

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

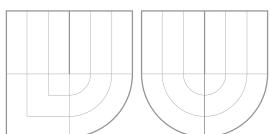
## **ANONYMIZACE SPZ VOZIDEL**

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

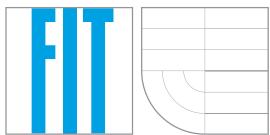
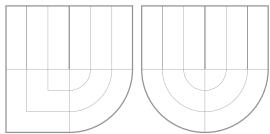
**AUTOR PRÁCE**  
AUTHOR

Bc. BARBORA SKŘIVÁNKOVÁ

BRNO 2016



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

## **Abstrakt**

## **Abstract**

## **Klíčová slova**

Detekce, anonymizace, SPZ, neuronové sítě, hluboké neuronové sítě, 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 poskytl Ing. Adam Kolář ze společnosti Seznam.cz.

.....  
Barbora Skřivánková  
5. ledna 2016

## Poděkování

Na tomto místě bych ráda poděkovala svojí vedoucí, paní Ing. Michaele 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.

© 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

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Anonymizace obrazových dat</b>	<b>3</b>
2.1	Osobní údaje a jejich ochrana . . . . .	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 . . . . .	6
2.4.1	Snímání fotografií . . . . .	7
2.4.2	Ukládání fotografií . . . . .	8
2.4.3	Sešívání fotografií . . . . .	8
2.4.4	Vyrovnávání fotografií . . . . .	8
2.4.5	Namapování na kouli . . . . .	8
<b>3</b>	<b>Neuronové sítě</b>	<b>10</b>
3.1	Biologický základ neuronových sítí . . . . .	10
3.2	Umělý neuron . . . . .	11
3.3	Neuronové sítě . . . . .	13
3.3.1	Topologie sítí . . . . .	14
3.3.2	Učení . . . . .	16
3.3.3	Učení backpropagation . . . . .	17
3.4	Hluboké neuronové sítě . . . . .	19
<b>4</b>	<b>Návrh</b>	<b>20</b>
4.1	Předzpracování dat . . . . .	20
4.1.1	Finetuning . . . . .	21
4.2	Detekce aut v obrázku . . . . .	21
4.3	Detekce SPZ ve vybrané části obrázku . . . . .	22
4.4	Zatemnění SPZ . . . . .	23
<b>5</b>	<b>Závěr</b>	<b>24</b>

# Kapitola 1

## Úvod

Cílem tohoto semestrálního projektu je položit teoretický základ pro následující zpracování diplomové práce na téma anonymizace SPZ vozidel v obrazových datech. Tato práce probíhá ve spolupráci se společností Seznam.cz a její výsledek by měl později sloužit k automatizované anonymizaci obrazových dat pro službu Panorama.

— toto je vykopčené ze závěru: V rámci semestrálního projektu 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.

V průběhu další práce na této diplomové práci bude program popsán v kapitole 4 implementován a otestován na množině dat ze služby Panorama společnosti Seznam.cz. Následně budou vyhodnoceny výsledky a tyto budou porovnány s výsledky stávajících řešení.

## Kapitola 2

# Anonymizace obrazových dat

Anonymizace dat je v dnešní době velmi aktuálním tématem. Velmi intenzivně se na ní pracuje jak na úrovni datové, tak na úrovni anonymizace obrazových dat, která je také předmětem této práce. Pokud je společnost schopna zajistit kvalitní anonymizaci dat, která spravuje, nejen, že tím 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 sekci , dále potom v sekci / popisem pojmu anonymizace dat a v sekci / jsou popsány metody, které se k anonymizaci dat používají v součastnosti.

### 2.1 Osobní údaje a jejich ochrana

Ochrana osobních údajů v České republice představuje soubor práv a povinností, které se vztahují k uchovávání 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., o ochraně osobních údajů a o změně některých zákonů 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 údajů 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 ta, 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 zpracovávání osobních údajů popsat. [citace ze zakona, pod carou: vyklaď z <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, který je snímán, jsou nasnímáni i lidé a další objekty, které se v daném prostranství pohybují. Jak bylo řečeno dříve v této kapitole, veškeré informace, které mohou prozradit jakékoli 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 SPZ vozidel, obličeje 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 může sledovat porovnáním obrázků 2.1 a 2.2. Oba obrázky jsou zpracovány stejným rozostřovacím filtrem. U SPZ na obrázku 2.1 by bylo dekódování původní poznávací značky automobilu

## 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.

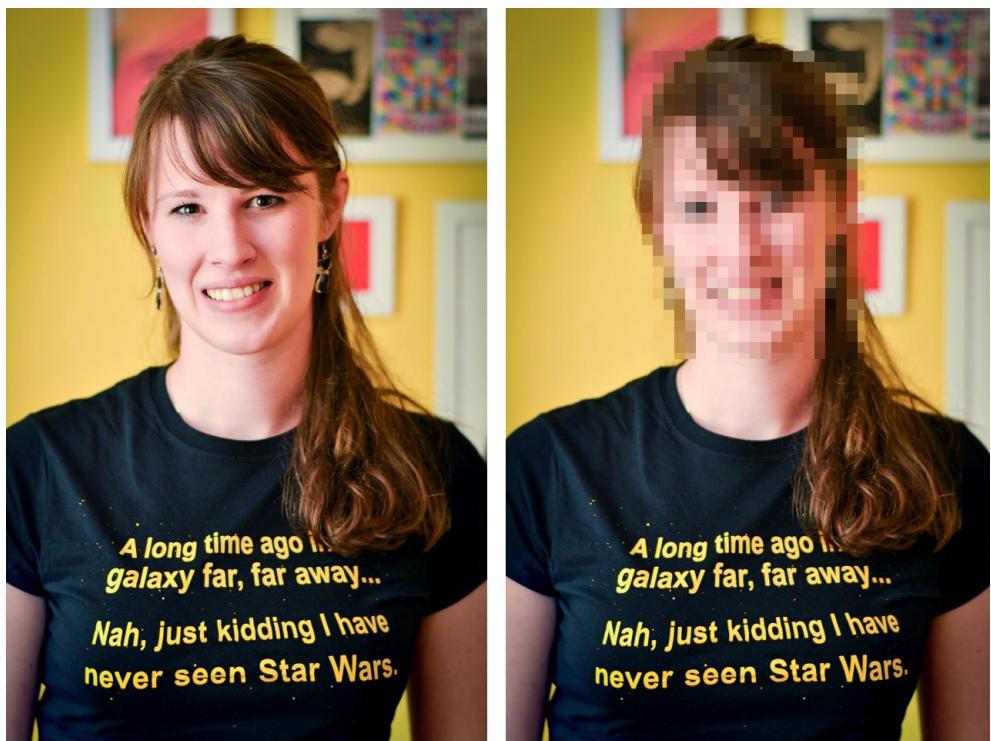
Anonymizace v obrazových datech zatím nedosáhla tak vysoké úrovně a to hned z několika důvodů:

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 datové

Anonymizace obrazových dat sestává ze dvou částí: detekce části obrázku určené k anonymizaci a vlastního procesu anonymizace. Je zřejmé, že vlastní 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.



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 barvou.

### 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 takto anonymizovaného obrázku 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 snímací 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

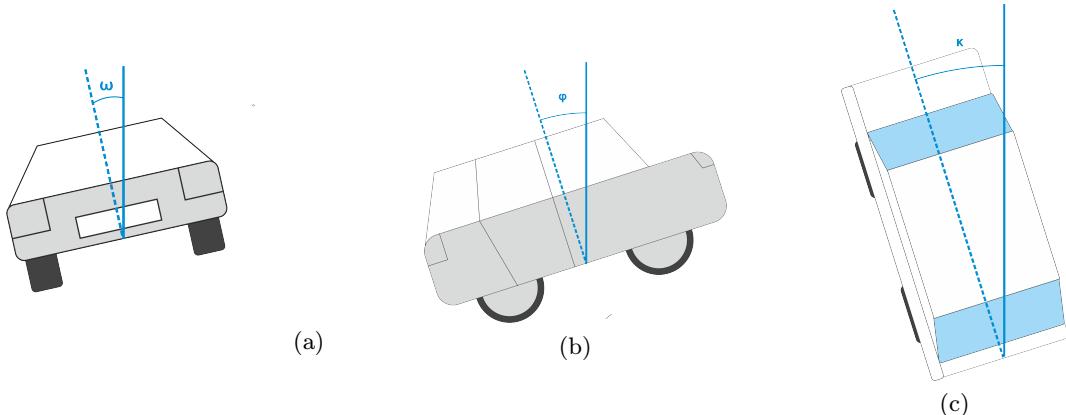
Automatizovaná anonymizace obrazových dat je zatím na počátku svého vývoje.

Eyedea anonymization - to se používá v Seznamu. TODO: zjistit jak to funguje a jaké jsou alternativy.

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 ???. 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 naprostě nežádoucí, aby v rámci automatické anonymizace SPZ vozidel byly anonymizovány i např. nápisy nad vstupy do obchodů a provozoven. V aktuálním stavu automatizované anonymizace je právě tato závada velmi častá. V této práci bude ukázáno, jakým způsobem se těmto chybám vyvarovat.

## 2.4 Obrazová data

Obrazová data byla dodána společností Seznam.cz a.s. V této sekci bude popsán způsob, jakým jsou obrazová data pro službu Panorama snímána, a jak jsou dále zpracovávána než se dostanou ke koncovému uživateli. Anonymizér, který je praktickým výstupem této



Obrázek 2.4: Zaznamenávané úhly při snímání fotomateriálu.

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 15 kamerami rozmístěnými na střeše. Tyto kamery zabírají  $360^\circ$  okolí snímacího automobilu. Způsob, jakým jsou kamery na střeše automobilu umístěny, je znázorněn na obrázku 2.5. Kamery jsou na snímací hlavě umístěny ve dvou řadách. První snímá dolní část výsledné fotografie - v té je umístěno šest kamer. V horní řadě je kamer umístěno 9 a snímají hlavní část obrazu. TODO: zjistit, jestli tam není ještě jedna kamera nahore - to by bylo 6-8-1.

Při snímání se auto pohybuje rychlosťí 70 km/h. Snímání se provádí na všech 15 kamerách současně každé 3 metry. Aby bylo možné později fotografií co nejvíce přiblížit tomu, jak prostředí vnímá člověk, který se v něm pohybuje, při každém snímku se zaznamená ještě sada metadat. Tyto zahrnují

- číslo snímku
- celek (15 fotografií), do kterého snímek patří a poloha v něm
- datum snímání
- zeměpisná poloha snímacího auta v souřadnicích UTM 33
- $\omega$  – úhel náklonu automobilu v pravolevém směru (viz 2.4a)
- $\varphi$  – úhel náklonu automobilu v předozadním směru (viz 2.4b)
- $\kappa$  – odklon směru jízdy automobilu od severního směru, dále označováno pojmem *heading* (2.4c)

Pro výpočet jednotlivých úhlů jsou k dispozici speciální senzory. Pro zaznamenávání headingu jsou na autě dvě vysoko 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.



Obrázek 2.5: Automobil s 15 kamerami umístěnými na snímací hlavě na střeše.

#### 2.4.2 Ukládání fotografií

Při snímání každé 3 metry v rychlosti 70 km/h musí kamery pořizovat fotografie v plném rozlišení v rychlosti 6,48 fps. V tomto okamžiku tedy vzniká problém obrovského množství dat, která je třeba okamžitě zpracovávat. Ještě předtím, než jsou fotografie ve dvou kopíích uloženy na jeden ze čtyř 600GB 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 kufru automobilu pomocí technologie *CUDA*.

Veškeré operace nad daty jsou prováděny v kufru automobilu, kam data ze střechy putují přes vysokorychlostní datové kabely. Každá z 15 kamer je s počítačem spojena pomocí své vlastní 1 Gbitové 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. Pomocí nástroje navrženého ve společnosti Seznam.cz je z 15 fotografií zachycených v jednom okamžiku vytvořena jediná širokoúhlá fotografia, nad kterou se nadále pracuje. V rámci sešívání fotografií jsou také odstraněny části auta, které se dostaly do záběru.

#### 2.4.4 Vyrovnaný fotografií

Na fotografií seštité z 15 fotek je značně deformován horizont. Nyní tedy přichází na řadu vyrovnání horizontu zachyceného prostředí.

Toto se provádí...jak se to dělá? :D

Vzor výstupu z této operace je znázorněn na obrázku 2.6

#### 2.4.5 Namapování na kouli

Výsledný obrázek je následně namapován na kouli, na které už je zobrazován koncovému uživateli. Natočení této koule v prostoru dokáže dorovnat zakřivení, které vznikne vlivem snímání pod úhlem. V tomto okamžiku jsou využity zaznamenané úhly  $\omega$  a  $\varphi$ , jejichž aplikace zamezí uživatelskému pocitu *padání* obrazu.



Obrázek 2.6: 360° fotografie po vyrovnání horizontu.

### Ekvirektangulární projekce

Pro namapování obdélníkového obrazu na kouli je využita ekvirektangulární projekce (v literatuře též označována jako ekvidistantní projekce). Tato projekce využívá přepočet souřadnic dle vztahů 2.1 a 2.2, kde  $x$  a  $y$  jsou souřadnice na fotografii v rovině,  $\lambda$  značí zeměpisnou délku od centrálního poledníku v daném zobrazení,  $\varphi$  značí zeměpisnou šířku a  $\varphi_1$  je parametr rovnoběžek, který určuje rozsah skutečné projekce.

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

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

# Kapitola 3

## Neuronové sítě

Neuronové sítě jsou jednou ze tří hlavních podskupin vědního oboru soft computing. 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.

### 3.1 Biologický základ neuronových sítí

Při studiích v oborech biologie a biofyziky bylo během posledních několika dekád 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*. Hlavní součásti neuronu jsou *dendrity*, *axony* a *tělo buňky (soma)*, jak je znázorněno na obrázku 3.1. Těchto neuronů se v lidské nervové soustavě nachází cca  $10^{11}$ .

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

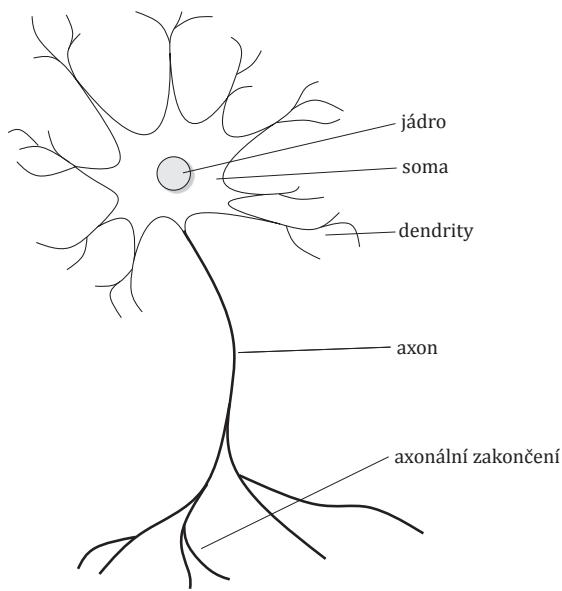
**Axon** – Část neuronu, která zajišťuje přenos vzruchů. Je ukončena *axonálním zakončením*, které je výstupním bodem neuronu a nachází se velmi blízko dendritu jiného neuronu.

**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.

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

**Soma** – tělo buňky. Sleduje, zda-li suma potenciálů přicházejících z jednotlivých dendritů nepřesáhla dný prah 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 impulzu 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 prah, soma



Obrázek 3.1: Schéma biologického neuronu. Na obrázku je zobrazeno *jádro*, které uchovává genetickou informaci buňky, *soma* je tělo buňky, *dendrity* jsou vstupní body neuronu, *axon* jsou výstupní výběžky tvořené axonovými vlákny a *axonální zakončení* jsou výstupní místa neuronu.

vygeneruje nový impulz a tím se vznich 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 0,1-100Hz.

### 3.2 Umělý neuron

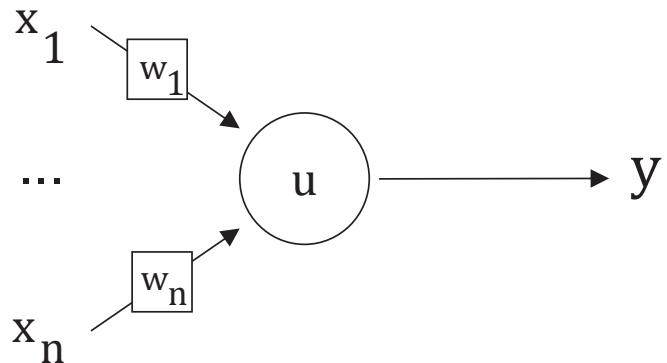
Princip biologického neuronu je napodobován umělým neuronem. Základní model umělého neuronu je zobrazen na obrázku 3.2, kde

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

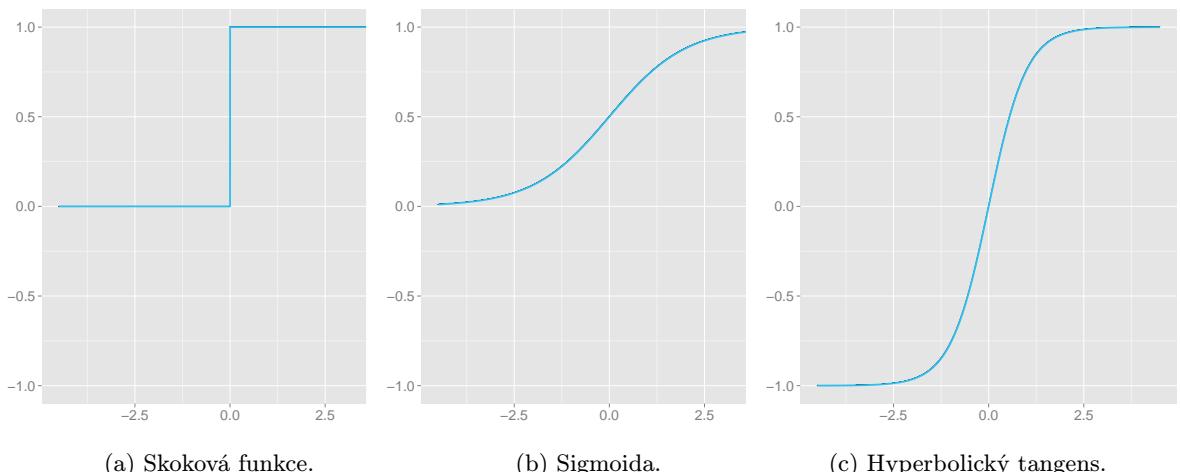
$w_i$  je váha propojení. Určuje jak moc bude konkrétní vstup  $i$  ovlivňovat výstup neuronu. Váhy se v průběhu výpočtu mění - probíhá *učení*, případně přes neměnnou váhu může být připojen tzv. *bias*.

$u$  je hodnota vnitřního potenciálu neuronu.

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  $< -1; 1 >$ , případně  $< -2; 2 >$ ), dále jsou nad nimi provedeny bázová 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.



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.



(a) Skoková funkce.

(b) Sigmoida.

(c) Hyperbolický tangens.

Obrázek 3.3: Nejčastěji využívané aktivační funkce.

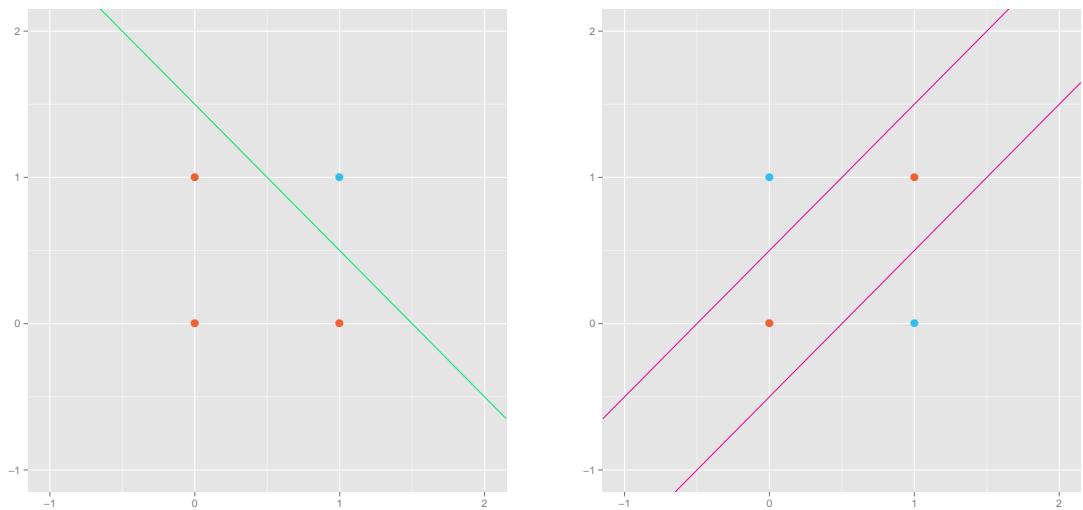
**Bázová funkce** - určuje způsob výpočtu vnitřního potenciálu  $u$  neuronu. Nejčastěji využívaná je lineární bázová funkce. Její vztah pro výpočet vnitřního potenciálu je uveden ve vztahu 3.1. Další z využívaných bázových funkcí je radiální bázová funkce - vztah 3.2.

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

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

**Aktivační funkce** - též nazývaná *přenosová funkce*, využívá se pro ni symbol  $\sigma$ . Určuje způsob výpočtu výstupu neuronu z jeho vnitřního potenciálu. Tři nečastěji 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.

Jednoduchý umělý neuron s jedním biasem a skokovou aktivační funkcí je v literatuře označován jako *perceptron*. Perceptron byl poprvé představen Frankem Rosenblattem v



(a) Lineárně separovatelná funkce AND.

(b) Lineárně neseparovatelná funkce XOR.

Obrázek 3.4: Problém lineární separovatelnosti.

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 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 Neuronové sítě

Neuronová síť je výpočetním systémem, o kterém může být uvažováno jako o *černé skřínce*, 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, jako například:

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

**Hledání vzorů** - na vstup je přiložen vzor a na výstupu je vygenerován výstup, který nejlépe odpovídá vzoru poskytnutému na vstupu.

**Doplňování vzorů** - na vstup je přiložen vektor znázorňující část hledaného vektoru. 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 zoptymalizované řešení.

**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.

V této práci se budu zabývat využitím klasifikačních neuronových sítí ke komprimaci dat a dále pak k hledání vzorů v získaných komprimovaných datech.

### 3.3.1 Topologie sítí

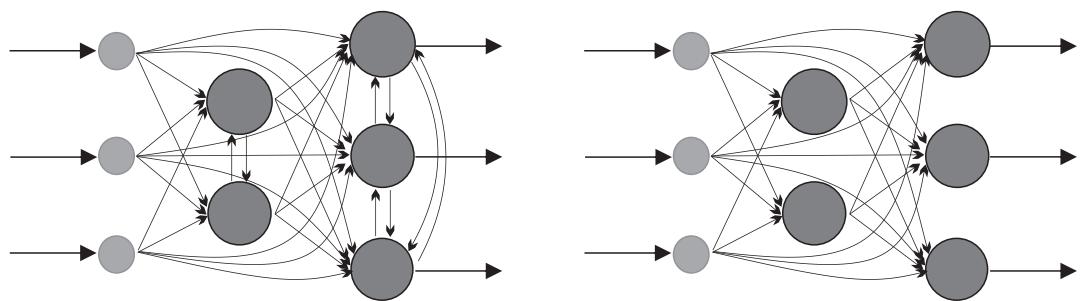
Aby bylo možné pomocí neuronových sítí řešit i lineárně neseparovatelné problémy, je třeba podstatně zesložitit jejich architekturu. To se dělá vkládání dalších vrstev neuronů do sítí. Důležitým faktorem přitom je, aby jednotlivé vrstvy neuronové sítě měly nelineární aktivační funkci. Pokud by totiž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

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. Plně propojená síť, ve které je každý neuron propojen se všemi neurony svojí vrstvy, vrstev předchozích i následujících se postupně vyvinula ve vrstvovou síť (viz 3.5a). Ve vrstvové 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 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.

### Vícevrstvé sítě

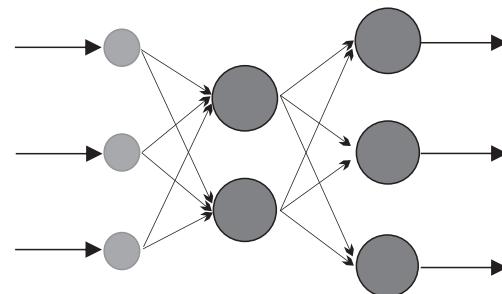
Vícevrstvá neuronová síť má více než dvě vrstvy neuronů. První vrstva se nazývá *vstupní vrstva* a nad načtenými vstupy neprovádí žádnou operaci. Do celkového počtu vrstev se vstupní vrstva v literatuře většinou nezapočítává. Poslední vrstva se nazývá *výstupní vrstva*. Tato vrstva se do celkového počtu 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.



(a) Vrstvová síť.

(b) Acyklická síť.



(c) Dopředná síť.

Obrázek 3.5: Vícevrstvé neuronové sítě.

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 [3.3.3](#).

### 3.3.2 Učení

Nejdůležitější vlastnosti neuronových sítí je pochopitelně schopnost najít řešení daného problému. Toto hledání řešení - *učení* - probíhá prostřednictvím změny vah jednotlivých propojení tak, aby neuronová síť podávala co nejpřesnější výsledky pro daný problém. Stejná neuronová síť na stejný vstup podá vždy stejný výstup. Nejobecnějším dělením algoritmů pro učení je na učení s učitelem a bez učitele.

**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 známymi během učení. Algoritmy učení bez učitele se snaží zorganizovat trénovací data a nalézt v nich co nejspecifitější společné vlastnosti.

Dále však lze učení dělit podle dalších kritérií, dalším z důležitých 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 přetrénována. 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. Nejznámějším zástupcem těchto metod učení je algoritmus *backpropagation*.

**On-line učení** je dynamičtější metoda učení. Pokud je do tímto způsobem učené síť 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 však dnes 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.

### 3.3.3 Učení backpropagation

Když byl algoritmus backpropagation v roce 1986 představen, 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.

Pro ilustraci práce backpropagation nyní zavedeme 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

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

Kde  $\sigma$  je jeho aktivační funkce a  $k$  jde přes všechny neurony z vrstvy  $l - 1$ .

Pro každou vrstvu  $l$  je dále definovaná váhová matice  $w^l$  (viz vztah 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  $\sigma$ , kdy platí vztah 3.5.

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

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

Se zavedeným zápisem už lze vztah pro výpočet výstupních hodnot vypočít 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 vztah 3.6.

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

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

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

Chybu sítě pro aktuální vstupní vektor vypočteme podle vztahu pro kvadratickou chybu sítě(viz 3.8), kde  $d(x)$  je očekávaný výstup sítě pro vstup  $x$ ,  $y^L(x)$  je koncový výstup sítě pro

vstup  $x$ . Pro celou množinu vstupů se potom využívá vztah 3.9, podle kterého vypočteme průměrnou chybu pro všechny vstupní vektory.

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

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

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 ve vztahu 3.10.

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

V procesu učení backpropagation tedy provádíme malé změny  $\Delta u_j^l$  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  $\Delta w_{jk}^l$  vypočtena podle vztahu 3.11, kde  $\delta$  je vypočtena podle vztahu 3.12 a  $\eta$  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 do 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.

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

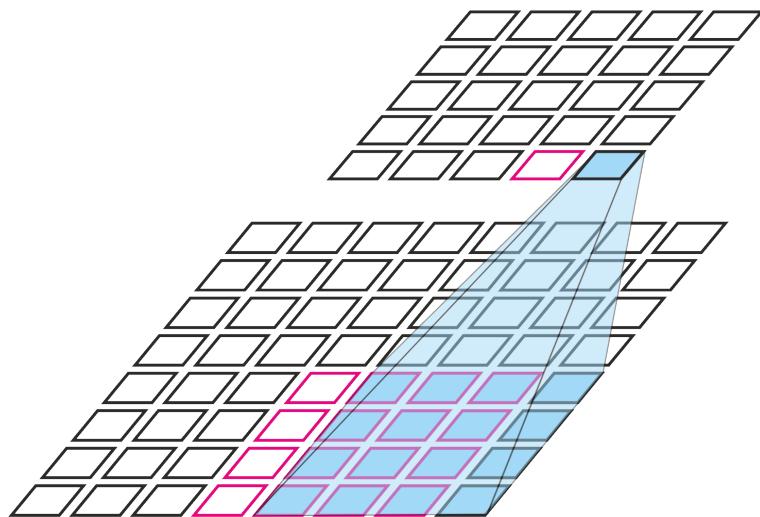
$$\delta^L = (y^L - d) \odot \sigma'(u^L) \quad (3.14)$$

Výpočet  $\delta^l$  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  $\Delta w_{jk}^l$ , 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 Hluboké neuronové sítě

V předchozích kapitolách této práce byly pojmem neuronové sítě vždy rozuměny plně propojené sítě. Tedy takové, ve kterých existuje propojení mezi každými dvěma neurony ležícími v sousedních vrstvách. Pokud však neuronová síť pracuje nad obrázky, tímto propojením se ztrácí prostorová informace. U hodnot pocházejících ze dvou sousedních pixelů můžeme pracovat se stejnou závislostí jako u hodnot pocházejících z pixelů z opačných hran obrázků. Pokud vezmeme v architektuře sítě v potaz, že některé pixely obrázku se nachází v bezprostřední blízkosti, zatímco jiné dvojice pixelů spolu vůbec nesousedí, můžeme získat mnohem lepší výsledky.

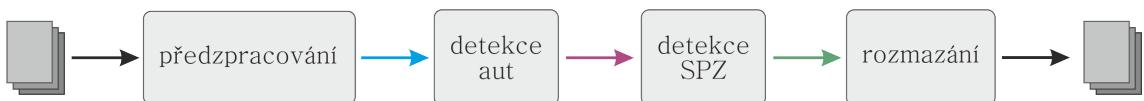


Obrázek 3.6: Schéma konvoluční vrstvy sítě.

# Kapitola 4

## Návrh

V této kapitole je popsán návrh implementovaného programu. Cílem této práce je vyrobit nástroj, který dokáže automaticky zpracovávat nafočené obrázky na vstupu a podávat anonymizované obrázky na výstupu. Práce s obrázky je rozdělena do čtyř základních fází znázorněných na obrázku 4.1, jejichž mechanismy budou v této kapitole podrobně popsány. Barevné šipky v obrázku 4.1 znázorňují jednotlivé mezivýsledky při zpracování a stejná barevná notace je použita i v dalších obrázcích této kapitoly.



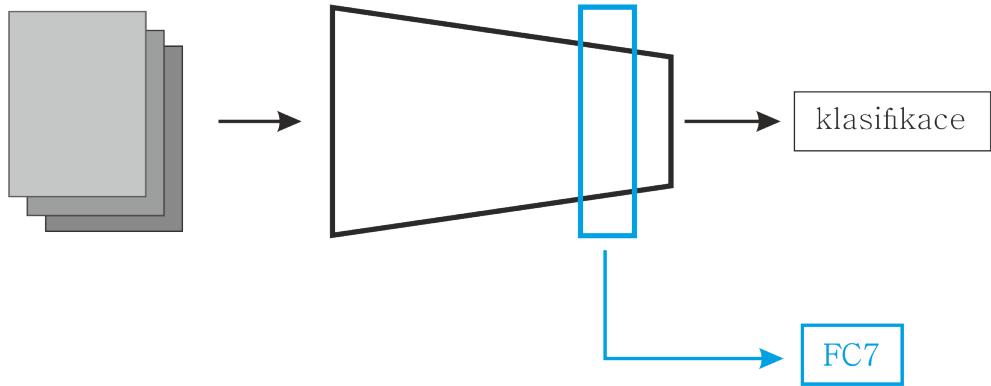
Obrázek 4.1: Základní struktura programu.

### 4.1 Předzpracování dat

Na primárním vstupu celého programu jsou přijímány fotografie nasnímané v rámci sběru dat pro službu Panorama. Všechny další fáze už však pracují nad čísly typu *double*, předzpracování dat tedy spočívá v reprezentaci celého obrázku pomocí číselného vektoru. Tento číselný vektor musí mít pro jednotlivé obrázky co nejlepší rozlišovací schopnost. Pokud je neuronová síť natrénována ke klasifikaci velkých dat do  $n$  tříd, jejím hlavním účelem je vyselektovat hodnoty, které jsou pro určování tříd podstatné - vytvořit tzv. *vektor rysů*. Tento vektor rysů, též *feature vector* je nejkondenzovanější informace z počátečních dat, výstupem z předzpracování dat je tedy právě feature vector.

K extrakci feature vectoru byla v této práci využita již naučená neuronová síť *Ilsvrc*, jež vznikla při příležitosti *ImageNet Large Scale Visual Recognition Challenge 2015* - odtud zkratka *Ilsvrc*. Tato výzva se koná každoročně a je zaměřena na algoritmy pro lokalizaci a detekci objektů v obraze a pak také na klasifikaci scény. Síť *Ilsvrc* byla naučena na množině obrázků *ImageNet*, která obsahuje více než 14 milionů obrázků rozřazených do 1000 kategorií.

Tato síť byla v rámci této práce pomocí *finetuningu* (viz část 4.1.1) optimalizována pro klasifikaci množiny obrázků bližší obrázkům, se kterými budu pracovat. Pro další použití není využíván primární výstup klasifikace (viz obrázek 4.2). Ze sítě je extraheován vektor rysů, jehož nejvíce informačně bohatá varianta se nachází v předposlední vrstvě sítě, která nese název *FC7*.



Obrázek 4.2: Schéma předzpracování dat.

#### 4.1.1 Finetuning

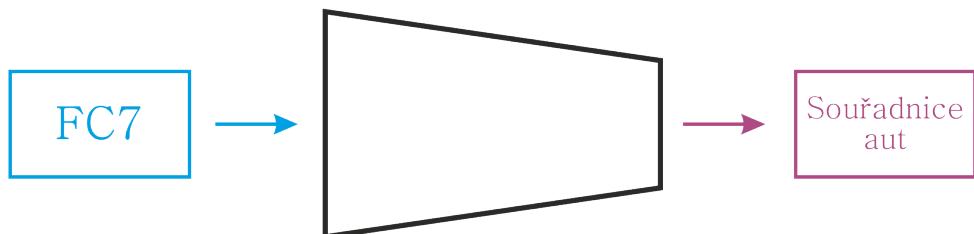
*Finetuning* je metoda využíváná v případě, že už máme úspěšně natrénovanou síť a chceme ji využít ke klasifikaci na podobná data. Standardně se na počátku tréninku neuronové sítě na nový typ vstupu všechny váhy inicializují náhodně. Pokud však máme k dispozici už síť naučenou na velké množině obrázků, můžeme namísto náhodné inicializace nové sítě inicializovat váhy nové sítě na výsledné váhy naučené sítě. Proces učení této sítě se potom nazývá *finetuning*. Ten lze provádět více způsoby. Nejznámějšími z nich jsou:

- Úpravou vah pouze v poslední vrstvě
- Přidáním několika nových vrstev do naučené neuronové sítě

Základním principem finetuningu je to, že do vah v naučené části sítě není příliš zasahováno. V případě úpravy vah pouze v poslední vrstvě jsou ostatní váhy striktně zachovány ve své původní podobě. V případě přidání nových vrstev do naučené sítě je tato celá zachována ve své původní podobě a učení probíhá pouze na nových vrstvách vzniklé sítě.

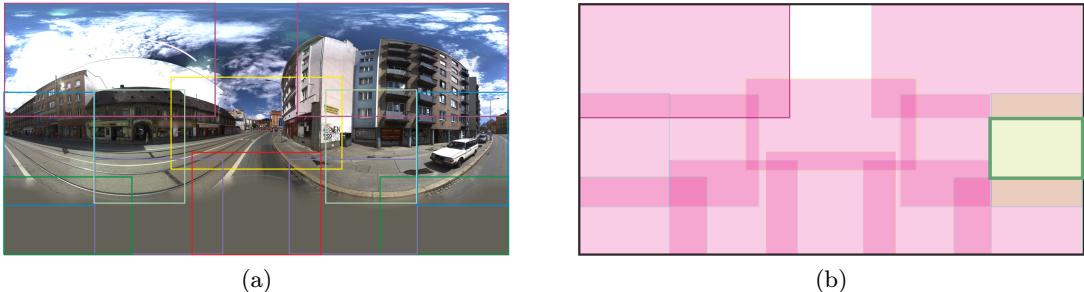
## 4.2 Detekce aut v obrázku

Neuronová síť provádějící detekci automobilů v obrázku je stěžejní částí této diplomové práce. Je to též její hlavní implementační část. Cílem je získat neuronovou síť, jež na vstupu dostane feature vektor vytvořený v rámci předzpracování dat a na výstupu podá informaci, kde se v analyzovaném obrázku nachází automobil.



Obrázek 4.3: Schéma detekce aut v obrázku.

Trénování neuronové sítě pro detekci automobilů bude provedeno na datové množině, která bude vždy obsahovat právě jedno nebo žádné auto. Výstup této sítě tedy bude binární



Obrázek 4.4: Rozdelení obrázku na podčásti a následné vytvoření heatmapy

- na obrázku je auto/na obrázku není auto. V rámci jednoho obrázku se však může nacházet více aut, proto je třeba vytvořit způsob, jakým detekovat všechna auta. Pro tuto práci jsem zvolila techniku, kdy na počátku, ještě před předzpracováním dat dle kapitoly 4.1, bude vstupní fotografie rozřezána na několik podčástí, které budou zkoumány samostatně. Neuronová síť pro detekci automobilů pro každou podčást podá výstup na obrázku je/není auto. Tyto hodnoty jsou potom umístěny zpátky na původní pozice a tím je vytvořena *heatmapa* (viz 4.4). Na obrázku 4.4b můžeme vidět postup tvorby heatmapy: výřezy, ve kterých nebylo nalezeno auto jsou znázorněny červenou barvou, výřezy, ve kterých bylo nalezeno auto jsou znázorněny barvou zelenou. Na místech, kde se překrývá zelená barva se zelenou je nejvyšší pravděpodobnost výskytu auta. Naopak při překryvu červených ploch je pravděpodobnost nejmenší. Při překryvu červené a zelené plochy je pravděpodobnost výskytu auta malá.

Výřez, ve kterém byla zjištěna nejvyšší pravděpodobnost výskytu automobilu je na obrázku zvýrazněn zeleným rámečkem. Vzniklý výřez bude pro další fáze zpracování identifikován čtyřmi souřadnicemi:  $x_1$  a  $y_1$ , které identifikují souřadnice levého horního rohu obdélníku, ve kterém se nachází auto a  $x_2$  a  $y_2$ , které identifikují souřadnice pravého dolního rohu tohoto obdélníku.

### 4.3 Detekce SPZ ve vybrané části obrázku

Další navazující částí této práce je detekce SPZ ve výřezech s pravděpodobným výskytom automobilu určených výstupy předchozího kroku. V této části se budu dvěma možným způsobům, které v rámci této práce budou porovnávány. Způsob, jakým tato část bude pracovat je jednoduše znázorněn na obrázku 4.5, kde v rámci implementace výpočetního bloku budou porovnávány následující možnosti:

**Detektor hran** je sada matematických metod sloužících k hledání výrazných (nespojitých) změn jasu v obrazu. Toto lze provádět buďto na základě vyšetřování funkce jasu a jejích derivací nebo na základě analýzy frekvencí v obrazu. Mezi mnoha představenými hranovými detektory je nejvýraznější *Cannův hranový detektor*, představený Johnem Cannym v roce 1986 ???. Tento detektor je založen na Gaussovském rozostření obrazu. Hrany detekuje pomocí hledání orientace gradientu a směrových derivací. Funguje velmi dobře při výskytu spojitéh a dostatečně silných hran. Výsledný obraz je velmi ostrý, ale ne příliš detailní.

**Neuronová síť** bude nasazena hlavně pro porovnání s detektorem hran. Naučení neuronové sítě samozřejmě bude výpočetně náročnější, nicméně pokud by výsledky byly

výrazně lepší než u konvenčního řešení, bude stále neuronová síť ve výsledku lepším řešením.

Výstupem detekce SPZ ve vybrané části obrázku je opět obdélník, který je identifikován čtyřmi souřadnicemi:  $x_1$  a  $y_1$  identifikujícími levý horní roh SPZ a  $x_2$  a  $y_2$  identifikující pravý dolní roh SPZ.



Obrázek 4.5: Schéma detekce SPZ ve vybrané sekci obrázku

#### 4.4 Zatemnění SPZ

Poslední fází zpracování je namapování detekované SPZ zpět do celého vstupního obrázku bez výrezů a její rozmazání. Souřadnice výřezu, ve kterém byla SPZ vyhledávána jsou známý z předchozího kroku zpracování. Namapování SPZ zpět do souřadnic celé Panorama fotografie tedy bude provedeno prostým součtem souřadnic  $[x_1^{vez} + x_1^{SPZ}, y_1^{vez} + y_1^{SPZ}]$ .

Pro potřeby této diplomové práce bude postačující přes celou SPZ umístit neprůhledný obdélník, který znemožní její čitelnost. Pro využití v praxi však bude nutné zvolit jinou variantu, aby anonymizované SPZ vozidel v Panorama zobrazení nepůsobily příliš rušivě. Nejvhodnější bude použít některou z dostupných rozostřovacích funkcí, například *gausovské rozostření* nebo *rozostření pohybem*.

# Kapitola 5

## Závěr

V rámci semestrálního projektu 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.

V průběhu další práce na této diplomové práci bude program popsán v kapitole 4 implementován a otestován na množině dat ze služby Panorama společnosti Seznam.cz. Následně budou vyhodnoceny výsledky a tyto budou porovnány s výsledky stávajících řešení.

◦