každého neuronu každé vrstvy připojen na vstup všech neuronů vrstvy následující. V každém z neuronů další vrstvy však může mít výstup aktuálního neuronu jinou důležitost. Toto se nastavuje pomocí vah jednotlivých propojení. Pokud chceme některé propojení úplně eliminovat, jednoduše mu nastavíme váhu 0 a daná hodnota nebude v následujícím neuronu vůbec uvažována.

Váhy jednotlivých propojení se pro přehlednost zapisují pomocí matic. V každé neuronové síti existuje jedna váhová matice pro každou dvojici za sebou následujících vrstev. Zápis jednotlivých vah je blíže popsán v části 3.3.2.

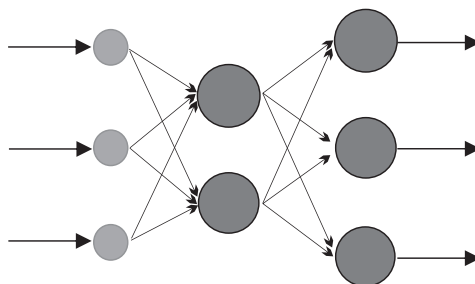
3.3.2 Učení neuronových sítí

Nejdůležitější vlastností neuronových sítí je pochopitelně schopnost najít řešení daného problému. Toto hledání řešení - *učení* - probíhá prostřednictvím změny vah jednotlivých



(a) Vrstvová síť.

(b) Acyklická síť.



(c) Dopředná síť.

Obrázek 3.5: Vícevrstvé neuronové sítě.

propojení tak, aby neuronová síť podávala co nejpřesnější výsledky pro daný problém. Nejobecnějším dělením algoritmů pro učení je na učení s učitelem a bez učitele [3].

Učení s učitelem je proces, který využívá vnějšího učitele nebo globálně známou informaci. V každém cyklu učení je vypočtena aktuální odchylka výstupu sítě od očekávaného výstupu, která se potom minimalizuje. Hlavní problémy, kolem kterých se diskuze o učení s učitelem ubírá jsou kdy učení ukončit, jak často a jak dlouho využívat k učení jeden vzorek vstupu a jak využívat informaci o vývoji učení a změně chyby. Učení s učitelem lze dále rozdělit na *strukturální učení s učitelem*, kde hodnota výstupu sítě závisí pouze na aktuálních hodnotách na vstupu, zatímco u *temporálního učení s učitelem* aktuální hodnota výstupu závisí i na vstupech a výstupech z předchozích cyklů učení.

Učení bez učitele je též nazýváno *samo-organizace*. Nebere v potaz žádného vnějšího učitele a pracuje pouze s informacemi známými během učení. Algoritmy učení bez učitele se snaží zorganizovat trénovací data a nalézt v nich co nejspecifičtější společné vlastnosti.

Dále však lze učení dělit podle dalších kritérií, jedním z nich je učení *on-line* a *off-line*.

Off-line učení je metoda učení, při které je síť nejprve trénována na celé množině dat a až v okamžiku, kdy pro každý vstupní vektor podává očekávané výsledky, váhy jednotlivých propojení jsou uchovány a síť je nasazena do provozu, kde už se nemění.

Výhodou takto naučené sítě je nesporně to, že se chová velmi předvídatelně a pro každý vstup podá korektní výstup. Nevýhodou však je, že v okamžiku, kdy chceme do množiny dat přidat další vstup, celá síť musí být znova natrénována. Další nevýhodou těchto metod učení je, že už v okamžiku učení musí být k dispozici všechny možné kombinace vstupů s jejich očekávanými výstupy a také že učení může velmi dlouho trvat.

Nejnámějším zástupcem těchto metod učení je algoritmus *backpropagation*, který je popsán dále v této kapitole.

On-line učení je dynamičtější metoda učení. Pokud je do tímto způsobem učené sítě potřeba přidat další vstupní vektor, je možné to udělat za běhu bez ztráty dříve získané informace. Výhodou sítí učených on-line tedy je možnost učení za běhu.

Pro komplikované, ale exaktně definované problémy je dnes ještě výhodnější používat metody off-line učení. Výzvou do budoucnosti neuronových sítí však je přijít s takovým algoritmem učení, který by dosahoval přesnosti off-line algoritmů, ale dokázal by pracovat s novými vstupními vektory za běhu, tedy on-line.

Učení backpropagation

Když byl algoritmus backpropagation v roce 1986 představen Davidem Rumelhartem [9], znamenalo to velký průlom v učení neuronových sítí. Na některých neuronových sítích dosahoval oproti předchozím algoritmům několikanásobně vyšší rychlosti učení. Tím umožnil využití neuronových sítí k řešení problémů, které do té doby byly neřešitelné. [7]

Pro ilustraci práce backpropagation nyní zavedme zjednodušující značení.

w_{jk}^l značí váhu na propojení z k -tého neuronu ($l - 1$). vrstvy do j -tého neuronu l -té vrstvy neuronové sítě.

y_j^l je výstup j -tého neuronu v l -té vrstvě.

b_j^l je bias, který vstupuje do j -tého neuronu l -té vrstvy.

Výstupní hodnotu každého neuronu lze tedy v souladu s touto notací zapsat jako

$$y_j^l = \sigma \left(\sum_{j,k} w_{jk}^l y_k^{l-1} + b_j^l \right), \quad (3.3)$$

kde σ je jeho aktivační funkce, k jde přes všechny neurony z vrstvy $l-1$ a j jde přes všechny neurony vrstvy l .

Pro každou vrstvu l je dále definovaná váhová matice w^l

$$w^l = \begin{bmatrix} w_{11} & w_{21} & \cdots & w_{n1} \\ w_{12} & w_{22} & \cdots & w_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1m} & w_{2m} & \cdots & w_{nm} \end{bmatrix}, \quad (3.4)$$

která přehledně znázorňuje všechny váhy. Hodnota, která se v matici nachází v j -tém sloupci k -tého řádku značí váhu, která se využívá při propojení z k -tého neuronu $l-1$. vrstvy do j -tého neuronu l -té vrstvy sítě. Podobným způsobem je definován vektor biasů b^l pro každou vrstvu a vektor výstupních hodnot y^l , také pro každou vrstvu. Oba dva obsahují postupně hodnoty pro všechny neurony dané vrstvy od 1 do n . Na výše definované vektory lze také aplikovat aktivační funkci σ , kdy platí následující vztah:

$$\sigma \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \sigma(x_1) \\ \sigma(x_2) \\ \sigma(x_3) \end{bmatrix}. \quad (3.5)$$

Se zavedeným zápisem už lze vztah pro výpočet výstupních hodnot vyjádřit pro celou jednu vrstvu zároveň, což nám rovnou umožní přestat o neuronové síti přemýšlet jako o množině neuronů, ale jako o množině vrstev, což je pro účely dalšího popisu praktičtější. Výpočet tedy oproti vztahu 3.3 zpřehledníme na

$$y^l = \sigma \left(w^l y^{l-1} + b^l \right), \quad (3.6)$$

$$u^l = w^l y^{l-1} + b^l. \quad (3.7)$$

Součástí vztahu 3.6 je také *vektor vnitřních potenciálů* neuronů vrstvy l - u^l (viz 3.7), která bude v následujícím výkladu taktéž využita.

Chybu sítě pro aktuální vstupní vektor vypočteme podle následujícího vztahu pro kvadratickou chybu sítě

$$C_x = \frac{1}{2} (d(x) - y^L(x))^2 \quad (3.8)$$

kde $d(x)$ je očekávaný výstup sítě pro vstup x , $y^L(x)$ je výstup poslední vrstvy sítě, tedy primární výstup sítě, pro vstup x . Pro celou množinu vstupů se potom využívá vztah následující vztah, podle kterého vypočteme průměrnou chybu C pro všechny vstupní vektory:

$$C = \frac{1}{n} \sum_x C_x = \frac{1}{2n} \sum_x (d(x) - y^L(x))^2. \quad (3.9)$$

Dále zavedeme operaci \odot , v literatuře též nazývanou *Hadamardův součin*. Pro dvě matice o stejných dimenzích $m \times n$ A a B je matice $C = A \odot B$ maticí taktéž o dimenzích $m \times n$, kde $C_{ij} = A_{ij} \times B_{ij}$.

Algoritmus backpropagation experimentálně zjišťuje, jaký vliv mají malé změny vah a biasů u jednotlivých neuronů na výslednou chybu sítě. Formálně tedy jde o výpočet parciálních derivací zapsaných v následujícím vztahu:

$$\frac{\partial C}{\partial w_{jk}^l}, \frac{\partial C}{\partial b_j^l}. \quad (3.10)$$

V procesu učení backpropagation tedy provádíme malé změny Δu_j^l vnitřního potenciálu j -tého neuronu v l -té vrstvě a zjišťujeme, zda toto bude mít pozitivní vliv na výslednou chybu sítě C . Tyto změny vnitřního potenciálu jsou prováděny pomocí změny vah u jednotlivých propojení. Tyto jsou propagovány zpět do celé sítě po výpočtu výstupu y^L a chyby sítě C . Pro neurony výstupní vrstvy je hodnota Δw_{jk}^l vypočtena podle vztahu 3.11, kde δ_j^L je vypočtena podle vztahu 3.12 a η je konstanta pro učení celé sítě nazývaná *learning rate*, která se většinou pohybuje v intervalu $(0; 1)$:

$$\Delta w_{jk}^l = \eta \delta_j^L y_k^{L-1} \quad (3.11)$$

$$\delta_j^L = \frac{\partial C}{\partial y_j^L} \sigma'(u_j^L) \quad (3.12)$$

Převedení vztahu 3.12 do maticové reprezentace, se kterou chceme při backpropagation pracovat už není obtížné. Lze jej zapsat jako 3.13, kde $\nabla_y C$ je vektor, jehož komponenty jsou parciální derivace $\frac{\partial C}{\partial y_j^L}$. Pro kvadratickou chybu sítě lze tedy tento vztah zjednodušit na vztah 3.14.

$$\delta^L = \nabla_y C \odot \sigma'(u^L) \quad (3.13)$$

$$\delta^L = (y^L - d) \odot \sigma'(u^L) \quad (3.14)$$

Výpočet δ^l pro každou další vrstvu se potom provádí od poslední vrstvy směrem k první podle vztahu 3.15, kde $(w^{l+1})^T$ je transponovaná váhová matice pro následující vrstvu. V takovéto podobě ji lze využít pro zpětný přenos chyby sítě k jednotlivým neuronům přesně podle jejich významu při dopředném průchodu. Váhy pro jednotlivá propojení se potom změní o Δw_{jk}^l , který je pro skryté vrstvy vypočten analogicky jako pro výstupní vrstvu dle vztahu 3.11.

$$\delta^l = \left((w^{l+1})^T \delta^{l+1} \right) \odot \sigma'(u^L) \quad (3.15)$$

3.4 Konvoluční neuronové sítě

Konvoluční neuronové sítě (též *hluboké neuronové sítě*) jsou neuronové sítě navržené speciálně pro zpracovávání obrazových dat. Na rozdíl od plně propojených neuronových sítí, které jsme uvažovali v předchozích kapitolách, konvoluční neuronové sítě dokáží s výhodou

využít prostorového uspořádání jednotlivých vstupních hodnot načítaných z obrázku. Konvoluční neuronové sítě se skládají ze tří typů vrstev, které jsou podrobně popsány dále v této kapitole: *konvoluční vrstva*, *pooling vrstva* a nakonec *výstupní vrstva*.

Odlišnost konvolučních neuronových sítí tkví v chápání jednotlivých vrstev sítě. V plně propojené síti je každá vrstva chápána jako lineární pole neuronů, zatímco v konvoluční neuronové síti je možné ji chápat jako dvourozměrné pole neuronů. Toto pole svým uspořádáním bude odpovídat uspořádání pixelů ve zpracovávaném obrázku.

Pro tuto kapitolu si vytvoříme jednoduchý ilustrativní obrázek o rozměrech 10×10 pixelů, nad kterým budeme demonstrovat fungování konvolučních neuronových sítí a zároveň provedeme srovnání počtu hledaných parametrů konvoluční a plně propojené sítě.

3.4.1 Vstupní vrstva

Vstupní vrstva konvoluční neuronové sítě slouží k načtení vstupu. Stejně jako u plně propojených sítí, ani v konvoluční síti vstupní vrstva neprovádí žádné zpracování informace a nepočítá se do celkového počtu vrstev. Rozměry vstupní vrstvy neuronové sítě určují rozměry obrázků, které bude možné pomocí dané sítě zpracovávat, v pixelech.

3.4.2 Konvoluční vrstva

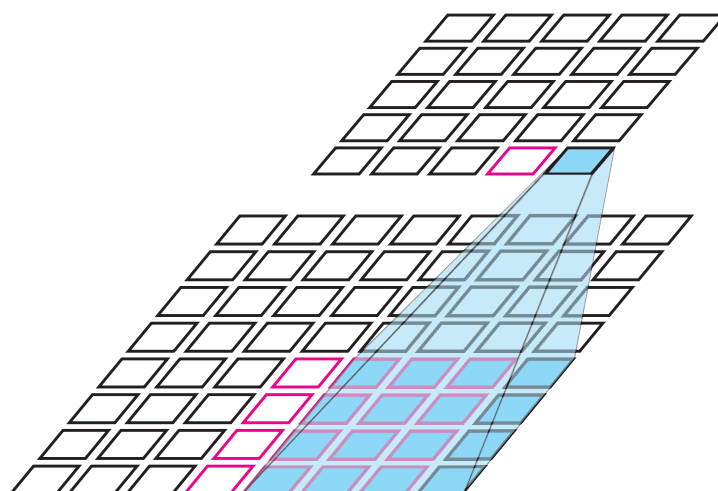
Konvoluční vrstva je stěžejním elementem konvolučních neuronových sítí a zároveň první vrstvou, která nad daty ve vstupní vrstvě provádí nějakou operaci. Konvoluční vrstva se skládá z několika map rysů (*feature maps*), které pracují nezávisle na sobě.

Feature map

Feature map, tedy mapa rysů je analogií dříve popsaného *vektoru rysů*. Popisuje primitiva, která se v analyzovaném vstupu nacházejí s tím rozdílem, že mapa rysů ukazuje i jejich přesnou polohu ve zdrojovém obrázku. Takovýmto primitivem tak může být například hrana v obraze. Každá mapa rysů však mapuje výskyty jedinného primitiva, v reálných aplikacích je tedy těchto map použitých vždy více. Práci konvoluční vrstvy v této kapitole ukáží na neuronové síti se třemi mapami rysů, v praxi je však zpravidla využíváno map rysů několikanásobně více.

Každá mapa rysů pracuje nezávisle na ostatních způsobem, který je zobrazen na obrázku 3.6. Vezměme v potaz obrázek o rozměrech 10×10 pixelů, který už máme načtený ve vstupní vrstvě a nyní má být transformován na mapu rysů. Hodnota každého neuronu v konvoluční vrstvě je vypočtena z hodnot podmnožiny neuronů vstupní vrstvy - z tzv. *lokální pole* (anglicky *local receptive field*), nikoliv ze všech jako tomu je u plně propojených sítí. V případě, který je uveden na obrázku 3.6 se jedná o podčást pole neuronů vstupní vrstvy o rozměrech 4×4 neuronů (označeno modře). Jakmile je vypočtena hodnota, pole se posouvá a je vypočtena hodnota dalšího neuronu z nové podmnožiny neuronů vstupní vrstvy (v obrázku 3.6 označeno růžovou barvou)

Krok – počet neuronů, o které se lokální pole posune mezi výpočtem hodnot dvou sousedních neuronů v mapě rysů. Základní a zároveň hojně využívanou hodnotou je 1, ale bývá experimentováno i s jinými. Při délce kroku 1 a velikosti lokálního pole 4×4 po vstupní vrstvě o velikosti 10×10 neuronů následuje mapa rysů o velikosti 5×5 neuronů. Pokud tedy zvolíme delší krok, ve výsledné síti se bude nacházet méně neuronů, což lze někdy s výhodou využít.



Obrázek 3.6: Schéma kroku konvoluční vrstvy sítě.

Sdílené váhy - mapa rysů provádí prohledání celého vstupního obrázku s cílem nalézt jeden konkrétní rys, proto není třeba pro každé lokální pole odvozovat váhy jednotlivých propojení. Pro každou mapu rysů stačí pouze jedinná matice vah s jedním BIASem, která je použita pro výpočet hodnoty každého neuronu.

Složení několika map rysů vzniká konvoluční vrstva. Pro ilustraci v této práci budeme uvažovat konvoluční vrstvu složenou ze tří map rysů, v praxi se však využívají mnohonásobně rozsáhlejší konvoluční vrstvy.

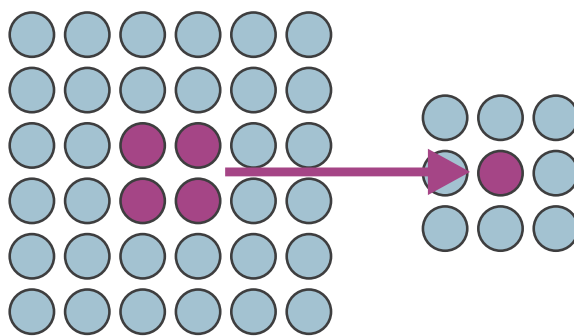
3.4.3 Pooling vrstva

Konvoluční vrstva dokáže lokalizovat jednotlivé rysy v obrázku a určit tak jejich přesnou polohu. S faktem, že ve většině reálně nasazených sítí je využito větší množství map rysů roste i náročnost výpočtů nad konvoluční neuronovou sítí. Proto je do sítě zařazována tzv. *pooling vrstva*. Tato vrstva vytváří z každé mapy rysů tzv. *kondenzovanou mapu rysů*. Pooling vrstva často bezprostředně následuje za konvoluční vrstvou a zajišťuje zmenšení informace, která je výstupem konvoluční vrstvy.

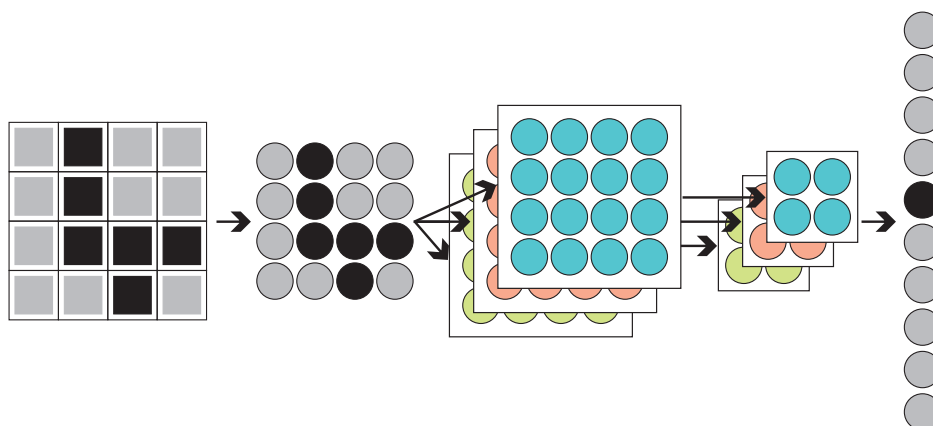
Pooling lze provádět několika různými způsoby, princip je ale vždy totožný: každý neuron pooling vrstvy bere v potaz jenom malou podmnožinu neuronů předchozí vrstvy a z nich vypočítává svoji hodnotu. Jednotlivé podmnožiny se nepřekrývají, čímž je získána větší komprese.

Max pooling

Max pooling je nejjednodušší a zároveň velmi významná metoda poolingů. Hodnota neuronu je vypočtena jako maximum z hodnot podmnožiny neuronů předchozí vrstvy, na kterou navazuje. Pokud se tedy rys, který je v rámci dané mapy ve vstupním obrázku hledán, nalezen někde v okolí analyzovaného místa, tato informace je zachována, zatímco jeho přesná poloha je zahazena.



Obrázek 3.7: Schéma kroku pooling vrstvy.



Obrázek 3.8: Struktura hluboké neuronové sítě.

L2 pooling

Další z velmi hojně využívaných metod poolingů. Hodnota neuronu je tentokrát opět vypočtena z podmnožiny neuronů předchozí vrstvy, tentokrát jsou v ní však zohledněny hodnoty všech neuronů, nikoliv jen ta nejvyšší. Hodnota neuronu n je vypočtena dle následujícího vztahu

$$n = \sqrt{\sum_{i=0}^{i < k} n_{pi}^2}, \quad (3.16)$$

kde $k = l^2$, kde l značí počet neuronů v hraně čtverce a n_{pi} je i -tý neuron předchozí vrstvy. Jeden krok poolingů je znázorněn na obrázku 3.7, v této vrstvě je $k = 4$ a $l = 2$.

3.4.4 Konvoluční síť

Trénujeme jednoduchou konvoluční neuronovou síť, která klasifikuje obrázky do deseti tříd (například obrázky s číslicemi 0-9). Taková síť bude obsahovat několik dvojic konvoluční vrstvy s pooling vrstvou, za kterými bude následovat deset neuronů výstupní vrstvy, ze kterých vždy právě jeden bude v aktivním stavu. Mezi výstupní a předposlední vrstvou je propojení všech neuronů mezi sebou – poslední vrstva je tedy plně propojená. Příklad velmi jednoduché konvoluční sítě je znázorněn na obrázku 3.8.

3.5 Finetuning

Mezi další důležité prvky využití v rámci této práce patří *finetuning*. Jedná se o metodu využívanou v situacích, kdy už máme úspěšně natrénovanou síť a chceme ji využít ke klasifikaci podobných dat. Za běžných okolností se na počátku tréninku neuronové sítě na nový typ vstupu všechny váhy inicializují náhodně. Pokud však máme k dispozici síť, která je již naučená na velké množině obrázků, můžeme namísto náhodné inicializace nové sítě inicializovat váhy do výsledného stavu vah jiné sítě. Proces učení této sítě se potom nazývá *finetuning*. Ten lze provádět více způsoby. Nejznámějšími z nich jsou:

- Úprava vah pouze v poslední vrstvě,
- přidání několika nových vrstev do naučené neuronové sítě.

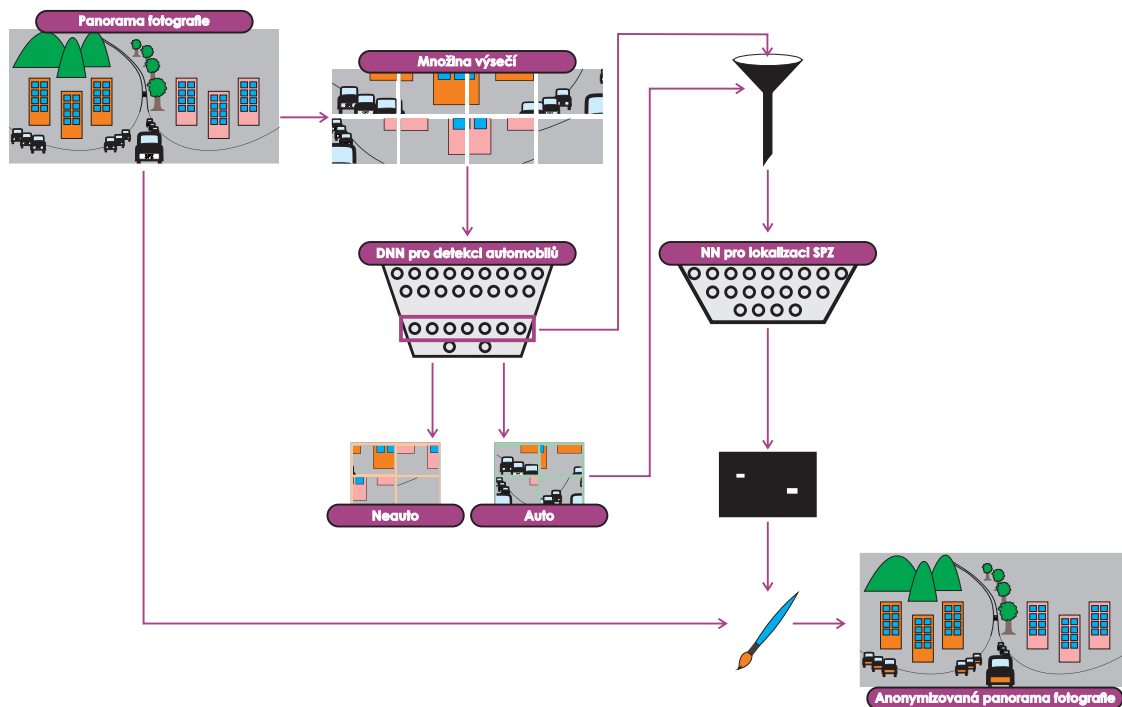
Základním principem finetuningu je, že do vah v naučené části sítě není příliš zasahováno. Váhy původní sítě mohou být buďto striktně zachovány v původním stavu nebo mohou být měněny jen minimálně. V této práci byla pro finetuning zvolena varianta zachování původních vah v nezměněném stavu.

Kapitola 4

Návrh

Tato kapitola je věnována návrhu programu určenému k anonymizaci SPZ ve fotografiích pro službu Panorama. Cílem této práce je vyrobit nástroj, který dokáže automaticky načítat nafocené obrázky na vstupu a podávat anonymizované obrázky na výstupu. Detekce SPZ je v současnosti intenzivně řešeným a mnohdy i vyřešeným problémem. Jak uvádí [6], detekce SPZ s využitím pokročilých metod zpracování obrazu dosahuje velmi dobrých výsledků.

Zásadní odlišností detekce SPZ v této práci je rozmanitost zpracovávaných dat. Data, ve kterých jsou běžně SPZ detekovány, pochází ze statických kamer umístěných v ideálním úhlu oproti automobilovému provozu – například na branách nad dálnicemi, radarech nebo jiných kamerách mířících na projíždějící automobily v konstantním úhlu. Data pro tuto práci však jsou snímána z jedoucího vozidla v 360° úhlu a to i v méně typických lokalitách, než jsou hlavní dopravní tepny. Takový typ dat si tedy vyžaduje speciální zacházení.



Obrázek 4.1: Základní struktura programu.



Obrázek 4.2: Sekce panorama fotografie usnadňující její analýzu.

Práce s panoramatickými fotografiemi je rozdělena do několika základních fází znázorněných na obrázku 4.1. Procesy, které jsou nad daty v jednotlivých fázích prováděny, budou v této kapitole podrobně popsány.

V první fázi jsou panoramatické fotografie analyzovány a dále rozřezány na menší výseče, jak je popsáno v části 4.1. Takto připravené výseče fotografií jsou ručně klasifikovány do připravených tříd. klasifikované obrázky jsou potom využity pro učení hluboké neuronové sítě k detekci vozidel (viz část 4.2) a neuronové sítě pro lokalizaci SPZ vozidel (viz část 4.3).

V části 4.4 je potom popsán způsob, jakým jsou z výsledků těchto neuronových sítí extrahovány souřadnice SPZ určené k anonymizaci a jak je provedena.

4.1 Předzpracování dat

Na primárním vstupu celého programu jsou přijímány fotografie nasnímané v rámci sběru dat pro službu Panorama. Tyto fotografie jsou pořizovány způsobem, který byl popsán dříve v této práci (viz část 2.4).

Součástí zadání této práce je datová množina připravená zadavatelem. Tato množina byla vybrána s ohledem na okrajové případy, které mohou při snímání dat pro službu Panorama nastat. Jedná se o:

- kolemjedoucí vozidla,
- vozidla zaparkovaná v různých úhlech okolo silnice,
- motocykly, autobusy a jiná méně častá vozidla.

Dodaná datová sada obsahuje 178 fotografií o rozměru $12\,288 \times 6\,144$ pixelů. V rámci předzpracování dat byla nejprve provedena analýza datové množiny poskytnuté společností Seznam.cz a.s.. Dále pak byl na základě této analýzy navržen způsob předzpracování dat, který by usnadnil následnou práci hlubokých neuronových sítí.

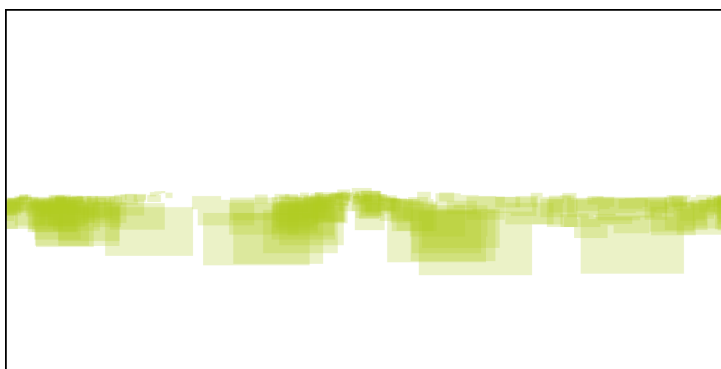
V rámci analýzy těchto dat byly sledovány automobily nacházející se na dodaných fotografiích, konkrétně pak:

- velikost automobilu na fotografii (v pixelech),
- umístění automobilu ve fotografii.

	Výška automobilu	Šířka automobilu
Průměr [px]	270	420
Medián [px]	227	332
Modus [px]	150	180
Maximum [px]	1080	1920
Minimum [px]	30	60

Tabulka 4.1: Velikosti vozidel v obrazovém materiálu pro službu Panorama.

Pro tuto analýzu byla fotografie rozdělena do devíti sekcí (viz obrázek 4.2). V rámci dodaného datasetu potom byly ručně lokalizovány automobily, které se na fotografiích nacházely a byla zaznamenávána jejich velikost a příslušnost k jednotlivým sekcím obrazu. Z tohoto procesu byla vytvořena mapa výskytů vozidel znázorněná na obrázku 4.3. Celkem bylo zaznamenáno 286 vozidel, z nichž všechny příslušely některé ze sekcí 7, 8 a 9. Statistiky velikostí automobilů jsou uvedeny v tabulce 4.1 ¹.



Obrázek 4.3: Zátěžová mapa ukazující místa výskytu a velikosti automobilů v analyzovaných datech.

Na základě provedené analýzy byl jako nejobecnější rozměr automobilu vybrán medián všech velikostí. Pro další zpracování byla část fotografií, na které se automobily převážně nachází (tedy sekce 7, 8 a 9 dle obrázku 4.2) rozřezána na nepřekrývající se výseče o velikosti 332×227 px.

4.2 Detekce automobilů

V dalším kroku jsou jednotlivé výseče vytvořené v rámci předzpracování klasifikovány do jedné ze dvou tříd - *auto*/*neauto*. To je realizováno pomocí hluboké neuronové sítě.

V této práci je k tomuto účelu využita již naučená neuronová síť *Ilsvrc* [1], jejíž architektura je zobrazena na obrázku 4.4. Tato síť vznikla v rámci výzvy *Ilsvrc*, která se koná každoročně a je zaměřena na algoritmy pro lokalizaci a detekci objektů v obraze a také na klasifikaci scény. Síť *Ilsvrc* byla naučena na množině obrázků *ImageNet*, která obsahuje více než 14 milionů obrázků rozřazených do 1 000 kategorií [1].

¹Rozdíly hodnot u jednotlivých sekcí obrázku byly nepatrné, proto byly zanedbány.



Obrázek 4.4: Struktura hluboké neuronové sítě pro detekci automobilů.

Síť Ilsvrc byla v rámci této práce pomocí finetuningu optimalizována pro klasifikaci fotografií pro službu Panorama. Pro další zpracování však není využíván pouze primární výstup klasifikace. K dalšímu zpracování je také extrahován vektor rysů, který je výstupem jedné z vrstev této neuronové sítě (viz obrázek 4.5). V této práci jsou zkoumány výstupy dvou vrstev použité konvoluční sítě.

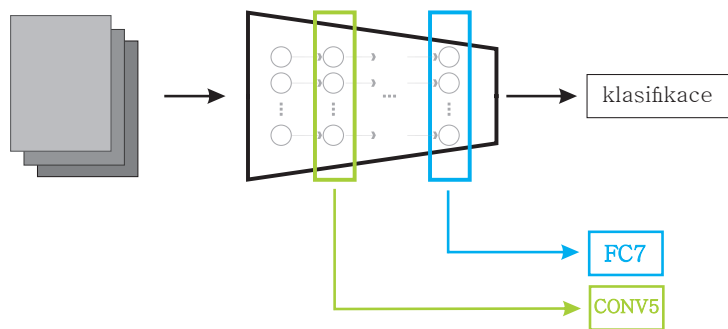
1. Výstup předposlední (sedmé) výpočetní vrstvy sítě, která je plně propojená (*fully connected*) – dále jen *FC7*.
2. Výstup poslední konvoluční vrstvy sítě (ještě před aplikací aktivační funkce), která je pátou konvoluční vrstvou v celé síti – dále jen *CONV5*.

4.3 Lokalizace SPZ

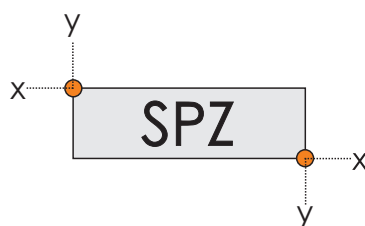
Další navazující částí této práce je lokalizace SPZ. Do této fáze zpracování vstupují vektory rysů odpovídající těm vzorkům, které byly v rámci předchozího kroku klasifikovány jako auto. Na vstup sítě pro detekci SPZ jsou tedy přiváděny výstupní hodnoty jedné z vrstev sítě pro detekci automobilů.

V této části se budu věnovat dvěma možným způsobům, které v rámci této práce budou porovnávány. Liší se vrstvou sítě pro detekci automobilů, jejíž výstupní hodnoty jsou do sítě pro detekci SPZ načteny jako vstup.

FC7 - předposlední vrstva neuronové sítě pro detekci automobilů. Jedná se o lineární pole reálných hodnot o délce 4096. Prostorová informace je po průchodu předchozími vrstvami již značně zploštělá. Pokud by se však podařilo z tohoto velmi kondenzovaného



Obrázek 4.5: Schéma detekce automobilů.



Obrázek 4.6: Identifikace rohů SPZ.



Obrázek 4.7: Identifikace SPZ pomocí masky.

vektoru rysů získat souřadnice SPZ vozidla, znamenalo by to velkou úsporu, jak co se výpočetní, tak co se paměťové náročnosti výpočtu týče. Neuronová síť by v tomto případě totiž mohla být velmi malá a celý výpočet by bylo možné dostat do několika vrstev.

CONV5 - pátá konvoluční vrstva neuronové sítě pro detekci automobilů. Jedná se o výstup 256 konvolučních filtrů ještě před aplikací aktivační funkce. Každý z těchto filtrů produkuje mapu rysů o rozměru 13×13 , jedná se tedy o $13 \times 13 \times 256 = 43\,264$ hodnot pro každý obrázek. Ačkoliv se ve srovnání s předchozím navrženým přístupem může tato reprezentace zdát jako nevýhodná, lze očekávat, že v této 3D matici bude obsaženo o mnoho více prostorové informace o vstupním obrázku a tedy povede k lepším výsledkům.

Na výstupu této části zpracování jsou opět porovnávány dva navržené přístupy. Neuronová síť bude buďto produkovat přímo čtveřici souřadnic identifikující lokalizovanou SPZ tak, jak je naznačeno na obrázku 4.6 nebo bude produkovat matici, která bude zastupovat masku, po jejímž pronásobení s analyzovanou výsečí bude získána část obsahující SPZ (viz obrázek 4.7).

4.4 Zatemnění SPZ

Poslední fází zpracování je namapování detekované SPZ zpět do celého původního obrázku a její rozmazání. Souřadnice výřezu, ve kterém byla SPZ vyhledávána, jsou známy z předchozích kroků zpracování. Namapování SPZ zpět do souřadnic celé Panorama fotografie tedy bude provedeno prostým součtem souřadnic $[x_1^{vyrez} + x_1^{SPZ}, y_1^{vyrez} + y_1^{SPZ}]$.

Na část fotografie, na které byla lokalizována SPZ vozidla, potom bude aplikován rozostřovací filtr nástroje ImageMagick.

Kapitola 5

Implementace

Výsledný program sloužící k detekci automobilů a lokalizaci jejich SPZ ve fotografiích pro službu Panorama je implementován v jazyce `Python2`. Skládá se z hlavního skriptu `anonymize.py` a python balíku `anonymize`, který obsahuje další skripty, objekty a metody využívané k anonymizaci. S výhodou jsou využívány některé systémové knihovny jazyka Python, jako jsou `os`, `numpy` nebo `subprocess`.

Program pracuje i s dalšími knihovnami, které nejsou součástí standardní distribuce. Mezi ně patří i `C++` framework `Caffe`, který zajišťuje velmi robustní zázemí pro práci s hlubokými neuronovými sítěmi. Dalším z využitých frameworků je `Python` framework `Keras`, který zajišťuje rychlou a jednoduchou práci s menšími neuronovými sítěmi.

V této části jsou nejprve, v sekci 5.1, popsány technologie a postupy využité k implementaci výsledného programu. Dále, v sekci 5.2, jsou potom popsány další skripty, které byly vytvořeny pro usnadnění ručního předzpracování dat a automatizaci vyhodnocení výsledků.

5.1 Výsledný program

Celý výpočet je řízen z hlavního skriptu `anonymize.py`, odkud jsou v jednotlivých fázích výpočtu volány další moduly.

5.1.1 Předzpracování dat

V rámci předzpracování dat, před jejich vlastním odesláním na vstup neuronové sítě, je provedeno rozřezání každé fotografie na 259 výšecí o rozměru 227×332 px. Toto je realizováno v `Bash` skriptu `crop.sh`. Z tohoto skriptu je dále volán nástroj `crop` ze sady `ImageMagick`, který musí být na stroji předem nainstalován.

V rámci předzpracování jsou rozřezány všechny obrázky a je vytvořena složka `crops`. Tato složka obsahuje výšece všech zpracovávaných obrázků a nad ní probíhají další kroky výpočtu.

5.1.2 Detekce automobilů

K detekci automobilů je využit framework `Caffe`, který je napsán v jazyce `C++`. Z hlavního python skriptu je s ním pracováno skrze poskytované python rozhraní `pycaffe`.

Nastavení a architektura sítě jsou pro `Caffe` specifikovány v konfiguračních souborech s notací `protobuf`. Každá síť je definována pomocí dvou souborů.

Solver.prototxt je konfigurační soubor pro učení sítě, ve kterém je možné specifikovat všechny důležité parametry, jako je počet kroků učení, koeficient učení, způsob jeho změn, ale i množství informačních výpisů v průběhu učení, počet zachycených snímků sítě v průběhu učení a další.

Tento soubor při následném používání naučené sítě již není třeba.

Trainval.prototxt je konfigurační soubor, který obsahuje celou specifikaci architektury sítě se všemi jejími vrstvami, aktivačními funkcemi, parametry zpětné propagace chyby sítě a způsobem inicializace jednotlivých vah.

Tento soubor je vyžadován jak při učení, tak při vlastním použití neuronové sítě ke klasifikaci neznámých vzorků (s drobnými úpravami). Zatímco varianta pro učení sítě obsahuje vrstvu pro načítání trénovacích dat a vyhodnocení úspěšnosti, při použití naučené sítě je třeba načtení dat obstarat mimo neuronovou síť. Pro soubor obsahující architekturu sítě určené ke klasifikaci neznámých vzorků se používá název `deploy.prototxt`.

Učení a klasifikace

V průběhu učení je spouštěn přímo program `caffe` a jeho metoda `train`. Po každých n iteracích jsou ukládány aktuální hodnoty vah do binárního souboru typu `caffemodel`. Po analýze průběhu učení je vybrána ta uložená sada vah, se kterou řešení dosahuje nejlepších výsledků jak na trénovací, tak na validační množině. Tato je potom použita ke klasifikaci testovacích, případně dalších neznámých dat. Klasifikace testovací množiny pomocí připravené sítě je zajišťována pomocí skriptu `classify.py` z vytvořeného balíku `anonymize`.

V rámci učení bylo nastavováno množství parametrů, mezi které patří:

Interval validate – nastavená parametrem `test_iter` určuje, jak často bude prováděna validace modelu, tedy jak často bude vypočtena jeho úspěšnost na validační sadě.

Počáteční learning rate – určuje, jak velkou roli hraje zpětná propagace chyby ve změně vah v síti od začátku učení.

Learning rate policy – protože v průběhu učení se výstup neuronové sítě přibližuje k cílovému řešení, je nutné postupně snižovat změny vah, aby nedocházelo k oscilaci okolo správného řešení. Toto lze dělat několika způsoby, mezi kterými je voleno pomocí parametru `lr_policy`.

Počet iterací – počet iterací je důležitým parametrem učení neuronové sítě. Po jistém počtu iterací již v síti nedochází k dalšímu zlepšování a tedy není žádoucí v učení pokračovat. Tyto hodnoty se pro jednotlivé typy problémů značně liší. Pro problém detekce automobilů byla volena hodnota 50 000 iterací.

Snapshot – zachycování snímků sítě - tzv. *snapshotting* slouží k jejich dalšímu využití, ať už k testování pomocí testovací množiny nebo ke klasifikaci neznámých dat. Tyto snímky není třeba dělat příliš často, při učení neuronové sítě byl volen interval 10 000 iterací.

Architektura sítě

Využitá síť obsahuje 14 vrstev, které jsou ilustrovány výše na obrázku 4.4. Váhy mezi všemi vrstvami zůstaly během učení sítě v původním stavu (tedy po natrénování na sadu

ImageNet), pouze váhy mezi vrstvami FC7 a FC8 byly iniciovány náhodně a trénovány přímo na míru klasifikaci výsečí ze snímků pro službu Panorama.

Zajímavým prvkem architektury sítě jsou vrstvy Drop6 a Drop7, které následují za oběma plně propojenými vrstvami. Jedná se o tzv. *dropout* vrstvy. Při učení s využitím dropoutu jsou v jednotlivých iteracích náhodně vyřazovány neurony dropoutované vrstvy. Takto navržená síť se tedy neučí přímo na míru trénovací sadě, protože v každé iteraci má mírně odlišnou architekturu.

Tyto vrstvy nehrají žádnou roli při dopředném průchodu sítí, nicméně jsou velmi důležité ve fázi učení. Jak uvádí [12], využití této metody při tréninku velkých neuronových sítí vede ke značnému omezení přeučování.

Paralelizace

Protože učení neuronové sítě je výpočetně velmi náročným procesem, byly zváženy všechny dostupné metody paralelizace. Jedním z hlavních argumentů pro využití frameworku Caffe je jeho jednoduchá paralelizace na grafické kartě s architekturou CUDA.

Možnost paralelizovat výpočet se vztahuje jak k učení sítě, tak k vlastní klasifikaci vzorků, což je nespornou výhodou vzhledem k velikosti datasetu, na kterém má být anonymizér nasazen v provozu.

5.1.3 Lokalizace SPZ

Pro lokalizaci SPZ v rámci výseče byla zvolena jednodušší neuronová síť. Potřeba využití frameworku Caffe tedy odpadla a proto byl zvolen framework Keras, který ve srovnání s Caffe poskytuje o mnoho jednodušší a přívětivější uživatelské rozhraní. Keras může běžet jako nástavba nad vysoce optimalizovanými matematickými python knihovnamy Theano nebo TensorFlow.

Učení a klasifikace

Učení a architektura sítě jsou definovány v jedinném souboru, který je implementován v jazyce Python. Jakmile je dokončen požadovaný počet iterací, automaticky je uložena jak architektura sítě do souboru typu json, tak finální váhy pro jednotlivé vrstvy do souboru typu hdf5.

Klasifikace je opět řízena ze skriptu v jazyce Python, do nějž je načtena jak architektura sítě, tak váhy pro jednotlivá propojení.

Architektura sítě

Definice architektury sítě je ve frameworku Keras prováděna přímo v jazyce Python. V rámci této práce byly porovnávány dvě varianty vstupních dat pro lokalizaci SPZ, což si samozřejmě žádalo i využití dvou různých architektur sítě.

Lineární vstup FC7 – pro variantu, ve které je na vstupu načítáno lineární pole 4096 hodnot, byla zvolena jednoduchá regresní síť se čtyřmi plně propojenými vrstvami, které postupně snižovaly počet výstupů až se dostaly na požadované 4 hodnoty určující souřadnice x a y dvou protilehlých rohů SPZ.

Jako aktivační funkce byla u všech vrstev využita funkce ReLU. Jako metoda zpětné propagace chyby byla zvolena metoda *Stochastic gradient descent* (dále jen *SGD*).

Maticový vstup CONV5 – pro maticovou variantu vstupu byla využita jedinná konvoluční vrstva. Výběr vhodné aktivační funkce pro tuto úlohu byl předmětem zkoumání v rámci této práce. Porovnávány byly funkce sigmoida a ReLU.

Paralelizace

Při implementaci neuronové sítě pro tuto práci jsem zvolila backendovou vrstvu **Theano**, která sice na rozdíl od varianty **TensorFlow** neběží na grafické kartě automaticky, nicméně pomocí argumentů příkazové řádky lze výpočet na GPU zapnout jak při učení neuronové sítě, tak při vlastní klasifikaci neznámých vzorků.

5.1.4 Zatemnění SPZ

Jakmile je SPZ v obraze lokalizována, poslední zbývající krok je její úprava do formy, ve které není čitelná ani lidským okem, ani strojově. Toto je zajištěno lokální aplikací rozostřovacího filtru ze sady nástrojů **ImageMagick**. V tomto nástroji jsou dostupné dvě varianty – **blur** a **Gaussian blur**.

Zpracování pomocí filtru **blur** je dle dokumentace rychlejší, jeho nevýhoda oproti filtru **Gaussian blur** je provedení ve dvou průchodech. To může způsobit zaokroulovací chybu při ukládání mezivýsledku. Tato nevýhoda ale s ohledem na účel použití a rozsah plochy, která je rozostřována, není směrodatná a proto byl zvolen filtr **blur** [14].

5.2 Pomocné skripty

Zásadní část této práce spočívala v přípravě a klasifikaci použité datové sady. Protože množství zpracovávaných dat bylo obrovské, každá alespoň částečně strojově proveditelná činnost celý proces přípravy datové sady urychlila. Pro tyto účely byly napsány pomocné skripty nejen v jazycích **Python** a **R**, ale důležitým prvkem byl i jazyk **JavaScript**.

5.2.1 Ruční klasifikace

Aby neuronová síť měla trénovací obrázky správně klasifikovány, je třeba je nejprve ručně anotovat. Toto bez zásahu lidské ruky zrealizovat nelze, celé zpracování však lze urychlit pomocí automatického zobrazování trénovacích obrázků a ukládání zvolených hodnot.

Zobrazovací nástroj byl pro tuto práci vytvořen pomocí webových technologií (**HTML**, **JavaScript**). Aby mohla být data přímo ukládána do souboru určeného k dalšímu zpracování, byl na zobrazovací nástroj připojen jednoduchý **Http** server v jazyce **Python**.

5.2.2 Zpracování výsledků

Pro automatické porovnávání výstupů neuronové sítě s očekávanými výstupy byla připravena sada skriptů v jazycích **Python** a **R**. Pomocí těchto skriptů byly zpracovávány výsledky klasifikace jednotlivých neuronových sítí a vyčíslovány chyby jak v průběhu výpočtu, tak po jeho skončení.

Pro vyhodnocení výsledků byla navržena speciální metoda vyčíslení správně a chybně detekovaných oblastí ve fotografii. Tato metoda spočívá ve vypočtení velikosti čtyř oblastí:

1. Oblast s SPZ detekovaná jako SPZ - *true positive*
2. Oblast s SPZ nedetekovaná jako SPZ - *false negative*

3. Oblast bez SPZ detekovaná jako SPZ - *false positive*

4. Oblast bez SPZ nedetekovaná jako SPZ - *true negative*

Pomocí těchto čtyř hodnot je vypočtena poměrná část SPZ, která je detekována správně a poměrná část SPZ, která je detekována chybně. Na základě průměru těchto dvou veličin přes datovou sadu je možno efektivně porovnávat úspěšnost jednotlivých detektorů.

Kapitola 6

Ověření funkčnosti

Účelem této práce je vytvořit anonymizér SPZ v obrazových datech, který na svém vstupu přijme adresář s fotografiemi nafocenými pro službu Panorama společnosti Seznam.cz a.s.. Na výstupu potom poskytne nový adresář s odpovídajícími fotografiemi, na kterých je zajištěna nečitelnost SPZ vyfocených vozidel.

V rámci výzkumné části této práce bylo analyzováno, jaký vliv mají jednotlivé parametry použitých neuronových sítí na úspěšnost lokalizace a anonymizace hledaných SPZ. Dále pak bylo zkoumáno, zda je vůbec možné data načtená a zpracovaná v rámci jedné neuronové sítě využít jako vstup jiné neuronové sítě. Tedy zda-li si neuronová síť trénovaná k jednomu typu klasifikace zachovává dostatek informací využitelných k jinému typu klasifikace.

V této části práce jsou nejprve popsána využitá data (sekce 6.1), jejich vlastnosti a způsob jakým byla využita. Dále jsou potom v části 6.2 popsány metody, kterými bylo implementované řešení testováno a jaké hardwarové zdroje byly k vyhodnocení využity. V části 6.3 jsou nakonec shrnuty dosažené výsledky.

6.1 Data

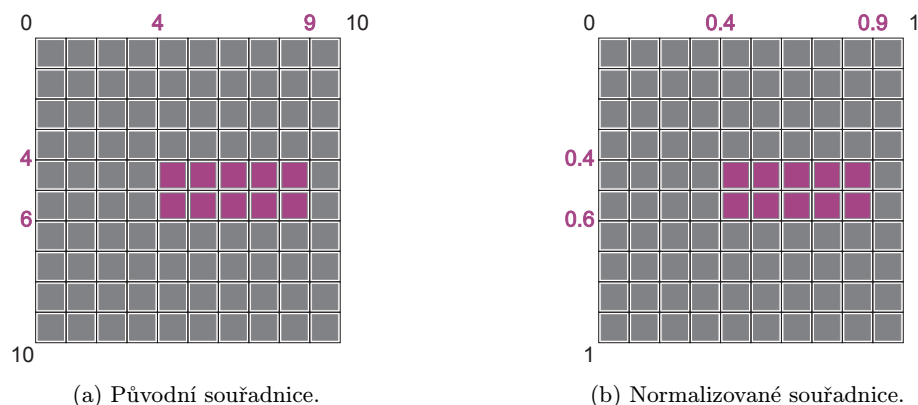
Stěžejní částí této práce byla implementace a učení hluboké neuronové sítě pro detekci automobilů a neuronové sítě pro lokalizaci SPZ. Ověření funkčnosti se tedy týkalo hlavně přesnosti a rozlišovací schopnosti získaných neuronových sítí.

Vstupní datová sada po předzpracování čítala 46 102 výsečí, které byly ručně klasifikovány do tříd `auto` a `neauto`. V reálných datech se automobil nachází pouze na 1.8 % obrázků, pro trénování však musela být datová sada upravena tak, aby bylo možné síť efektivně trénovat. Zkoumáno bylo učení s rovnovážným poměrem pozitivních a negativních vzorků. V takovém případě však je pro trénování sítě využito pouze 3.6 % datové sady, což čítá pouze 1659 vzorků.

Do další fáze zpracování – lokalizace SPZ – už přecházely pouze vzorky, které byly v první fázi klasifikovány jako `auto`. V těchto obrázcích byly poté ručně lokalizovány SPZ. Kolem každé SPZ byl vytvořen hraniční obdélník, určený dvěma rohovými body. Část obrázku nacházející se uvnitř tohoto obdélníku je pro účel této práce bez ohledu na deformaci a zkreslení označována jako SPZ.

V rámci dalšího zpracování poté byly porovnávány dvě různé formy využití takto získaných souřadnic:

Čtveřice souřadnic – tato varianta zahrnuje využití souřadnic ve stavu, v jakém byly zís-



Obrázek 6.1: Normalizace souřadnic SPZ v obraze.

kány. Cílem neuronové sítě je podat na výstupu čtveřici souřadnic náležící do požadovaného rozsahu jak pro x , tak pro y . Protože splnění těchto omezení je pomocí neuronové sítě obtížně dosažitelné, byla zvolena strategie normalizace souřadnic z intervalů $(\langle 0, 227 \rangle, \langle 0, 332 \rangle)$ do intervalů $(\langle 0, 1 \rangle, \langle 0, 1 \rangle)$, jak je ilustrováno na obrázku 6.1.

Maska SPZ – tato varianta poskytuje jednu zásadní výhodu, kterou je možnost detekce libovolného množství SPZ v rámci jedné výseče. Cílem sítě je podat matici o rozměru vstupního obrázku, která pomocí masky odděluje místa, na kterých se nachází SPZ od míst, na kterých se SPZ nenachází. Narozdíl od čtveřic souřadnic, jejichž počet musí být limitován před navržením architektury sítě, v případě masky SPZ je výstup vždy stejné velikosti, bez ohledu na počet lokalizovaných SPZ. Hodnoty, kterými byly části obsahující a neobsahující SPZ v rámci trénovacích dat rozlišovány byly vybrány vždy s ohledem na použitou aktivační funkci.

Sigmoida – v případě použití aktivační funkce sigmoida byly pixely výseče, na kterých se SPZ nachází v cílové masce nahrazeny hodnotou 1, zatímco ostatní pixely byly zastoupeny hodnotou -1 .

ReLU – v případě využití lineárního usměrňovače byly pixely uvnitř oblasti s SPZ v masce zastoupeny hodnotou 255, zatímco ostatní pixely byly zastoupeny hodnotou 0.

V rámci výsečí o již zmíněném rozměru 255×332 se nacházely automobily s SPZ v různých vzdálenostech (a tedy i velikostech a deformacích) od snímacího zařízení. Celkem bylo zaznamenáno 925 SPZ v 830 obrázcích. Jejich velikosti se pohybovaly v rozsahu zaznamenaném v tabulce 6.1.

Získaná a předzpracovaná data byla pro každou část zpracování rozdělena na trénovací, validační a testovací množinu. Zvolený poměr těchto množin byl $0.6 : 0.2 : 0.2$ a jsou vzájemně disjunktní. Důležitým faktorem také je správné promíchání vzorků v rámci množiny. Toto bylo voleno vždy nejrovnoměrnejším možným způsobem a realizováno pomocí jednoho z pomocných skriptů.

	Výška SPZ	Šířka SPZ
Průměr [px]	26	63
Medián [px]	17	46
Modus [px]	9	31
Maximum [px]	202	330
Minimum [px]	3	1

Tabulka 6.1: Velikosti SPZ v obrazovém materiálu pro službu Panorama.

6.2 Metoda testování

Celé testování proběhlo na serverovém stroji společnosti Seznam.cz pod systémem Ubuntu 16.04. Většina výpočtů byla prováděna na grafické kartě GeForce GTX 690 pomocí nástroje CUDA 7.5. Překlad kódu knihoven v jazyce C/C++ byl prováděn pomocí překladače GCC 5.3. Interpretace zdrojového kódu v jazyce Python potom byla prováděna pomocí interpretu Python 2.7.

6.2.1 Spouštění

Pro problémy, v jejichž výsledcích nehraje roli náhoda, byly pro každé nastavení výsledky získány z jedinného běhu. Jedná se o neuronovou síť pro detekci automobilů, která vychází z naučené sítě pro klasifikaci obrázků ImageNet, v celém procesu učení tedy není využito generování pseudonáhodných čísel.

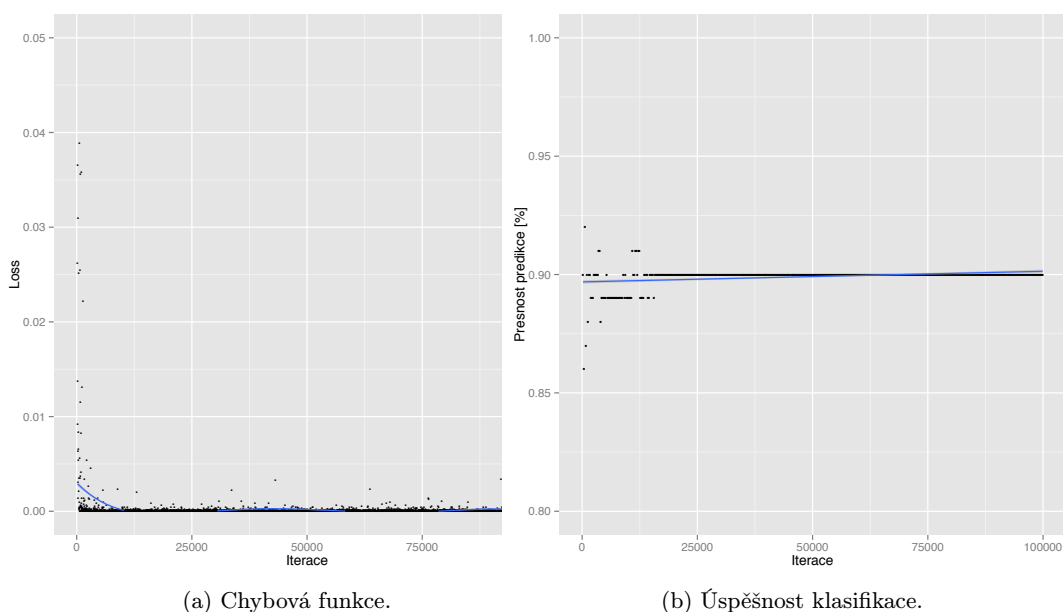
Pro každou variantu nastavení, jejíž výsledky závisí na počáteční inicializaci hodnot byl výpočet spuštěn třikrát a výsledné hodnoty byly vypočteny jako průměr z jednotlivých běhů. Toto se týká neuronové sítě pro lokalizaci SPZ, která na počátku využívá pseudonáhodné inicializace vah. Tato inicializace je prováděna pomocí náhodného generátoru Mersenne Twister z knihovny numpy s využitím rovnoměrného rozložení.

6.3 Dosažené výsledky

V této části práce jsou shrnuty dosažené výsledky implementovaných neuronových sítí pro dvě části práce – pro detekci automobilů a lokalizaci SPZ. Protože tyto části na sebe ve výsledném programu navazují, jejich úspěšnost je třeba chápat jako celek. Část výsledků detekce automobilů vstupuje do lokalizace SPZ jako vstup a úspěšnost detekce automobilů je tedy spíše jakýmsi mezivýsledkem než faktem podávajícím celkový obraz o úspěšnosti klasifikace.

6.3.1 Detekce vozidel

K trénování neuronové sítě pro detekci vozidel byla využita čtrnáctivrstvá konvoluční neuronová síť vycházející ze sítě vzniklé v rámci každoročního contestu ILSVRC. Do této sítě vstupovaly všechny výseče vytvořené ze zpracovávané množiny panorama obrázků. Každá z těchto výsečí byla pomocí vzniklé sítě klasifikována do jedné ze dvou tříd - `auto` nebo `neauto`.



Obrázek 6.2: Vývoj parametrů v průběhu učení.

Průběh učení

Na obrázku 6.2a je uveden vývoj hodnoty chybové funkce v průběhu učení neuronové sítě k detekci vozidel. Na obrázku 6.2b je uveden vývoj přesnosti klasifikace na trénovací sadě.

Graf uvedený na obrázku 6.3 potom ukazuje, jak se v průběhu stejného učení měnila úspěšnost klasifikace na testovací sadě. Procento správně klasifikovaných vzorků je vypočteno jako poměr správně klasifikovaných vzorků jedné třídy ke všem vzorkům náležícím dané třídě.

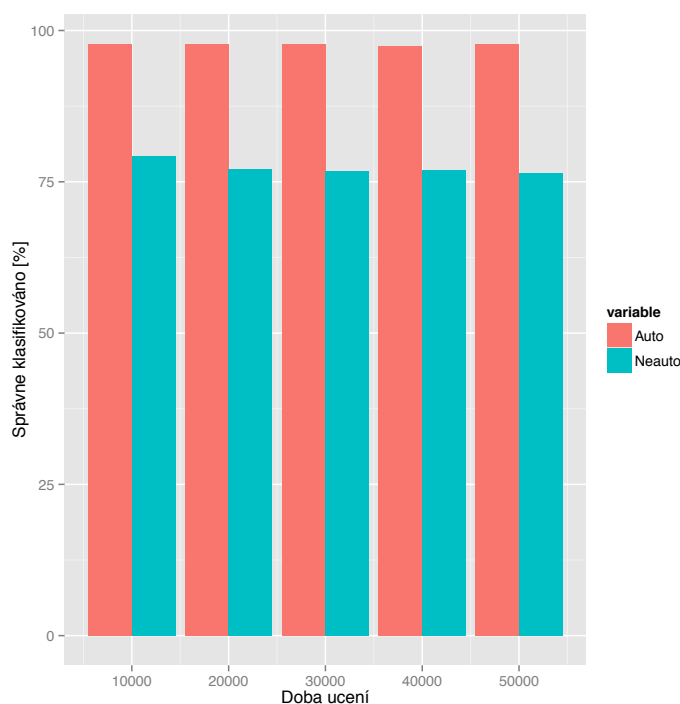
Zatímco na trénovací sadě je i při velkém počtu iterací stále znatelný mírný pokles chybové funkce a nárůst hodnoty přesnosti, na testovací sadě jsou výsledky s velkým počtem iterací stále velmi podobné, nepatrně se zhoršující. Během prvních 10 000 kroků měla přesnost na testovací množině zvyšující se tendenci. Poté již docházelo k přeučování sítě na trénovací množinu a zhoršování výsledků na testovací sadě.

6.3.2 Lokalizace SPZ

Síť pro lokalizaci SPZ měla za úkol lokalizovat oblast obrázku, ve kterém se nachází SPZ. Protože se s navrženým přístupem nedařilo dosáhnout očekávaných výsledků, v rámci trénování této sítě bylo na trénovací množině provedeno několik optimalizací. Výsledky optimalizovaných variant trénovací množiny jsou popsány v této části.

FC7

Jak je uvedeno v kapitole 4 – Návrh, záměrem této práce bylo využít výstup vrstvy FC7 neuronové sítě pro detekci automobilů a využít jej jako vstup jednoduché regresní neuronové sítě, na jejímž výstupu budou získávány souřadnice SPZ ve výšce obrázku. S tímto přístupem však nebylo dosaženo uspokojivých výsledků, proto bylo přistoupeno k dalším optimalizacím.



Obrázek 6.3: Vývoj úspěšnosti detektoru vozidel na testovací množině.

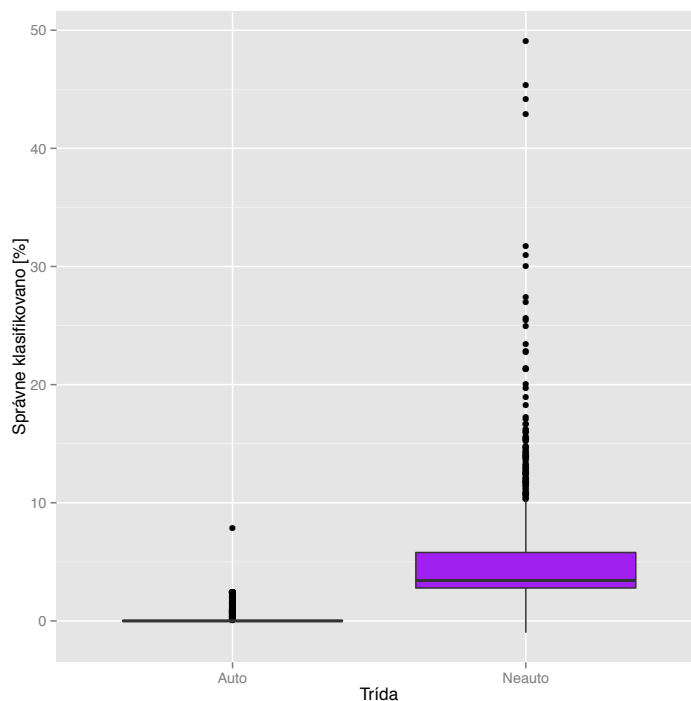
Výsledky sítě, která byla trénována na plnění těchto podmínek, jsou znázorněny na obrázku 6.4. Z obrázku je zřejmé, že výsledky zdaleka nejsou uspokojivé. Pokud se však podíváme na výstupy sítě blíže, jednoduše zjistíme, že zásadní chybou je, že síť podává bez ohledu na vstup stále stejné výsledky. Občasná podobnost výstupů sítě s očekávanými výstupy je tedy způsobena tím, že SPZ se v trénovací množině nejspíše často nacházejí v podobné části výšece. Tento fakt tedy lze považovat za přeučení sítě na konkrétní podmnožinu trénovacích vektorů.

Jako optimalizace tohoto přístupu tedy byla zvolena normalizace souřadnic SPZ v trénovací množině. Jak je uvedeno v části 6.3.1, souřadnice byly normalizovány do intervalu $\langle 0, 1 \rangle$ na obou osách, což do omezení pro přípustné hodnoty výstupu neuronové sítě vneslo mnohem jasnější podmínky. Síť se však potýkala s problémem přeučování i v případě normalizovaných souřadnic. Oproti předchozí variantě už nepodává stejné výstupy pro všechny testovací vzorky, nicméně výstupní hodnoty jsou si stále všechny velmi podobné a tedy nelze je považovat za korektní reakci na podaný vstup.

CONV5

Po neúspěchu s lokalizací SPZ pomocí výstupu vrstvy FC7 bylo přistoupeno k další z optimalizací – využití výstupu vrstvy CONV5, jejíž struktura je popsána výše. Tady už se jedná o složitější neuronovou síť, která pracuje nad maticovými daty. K tomuto kroku bylo přistoupeno ze dvou důvodů:

1. Úspěšnost s původně navrženou variantou FC7 nebyla přesvědčivá.
2. Ve výšece z Panorama fotografie se mnohdy nacházelo více čitelných SPZ, s čímž



Obrázek 6.4: Výsledky pro lokalizaci SPZ pomocí souřadnic.

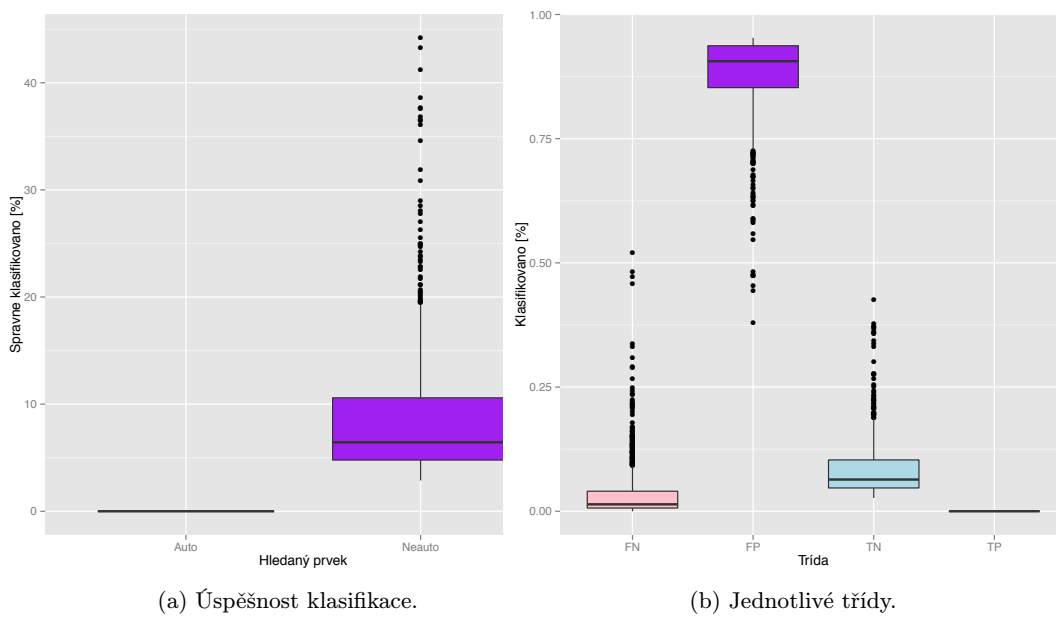
v původním návrhu vzhledem k určené velikosti SPZ a rozlišení fotografie nebylo počítáno. Více SPZ pomocí neuronové sítě s původně navrženou architekturou nelze detekovat. S nově navrženou architekturou sítě, na jejímž výstupu je maska výseče, lze detekovat libovolně velké množství SPZ.

Protože v okamžiku návrhu nové konvoluční neuronové sítě pro lokalizaci SPZ nebylo jasné, jaká aktivační funkce by pro tuto problematiku byla nejvhodnější, byly vybrány dvě funkce - sigmoida a ReLU, jejichž výsledky byly na dané problematice porovnány. Hodnotám na výstupu těchto funkcí byla samozřejmě přizpůsobena i trénovací množina, jak je uvedeno v části 6.1.

První z testovaných aktivačních funkcí byla sigmoida. Úspěšnost lokalizace SPZ vozidel pomocí neuronové sítě, ve které byla využita tato aktivační funkce je znázorněna na obrázku 6.5a. Hodnoty uvedené v grafu byly vypočteny jako poměrná část správně klasifikovaných pixelů dané třídy oproti všem pixelům dané třídy. Tato metoda vyčíslení nezohledňuje vzdálenost chybně klasifikovaného pixelu od zbytku pixelů náležících do stejné třídy, proto s ní logicky velmi pravděpodobně nebude dosaženo výsledků blížících se 100% ani v případě dobře fungujících sítí.

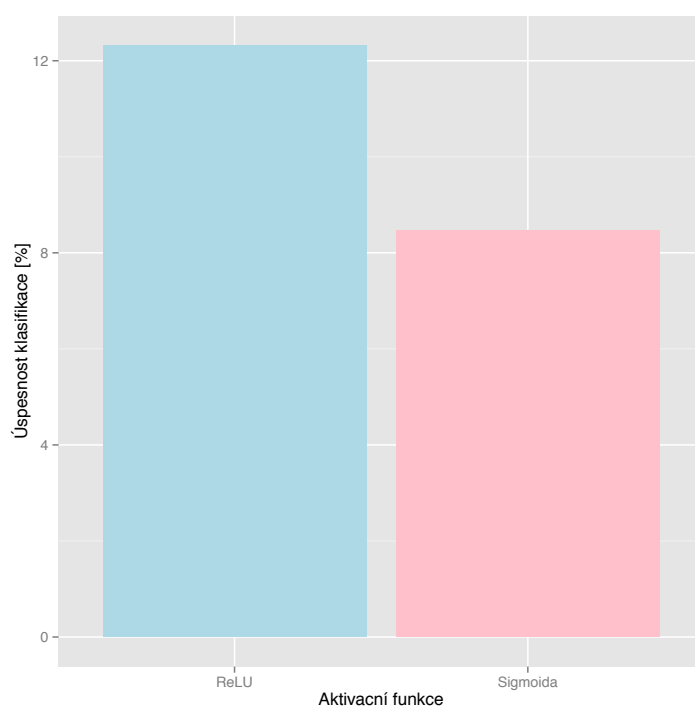
Z obrázku 6.5a plyne, že třída **Auto** nebyla pomocí této neuronové sítě vůbec klasifikována. Pro objasnění původu tohoto faktu je na obrázku 6.5b znázorněno rozložení celého spektra klasifikace. Zde je patrné, že největší částí výstupních hodnot jsou falešná pozitiva, která byla v úvodu této práce pojmenována jako největší problém stávajícího řešení, proto je tento způsob klasifikace nevyhovující.

Porovnání celkové úspěšnosti lokalizace SPZ pomocí obou aktivačních funkcí je znázorněno na obrázku 6.6. Z obrázku je zřejmé, že aktivační funkce ReLU dosahovala o něco



Obrázek 6.5: Výsledky pro aktivační funkci sigmoida.

vyšší úspěšnosti, než aktivační funkce Sigmoida.



Obrázek 6.6: Porovnání úspěšnosti lokalizace pro zvolené aktivační funkce.

Kapitola 7

Závěr

Účelem této diplomové práce bylo vytvořit anonymizér SPZ vozidel optimalizovaný pro službu Panorama společnosti Seznam.cz. Aktuálně využívaný anonymizér vykazuje velkou chybovost a také výpočetní náročnost.

V rámci této práce jsem prostudovala principy a metody využívané pro detekci a anonymizaci SPZ vozidel zachycených na fotografiích a nástroje využívané v současnosti. Dále jsem také analyzovala, které problémy současná řešení dokáží zpracovávat dobře a které jsou naopak jejich slabými stránkami.

Ve spolupráci se zaměstnanci firmy Seznam.cz jsem analyzovala obrazový materiál pro službu Panorama a způsob jeho sběru. V souvislosti s těmito materiály jsem také nastudovala platnou legislativu ohledně zpracování osobních údajů a manipulace s nimi.

Na základě získaných informací jsem navrhla program umožňující provádět automatizovanou detekci a anonymizaci SPZ v obrazovém materiálu ze služby Panorama s využitím metody založené na hlubokých neuronových sítích. Tento program byl navržen s ohledem na to, aby měl co největší potenciál odbourat problémy, se kterými se potýkají stávající komerční řešení.

Aby se program vyvaroval falešně pozitivní detekci SPZ na místech, kde se nachází běžné nápisy, byla detekce SPZ rozdělena v rámci této práce do dvou kroků: 1. detekce automobilů v obraze a 2. lokalizace SPZ na detekovaném vozidle. Každá z těchto částí byla realizována pomocí samostatné neuronové sítě pracující nad obrázky.

Detekce automobilů je prováděna s využitím frameworku **Caffe** a čtrnáctivrstvé konvoluční neuronové sítě finetunované na danou sadu dat. Tato síť na vstupu načítá obrázky, které klasifikuje do jedné ze dvou definovaných tříd - **Auto** nebo **Neauto**.

Lokalizace SPZ je potom prováděna s využitím frameworku **Keras** a jednoduché neuronové sítě obsahující jedinou konvoluční vrstvu. Aby byl minimalizován potřebný výpočetní výkon, síť pro lokalizaci SPZ už nenačítá znovu obrázky, ale využívá předzpracované obrázky z neuronové sítě pro detekci automobilů. Tato metoda již byla dříve ve společnosti Seznam.cz využita u jiných problémů a bylo s ní dosaženo dobrých výsledků. Na výstupu této sítě jsou potom vydávány masky určující, která část obrázku je SPZ a která nikoliv.

V rámci lokalizace SPZ bylo provedeno několik optimalizací s cílem vylepšit dosažené výsledky. Ačkoliv došlo k mírnému zlepšení výsledků lokalizace SPZ, ani po nasazení optimalizací nebyly výstupy uspokojivé.

Nepříznivé výsledky této práce mohou být způsobeny velikostí datové sady určené pro trénování neuronové sítě pro lokalizaci SPZ. Ačkoliv se datová sada obsahující 46 000 obrázků ze služby Panorama zpočátku zdála dostatečně robustní, po ruční klasifikaci a lokalizaci SPZ v těchto obrázcích se ukázalo, že dobře viditelná SPZ se nachází na pouhých 830

obrázcích, což pro naučení složitější neuronové sítě nemusí být dostačující.

Jako možné pokračování této práce navrhuji rozšíření datové sady, případně experimentování s dalšími parametry obou využitých neuronových sítí.

V průběhu řešení této práce jsem se setkala s využitím neuronových sítí na reálných datech a s důležitostí jejich pečlivého předzpracování. To jsem realizovala pomocí několika webových technologií, které mi práci s daty usnadňovaly. Při práci s využitými frameworky jsem pronikla do způsobu práce s optimalizovanými C++ knihovnami pomocí Python wrapperů, což bych chtěla dále rozvíjet. Dále jsem se potom při zpracování výsledků hlouběji seznámila s metodami statistického zpracování a se širším využitím statistického jazyka R.

Reference

- [1] ImageNET Large Scale Visual Recognition Challenge [online]. [cit. 2016-05-20]
Dostupné z: URL <http://www.image-net.org/challenges/LSVRC/>, 2015.
- [2] Zákon č. 101/2001 Sb., o ochraně osobních údajů a o změně některých zákonů:
Informace, Informatika, eGovernment, Ostrava: Sagit, 2016, ISBN 978-80-7488-052-0, s. 376.
- [3] Eberhart, R. C.; Shi, Y.: Neural network concepts and paradigms. *Computational Intelligence: Concepts to Implementations*, Morgan Kaufman Publ., 2007, ISBN 978-1-55860-759-0, s. 145–196.
- [4] Kolektiv autorů brněnských neurologických klinik: *Učebnice obecné neurologie*. Brno: MU Brno, 2003, ISBN 80-210-3309-6, s. 198.
- [5] Kulkarni, A. D.: *Artificial Neural Networks for Image Understanding*. New York: Van Nostrand Reinhold, 1994, ISBN 0-442-00921-6, s. 210.
- [6] Mousa, D. A.: Canny Edge-Detection Based Vehicle Plate Recognition. *International Journal of Signal Processing: Image Processing and Pattern Recognition*, ročník 5, č. 3, 2012: s. 8, ISSN 2005-4254.
- [7] Nielsen, M.: Neural Networks and Deep Learning [online]. [cit 2016-01-15]
Dostupné z: <http://neuralnetworksanddeeplearning.com>, 2015.
- [8] Paugam-Moisy, H.; Bohte, S.: Computing with Spiking Neuron Networks. *Handbook of Natural Computing*, Springer Berlin Heidelberg, 2012, ISBN 978-3-540-92909-3, s. 335–376.
- [9] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J.: Learning representations by back-propagating errors. *Nature*, ročník 323, č. 6088, 1986: s. 533–536, ISSN 0028-0836.
- [10] Schwarz, J.; Sekanina, L.: *Aplikované evoluční algoritmy*. FIT VUT v Brně, 2006.
- [11] Snyder, J. P.: *Flattening the earth*. Chicago: University of Chicago Press, 1997, ISBN 02-267-6747-7, s. 384.
- [12] Srivastava, N.; Hinton, G.; Krizhevsky, A.; aj.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, ročník 15, 2014: s. 1929–1958, ISSN 1938-7228.
- [13] Talášek, M.: *Mapy.cz Panorama*. Praha: Seznam.cz [interní materiál], 2015.

- [14] Thyssen, A.: Blurring and Sharpening [online]. [cit. 2016-04-30]
Dostupné z: <http://www.imagemagick.org/Usage/blur/>, 2004.
- [15] Zbořil, F.: Úvod. Biologický a umělý neuron, umělé neuronové sítě. Perceptron a Adaline. *Přednášky do předmětu softcomputing*, FIT VUT v Brně, 2015.

Příloha A

Obsah CD

A.1 Technická zpráva

Technická zpráva ve formátu pdf se nachází v adresáři `theory` na přiloženém CD. Zdrojové kódy pro a všechny využití balíčky se nachází v podadresáři `theory/src`. Po případně provedených změnách lze technickou zprávu ve formátu pdf vygenerovat pomocí příkazu `make`.

A.2 Praktická část

Praktická část práce se nachází v adresáři `practice` na přiloženém CD. Ten má následující strukturu:

anonymize - python balík vytvořený v rámci této práce. Obsahuje veškeré skripty, které jsou spouštěny v průběhu anonymizace dodaného adresáře s obrázky určenými k anonymizaci.

data - adresář obsahující data využitá při učení neuronové sítě. Tyto jsou dále členěna do několika podadresářů dle svého typu: `lists`, `imgs`, `crops`, ...

scripts - adresář obsahující další skripty jazyka python, které byly použity pro trénování některé z použitých sítí. Skripty, které se nachází v tomto adresáři nejsou při vlastní anonymizaci volány.

results - adresář obsahující výsledky práce s výslednými neuronovými sítěmi. Obsahuje podadresáře s mezivýstupy jednotlivých kroků výpočtu, jako jsou `fc7`, `conv5`, `coords`, `masks`, ...

Příloha B

Spuštění programu

Spuštění programu se provádí spuštěním hlavního skriptu `anonymize.py`. Tomuto skriptu je možné nastavit několik parametrů, které jsou uvedeny v jeho nápovědě. Pokud je však skript spuštěn bez parametrů, všechny hodnoty budou nastaveny defaultně – anonymizace bude probíhat nad obsahem adresáře `img` a anonymizované obrázky budou po skončení procesu k dispozici v adresáři `img.anonymized`.

B.1 Závislosti

Protože program využívá několika frameworků, je třeba k němu nainstalovat několik závislostí.

B.1.1 Caffe

Instalace frameworku `Caffe` je popsána v dokumentaci¹. Bez tohoto frameworku není možné anonymizér spouštět. Cestu ke složce obsahující framework `Caffe` je třeba zadat hlavnímu skriptu pomocí parametru `-r`. Pro správné fungování frameworku `Caffe` je taktéž třeba nastavit cestu k Python balíku `Caffe` do proměnné prostředí `PYTHONPATH` a cestu ke sdílenému objektovému souboru `_caffe.lib.so` do proměnné prostředí `LD_LIBRARY_PATH`.

B.1.2 ImageMagick

Obrazové operace jsou realizovány pomocí Unixového nástroje `ImageMagick`. Ten je k dispozici ze standardních linuxových repozitářů a musí být na stroji před spuštěním anonymizéru nainstalován.

B.1.3 Keras

Dalším využitým frameworkem je `Keras`. Tento je již připraven uvnitř balíku `anonymize`, nicméně v některých situacích (zejména pak při využití `GCC 5.3` a `CUDA 7.5`) může působit problémy, proto je zde uveden.

B.1.4 CUDA

Tato závislost je nepovinná. V případě, že na využitém stroji není k dispozici grafická karta s technologií `CUDA`, celý program může běžet, pochopitelně o něco pomaleji, čistě na `CPU`.

¹<http://caffe.berkeleyvision.org/installation.html>