



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

KONVOLUČNÍ NEURONOVÁ SÍŤ PRO
SEGMENTACI OBRAZU

A CONVOLUTIONAL NEURAL NETWORK FOR IMAGE SEGMENTATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Michal Mitrenga

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Václav Jirsík, CSc.

BRNO 2018

ABSTRAKT

Cílem bakalářské práce je seznámit se s problematikou konvolučních neuronových sítí a realizovat segmentaci obrazu. Toto téma v sobě zahrnuje obor počítačového vidění, který je používán v systémech s umělou inteligencí. Zvláštní pozornost je věnována procesu segmentace obrazu. Dále se práce věnuje základním principům umělých neuronových sítí, struktuře konvolučních neuronových sítí a zejména pak popisu jednotlivých architektur sémantické segmentace. Vybraná architektura SegNet je použita v praktické aplikaci spolu s před-učenou sítí. Součástí práce je databáze obrazů CamVid, která je použita pro trénování. Pro testování je vytvořena databáze vlastních snímků. Praktická část je zaměřena na trénování CNN a hledání nevhodnějších parametrů pro učení sítě za pomoci SW Matlab.

KLÍČOVÁ SLOVA

Neuronové sítě, konvoluční neuronové sítě, segmentace, sémantická segmentace, hluboké učení, počítačové vidění, klasifikace

ABSTRACT

The aim of the bachelor thesis is to learn more about the problem of convolutional neural networks and to realize image segmentation. This theme includes the field of computer vision, which is used in systems of artificial intelligence. Special Attention is paid to the image segmentation process. Furthermore, the thesis deals with the basic principles of artificial neural networks, the structure of convolutional neural networks and especially with the description of individual semantic segmentation architectures. The chosen SegNet architecture is used in a practical application along with a pre-learned network. Part of the work is a database of CamVid images, which is used for training. For testing, a custom image database is created. Practical part is focused on CNN training and searching for unsuitable parameters for network learning using SW Matlab.

KEYWORDS

Artificial neural network, convolutional neural network, semantic segmentation, deep learning, computer vision, classification

Bibliografická citace

MITRENGA, M. Konvoluční neuronová síť pro segmentaci obrazu. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 55 s. Vedoucí bakalářské práce doc. Ing. Václav Jirsík, CSc..

PROHLÁŠENÍ

Prohlašuji, že svoji bakalářskou práci na téma Konvoluční neuronová síť pro segmentaci obrazu jsem vypracoval samostatně pod vedením vedoucího doc. Ing. Václava Jirsíka, CSc. a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor bakalářské práce Konvoluční neuronové síť pro segmentaci obrazu dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

(podpis autora)

Obsah

Seznam symbolů, veličin a zkratk.....	7
1 Úvod	11
2 počítačové vidění.....	12
2.1 Snímání obrazu.....	12
2.2 Předzpracování obrazu	13
2.2.1 Bodové jasové transformace	13
2.2.2 Geometrická transformace.....	13
2.3 Lokální předzpracování	14
2.3.1 Filtrace obrazu	14
2.3.2 Detektory hran.....	14
2.4 Segmentace obrazu.....	16
2.4.1 Segmentace prahováním.....	16
2.4.2 Segmentace narůstáním oblastí	17
2.4.3 Sémantické metody segmentace.....	19
2.5 Popis	19
2.6 Klasifikace.....	19
3 Umělá neuronová síť	20
3.1 Princip sítě.....	20
3.2 Matematický model neuronu	21
3.3 Principy učení	22
3.3.1 Učení bez učitele.....	23
3.3.2 Učení s učitelem.....	23
3.4 Zpracování obrazu.....	24
4 Konvoluční neuronové sítě	25
4.1 Struktura konvoluční neuronové sítě.....	25
4.2 Architektura sítě	25
4.2.1 Konvoluční vrstva	26
4.2.2 Přenosová funkce.....	27
4.2.3 Sdružení.....	29
4.3 Učení CNN.....	30
4.4 Princip vybavování.....	30
4.5 Před-učené sítě	30
4.6 Praktické aplikace CNN pro segmentaci obrazu.....	31
4.6.1 Automatické rozpoznání stavu elektroměru.....	31
4.6.2 Detekce mobilního robota zpracováním obrazu	31
4.6.3 Analýza zdravotnických obrazů	32
4.6.4 Segmentace satelitních snímků	32

5	Sémantická segmentace	33
5.1	FCN	33
5.2	SegNet	34
5.3	PSPNet.....	35
5.4	RefineNet	35
5.5	Dilatační konvoluce.....	36
5.6	U-Net.....	37
6	Databáze obrazů.....	38
6.1	Databáze CamVid	38
6.2	Databáze MTech.....	40
7	Implementace sítě	41
7.1	Popis vybrané architektury	41
7.2	Rozdělení skriptu.....	42
7.3	Výpočetní výkon.....	44
7.4	Výsledky učení	44
7.4.1	Vysvětlení pojmů.....	45
7.4.2	Statistiky výsledků učení	46
7.4.3	Segmentované obrazy	49
8	Závěr	51
	Literatura	52

Seznam symbolů, veličin a zkratek

FEKT	Fakulta elektrotechniky a komunikačních technologií
VUT	Vysoké učení technické v Brně
NS	Neuronová síť
UNS	Umělá neuronová síť
CNN	Konvoluční neuronová síť
CONV	Konvoluční vrstva
POOL	Sdružování
GPU	Grafická karta
CPU	Procesor
RAM	Random Access Memory
CV	Computer vision
HW	Hardware
SW	Software
PT	Před-učená
SS	Sémantická segmentace

Seznam obrázků

Obr. 1 Řetězec zpracování obrazu.....	12
Obr. 2 Transformace jasové stupnice.....	13
Obr. 3 Geometrická transformace.....	13
Obr. 4 Příklad filtru pro průměrování.....	13
Obr. 5 Detekce hran.....	14
Obr. 6 Histogram a segmentace pomocí jednoho prahu.....	16
Obr. 7 Metoda spojování oblastí.....	17
Obr. 8 Spojování a štěpení.....	17
Obr. 9 Uspořádání neuronů do vrstev.....	29
Obr. 10 Matematický model neuronu.....	20
Obr. 11 Přenosové funkce.....	21
Obr. 12 Schéma architektury CNN.....	24
Obr. 13 Konvoluce.....	25
Obr. 14 Princip konvoluční vrstvy.....	26
Obr. 15 Sigmoida.....	27
Obr. 16 Tanh(x).....	27
Obr. 17 ReLU.....	27
Obr. 18 Leaky ReLU.....	27
Obr. 19 Zmenšování obrazu.....	28
Obr. 20 Možnosti sdružování.....	28
Obr. 21 Fotografie mobilního robota, pohled od stropu.....	30
Obr. 22 Segmentace satelitního snímku.....	31
Obr. 23 Architektura FCN – plně propojené vrstvy.....	33
Obr. 24 Architektura sítě SegNet.....	33
Obr. 25 SegNet - rozšiřování.....	34
Obr. 26 Architektura sítě PSPNet.....	34
Obr. 27 Ukázka architektury sítě RefineNet.....	35
Obr. 28 Dilatační sdružovací vrstva.....	35
Obr. 29 Architektura sítě U-Net.....	36
Obr. 30 Rozdělení CamVid do 32 kategorií.....	37
Obr. 31 CamVid – RGB.....	38
Obr. 32 CamVid – Labels.....	38
Obr. 33 Vlastní data MTech.....	39
Obr. 34 Statistika počtu pixelu v kategoriích.....	42
Obr. 35 CamVid – segmentace.....	48
Obr. 36 MTech – segmentace.....	49

Seznam tabulek a grafů

Tabulka 1 Parametry použitého Hardwaru.....	43
Graf 1 Rozlišení obrazů.....	46
Graf 2 Závislost počtu Epoch.....	46
Graf 3 Rozložení vah.....	47
Graf 4 Počáteční rychlost učení.....	47
Graf 5 Rozdělení obrazů.....	48

1 ÚVOD

V současné době nás obklopuje velké množství dat, které musíme nějak zpracovat. S tím souvisí rychlost veškerého dění ať už v průmyslu nebo i ve způsobu běžného života. Proto se snažíme vyvíjet nové technologie, které mohou být nápomocné se zpracováním velkého množství dat.

Jednou z těchto technologií je automatické získávání informací o tom, jaké objekty se v obraze nacházejí. Obraz je potřeba segmentovat a vzniklé oblasti klasifikovat. Tyto operace s obrazy se v dnešní době začínají víc a víc využívat. Detekce objektů ve snímku je velmi důležitým prvkem každé operace zahrnující práci s obrazem. Uplatňuje se v různých oblastech od autonomní řízení, přes zabezpečení (př. kontrola, zda je vše na svém místě) až po využití ve zpracování lékařských snímků [31].

Jako názorná ukázka, kdy potřebujeme zpracovat velké množství obrazových dat, může sloužit tato práce. Je zaměřená na segmentaci neboli rozdělení obrazu do jednotlivých spojitých částí, které se dají lépe využít při následném zpracování. Tyto zpracovaná data z kamery využívá firma Artin pro svůj projekt Roboauta (autonomní vozidlo, které se na základě dat z kamer a senzoru dokáže samostatně pohybovat). Jde o další projekt z dnes tak moderní oblasti samořízených aut, na jejímž vývoji pracují ve velkých firmách jako je např. Google nebo Tesla.

Druhá kapitola práce je zaměřená na počítačové vidění. V kapitole jsou podrobně rozebrány varianty segmentace a popis dalších metod. V části o klasifikaci (kapitola 3) je popsáno, jak funguje neuronová síť (Neural network – NS). Její základní principy, architektura sítě, matematický popis neuronu a metody učení.

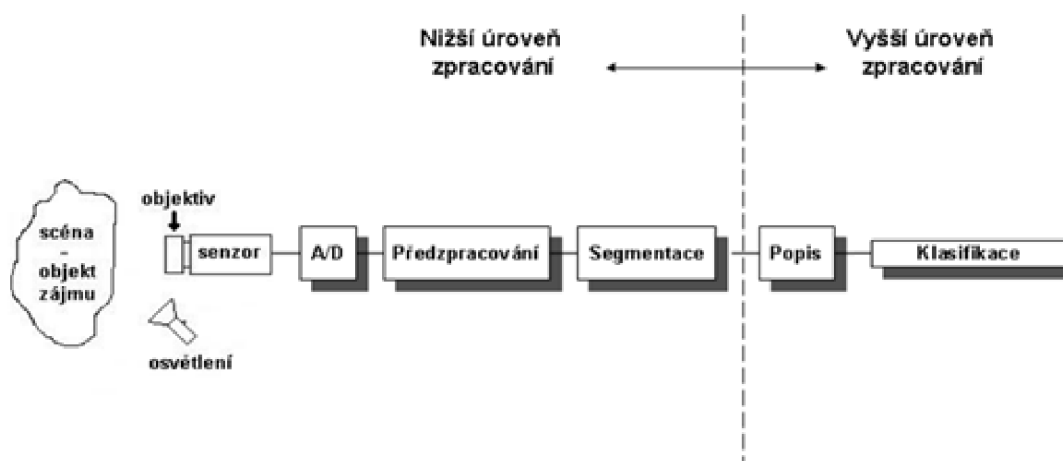
V kapitole čtyři je popis architektury konvoluční neuronové sítě (convolutional neural network – CNN) a vysvětlen princip konvoluční vrstvy, přenosová funkce a sdružování vrstvy. Pátá kapitola obsahuje popis architektury CNN pro sémantickou segmentaci.

V šesté kapitole je popis vlastností databázi obrazů CamVid, která je použita pro doučení sítě a následné testování. Speciálně pro ověření funkčnosti slouží databáze vlastních snímků MTech.

V kapitole sedm je popis před-učené sítě (Pretrained – PT), která je použita v praktické aplikaci, kdy slouží jako výchozí bod ze kterého začíná proces učení. Učit však celou CNN od začátku je velmi časově náročné, proto se využívají již PT CNN, které se upraví (doučí) na datech pro konkrétní aplikaci.

2 POČITAČOVÉ VIDĚNÍ

Počítačové vidění (computer vision) je odvětví výpočetní techniky zabývající se získáváním informací z obrazu. Obsahuje celý proces od snímání obrazu přes zpracování, segmentaci a popis až po vyhodnocení (obr. 1). Výběr vhodných operací vede k dobrému výsledku. Pokud již od začátku víme, k čemu bude obraz použit je vhodné k tomu směřovat jednotlivé kroky. V následujících kapitolách budou popsány jednotlivé části tohoto řetězce [16].



Obr. 1 Řetězec zpracování obrazu [21]

2.1 Snímání obrazu

Snímáním rozumíme převedení některé optické veličiny na elektrickou. Snímacímu zařízení je potřeba věnovat mnoho pozornosti. Snímací zařízení, vhodné osvětlení atd. mají na další zpracování rozhodující vliv [31].

Pro osvětlení snímané scény je mnoho způsobů, jaký typ osvětlení použít. Musíme si však hlídat všechny vlastnosti jednotlivých svítidel. Důležitým parametrem je blikání, které pouhým okem sice vidět nemůžeme ale ve výsledném obraze je to poznat. Příkladem svítidel, které blikají je zářivka nebo žárovka, naopak jako neblinkající osvětlení jsou vhodné LED žárovky. Někdy se používá pouze sluneční záření, avšak je zde riziko širokého spektra intenzity osvětlení.

Objektivy dělíme podle ohniskové vzdálenosti, změny ohniska, clony a speciálních vlastností (př. mikroskopické).

Hlavní funkcí senzoru je převedení světelného signálu na elektrický, pro tuto funkci se využívá fotoelektrický jev v polovodičích.

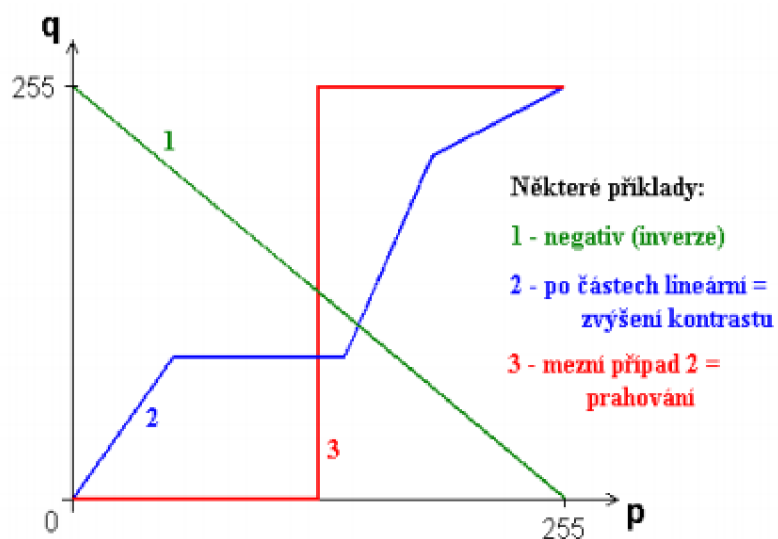
Převodník z analogových hodnot na digitální, v sobě zahrnuje proces vzorkování a kvantování [21].

2.2 Předzpracování obrazu

Na vstupu této části jsou digitální data z kamery, které je potřeba poupravit pro další zpracování, ovšem stále je na výstupu obraz. Hlavním cílem segmentace je potlačit šum, odstranit zkreslení a potlačit či zvýraznit rysy obrazu. Popíšeme čtyři základní metody pro předzpracování obrazu. [21].

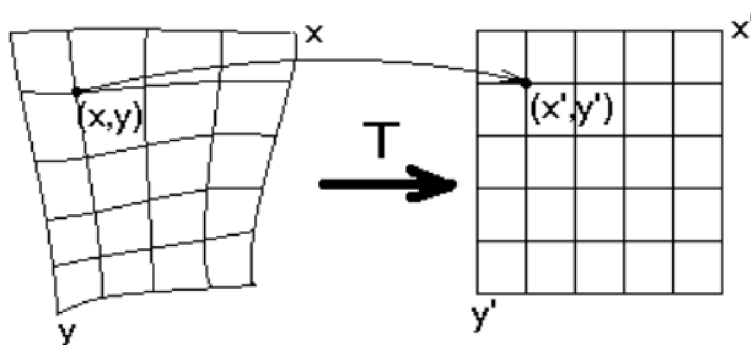
2.2.1 Bodové jasové transformace

Při jasové korekci měníme pouze jednotlivé body v obraze, který mohly být pozměněné odlišným hardwarem. Určitá jasová hodnota v obraze je transformována na jinou bez ohledu na její pozici [21]. Příkladem operací (viz obr. 2), které se zde provádějí je: inverze, ekvalizace histogramu, kontrast, prahování.



Obr. 2 Transformace jasové stupnice [21]

2.2.2 Geometrická transformace



Obr. 3 Geometrická transformace [21]

Tato transformace se používá pro odstranění geometrického zkreslení, které vzniklo při špatném úhlu objektu vůči snímané ploše nebo širokouhlým snímačem. Příklad takového zkresleného obrazu lze vidět na obr. 3. Další možností použití je změna rozlišení obrazu, posunutí nebo otočení. Při těchto operacích se setkáváme s častými problémy. V transponovaném obraze mohou vznikat díry nebo naopak se několik pixelů mapuje na totéž místo, část původního obrazu může ležet mimo obraz. Je potřeba najít vždy správnou transformační funkci. Možnými řešeními je aritmetický průměr čtyř nejbližších sousedů nebo lineární, kubická interpolace. Využívá se nadbytečnosti údajů v obraze (sousední obrazové body mají většinou téměř stejnou hodnotu) [21].

2.3 Lokální předzpracování

Lokální předzpracování využívá pro výpočet výstupní hodnoty obrazového bodu pouze lokální okolí ve vstupním obraze. Tuto metodu můžeme rozdělit do dvou kategorií: vyhlazování obrazu (filtrace) a detekce hran [31]. Celý tento mechanismus pracuje na principu diskrétní konvoluce, jenž bude popsána v kapitole 4.2.1.

2.3.1 Filtrace obrazu

Častým cílem filtrace je vyhlazení šumu v obraze. Konvoluční vrstva prochází celý obraz filtrem, který může mít více možností rozložení hodnot. Základní metodou je průměrování. Tento filtr přiřadí novou jasovou hodnotu bodu jako aritmetický průměr jasů jeho okolí. Lokální aritmetický průměr můžeme i upravit zvýšením váhy ve středu filtru, viz obr. 4.

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{zvýšení váhy středu: } h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{nebo 4-sousedů: } h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Obr. 4 Příklad filtru pro průměrování [21]

Další možností, jak zvolit typ filtru je Gaussovo rozložení, která dává důraz na střed masky. Tyto metody průměrování mají jako velkou nevýhodu rozmazání hran v obraze. Tento problém lze vyřešit rozložení hodnot v masce použitím operace mediánu. Tento typ má jako výhodu redukci rozmazávání hran, ale jeho velkou nevýhodou je, že poškozují tenké čáry a ořezává ostré rohy [31].

Filtrace metodou rotující masky se snaží podle homogenity okolí najít k filtrovanému bodu část jeho okolí, ke kterému pravděpodobně patří.

2.3.2 Detektory hran

Hrana v obraze je vlastnost obrazového elementu a jeho okolí. Při detekci hran se vychází z toho, že hrana je místo, kde se skokově mění hodnota jasu. Hrana je určena velikostí gradientu a směrem.

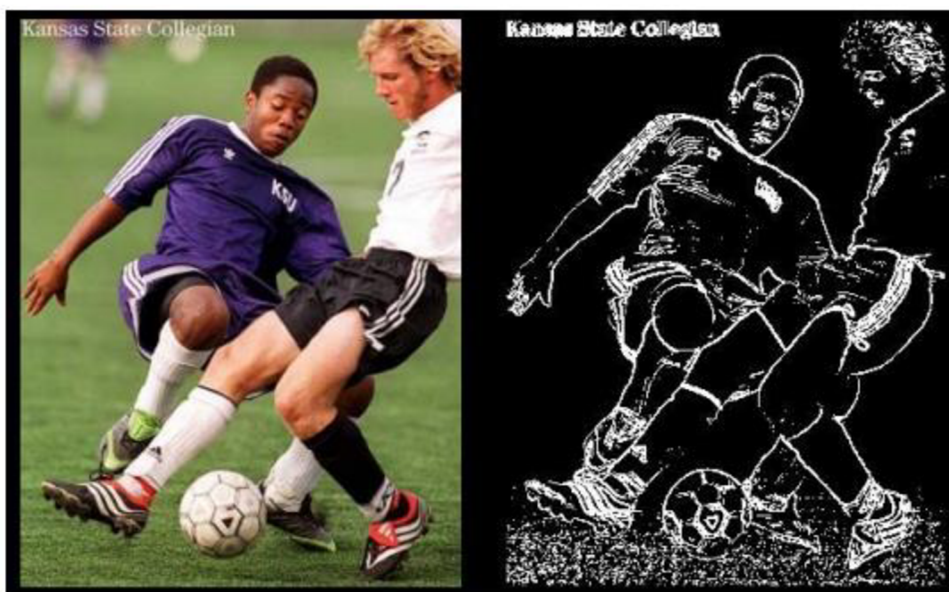
Hranice oblastí obrazu jsou tvořeny hranami, které můžeme najít použitím některého hranového operátoru:

- Robertsův operátor
 - využívá okolí 2×2
 - nevýhodu velká citlivost na šum, z důvodu použití malého okolí
- operátor Prewittové
 - hodnota gradientu odhadována z okolí 3×3
 - pokračuje se ve směru, který má největší modul gradientu
- Sobelův operátor
 - pracuje podobně jako Prewittové, avšak větší důraz na centrální buňku
 - nejvíce se využívá pro detekci vodorovných a svislých hran
- Cannyho hranový detektor
 - využívá konvoluce Gausiánu se obrazem a následně derivace ve směru gradientu
 - pro určení důležitých hran se výstup detektoru ještě prahuje
 - používá se prahování s hysterezi, vytváří souvislé hrany

Výstupem tohoto operátoru je obraz, kde jsou hrany zdůrazněny ale může obsahovat další artefakty odpovídající nehomogenitám v obraze. Proto je nutno obraz ještě upravit, obvykle postačuje prahování. Na obr. 5 je ukázka detekce hran z barevného obrazu.

Metody tedy pracují na principu nalezení lokálních maxim v první derivaci signálu nebo v nalezení průchodu nulou po druhé derivaci. Nejčastěji je využívána konvoluce obrazu s maskou, která aproximuje první derivaci.

Dalším využitím hranového detektoru je ostření obrazu. Využívá se často v tiskařském průmyslu pro zvýšení kontrastu obrazů [31].



Obr. 5 Detekce hran [25]

2.4 Segmentace obrazu

Hledání objektů patří mezi analýzu obrazu. K těmto úkonům se používá právě metoda segmentace obrazu, jejímž hlavním úkolem je rozdělit obraz do takových částí, které mají úzkou souvislost s oblastmi reálného světa zachyceného na obraze nebo mají společné vlastnosti. Tyto vlastnosti (charakteristiky) závisí na typu objektu, který se snažíme rozpoznat. Pokud rozpoznáváme banány od jablek bude jako charakteristiku objektu volit tvar (kulatost) případně barvu [16]

Základní charakteristiky:

- barva – můžeme použít průměrnou nebo jen jednu námi zvolenou,
- konvexnost, konkávnost,
- tvar objektu – např. kulatý, excentrický,
- plnost objektu – poměr obsahu pozadí ku obsahu objektu.

Výsledkem segmentace je soubor vzájemně se nepřekrývajících oblastí. Podle výsledku dělíme segmentaci na kompletní (oblasti odpovídají objektům) a neúplnou (oblasti odpovídají jen určitým vlastnostem). Velkou výhodou tohoto procesu je snížení objemu dat. Problémem však může být nejednoznačnost výsledků nebo šum.

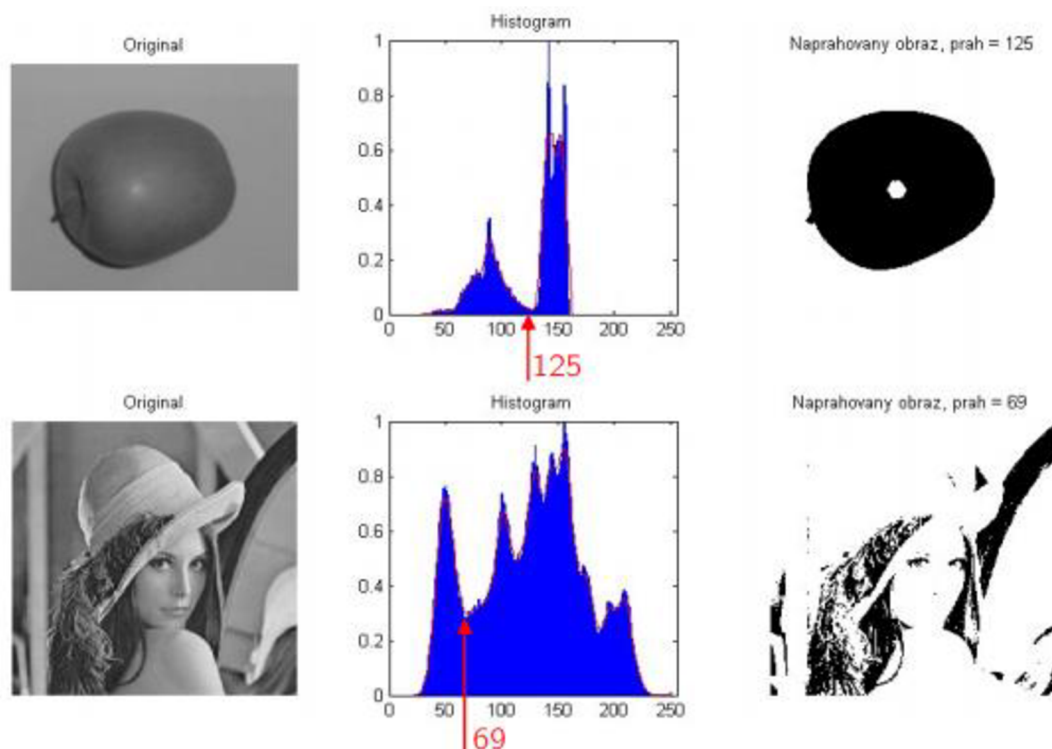
2.4.1 Segmentace prahováním

Mezi nejstarší a zároveň výpočetně nejjednodušší metody segmentace se řadí prahování (thresholding). Využívá se však většinou pouze u jednoduchých případů (předměty na bílém pozadí). Objekty jsou charakterizovány konstantní odrazivostí či pohltivostí svého povrchu. Využijeme hodnoty jasu k oddělení objektů od pozadí. Volba správné velikosti prahu je zásadní pro úspěšný výsledek. V ideálním případě je výstupem prahování kompletní segmentace celého obrazu. Avšak u nehomogenního obrazu (obsahuje různě jasné úseky) nelze určit jednotný práh. Existuje ale základní metod, jak určit hodnotu prahu, např. adaptivní prahování, s více prahy, prahování s hysterezí. Důležitou roli při volbě, jakou metodu prahování použijeme, hrají histogramy jasových hodnot. Podle histogramu se určuje i hodnota prahu v některých případech, názorná ukázka je na obr. 6. Pokud máme obraz černobílý vznikne histogram intenzity jasu, kde na ose x máme stupně (0 = černá až 255 = bílá) a na ose y počet odpovídajících obrazových bodu. Tuto operaci můžeme popsat vztahem,

$$g(i, j) = \begin{cases} 1, & f(i, j) < T \\ 0, & f(i, j) \geq T \end{cases} \quad (2.1)$$

kde T představuje hodnotu prahu.

Všechny body, které mají po segmentaci hodnotu 1 jsou objekty. Body s hodnotou 0 jsou pixely pozadí. Toto však je jen ukázka při využití jednoho prahu, se kterou se v praxi nedá moc pracovat [1].



Obr. 6 Histogram a segmentace pomocí jednoho prahu [25]

2.4.2 Segmentace narůstáním oblastí

Tyto metody pomáhají najít body, které mají nějakou společnou vlastnost. Spojování segmentu jde postupně od jednoho bodu až po celou oblast. Iniciální pixely jsou rozmístěny rovnoměrně nebo náhodně, segment pak vzniká iterativním rozrůstáním. Základní myšlenkou je rozčlenit obraz maxima souvislých oblastí, aby z hlediska vlastností byly homogenní. Výhodou těchto metod je schopnost segmentovat i takový obraz, který obsahuje značné množství šumu [31].

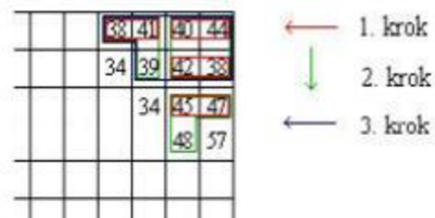
Mezi regionální metody tedy patří:

- spojování oblastí
- štěpení oblastí
- štěpení a spojování
- watershed

Spojování homogenních oblastí - před započítím operace je potřeba obraz rozdělit na mnoho malých oblastí, následně definujeme kritérium pro spojování obrazových bodů. Operace je dokončena, pokud již nelze spojit žádnou oblast, aniž bychom porušili vstupní kritérium [25]. Celý proces je názorně zobrazen na obr. 7.

Příklad:

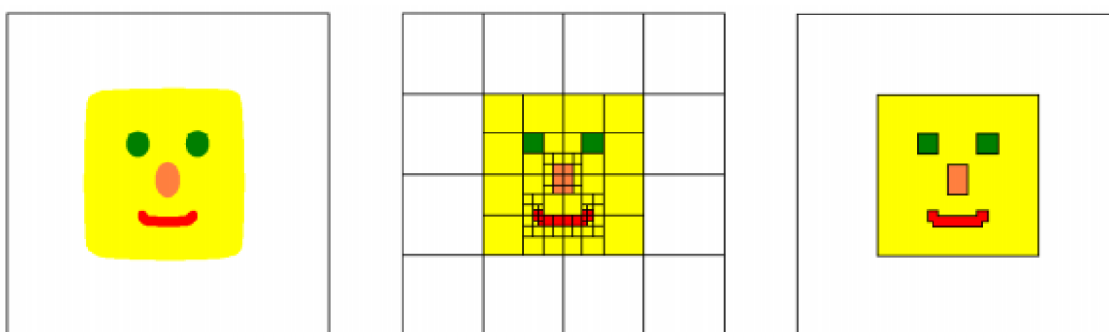
Pokud rozdíl jasů dvou sousedních oblastí je menší než 5, pak tyto oblasti spoj.



Obr. 7 Metoda spojování oblastí [25]

Štěpení oblastí – je principiálně opačný postup než spojování. Celý obraz dělíme tak dlouho, dokud jednotlivé oblasti neodpovídají zvolenému kritériu. Můžeme se však dostat pokaždé k jinému výsledku v závislosti na volbě výchozího bodu.

Štěpení a spojování – využití pyramidální reprezentace v rámci čtvercových oblastí v obraze. Pokud je daná oblast v úrovni pyramidy nehomogenní je rozdělena na 4 další podoblasti. Pokud jsou následně všechny 4 oblasti homogenní, lze je zpátky spojit [26]. Ukázka celého postupu je na obr. 8.



Obr. 8 Spojování a štěpení [25]

Watershed (vodní předěl) – obraz je chápán jako topografický reliéf, kde jas vstupního obrazu určuje výšku terénu (černá nejnižší, bílá je nejvyšší). Postupně je terén zaplavován, vznikají tedy oblasti se stejným indexem.

2.4.3 Sémantické metody segmentace

Segmentace je rozdělení obrazu do několika částí se stejnými vlastnostmi, avšak bez jakéhokoli pokusu o pochopení toho, co tyto části představují. Kdežto sémantická segmentace (semantic segmentation – SS) se pokouší rozdělit obraz do sémanticky významných částí a klasifikovat každou část do jedné z předem určených kategorií (silnice, člověk, auto atd.) [18]. Metody, jak toho lze dosáhnout a konkrétní architektury jsou v kapitole 5.

2.5 Popis

Cílem popisu je určit buď číselný vektor příznaků nebo nečíselný syntaktický popis charakterizující tvarové vlastnosti oblasti. Popis objektů je proces, kdy získáváme příznaky ze segmentovaných dat. Příznaky neboli charakteristické rysy slouží pro následnou klasifikaci objektů. Na příznaky jsou dány nároky [31]:

- invariantnost – příznaky jsou nezávislé na změně jasu, kontrastu, rotaci apod.,
- spolehlivost – objekty v jedné třídě mají podobnou hodnotu příznaků,
- diskriminabilita – objekty z jiných tříd musí mít jiné hodnoty příznaků,
- efektivita výpočtu – příznaky lze jednoduše detekovat,
- časová invariance – příznak je stejný i při dynamickém zpracování obrazu.

Vstupními daty pro popis je v ideálním případě poloha a tvar objektu v obraze. Cílem popisu je vlastně matematická reprezentace objektu vektorem \mathbf{X} (příznakový vektor). Tento vektor je však příliš velký pro další zpracování, proto je nutné jej redukovat pomocí metod selekce nebo extrakce [15].

2.6 Klasifikace

Klasifikace je posledním krokem při zpracování obrazu a často se vyskytuje jako nepostradatelný prvek jiných řídicích strategií. Téměř vždy jsou k dispozici informace o možných skupinách třízení. Na základě dostupného popisu jsou objekty či oblasti klasifikovány a s oblastmi se dále pracuje jako se samostatnými jednotkami charakterizovanými informacemi o třídě, ke které náleží [31]. Správná volba klasifikačního algoritmu je pro úspěch zásadní. Popíši Vám nyní jeden z základních klasifikačních algoritmu, a to umělou neuronovou síť.

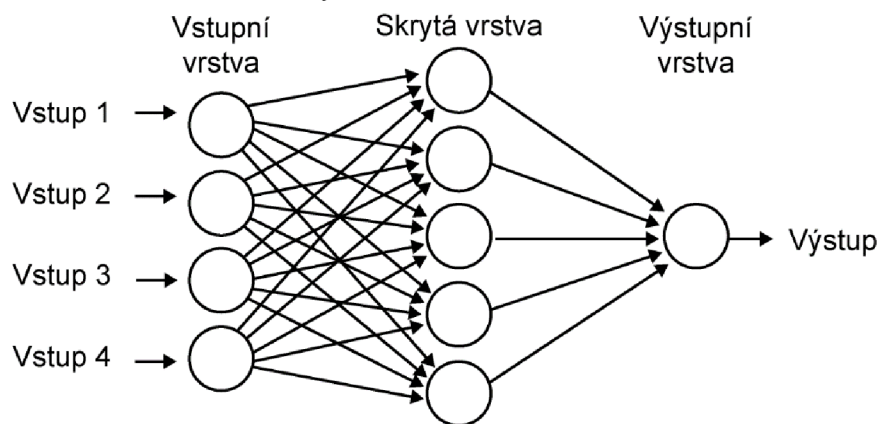
3 UMĚLÁ NEURONOVÁ SÍŤ

Umělá neuronová síť (UNS) je paralelní distribuovaný systém výkonných prvků modelujících biologické neurony, které jsou účelně uspořádány tak, aby systém byl schopen požadovaného zpracování informací [20]. Tento algoritmus byl vytvořen na základě fungování lidského mozku. Bylo zjištěno, že mozek pracuje na principu posílání impulsů mezi mnoha spojenými buňkami nazývanými neurony. Tedy již od začátků prvních počítačů, bylo za cíl vytvořit algoritmus, který by napodoboval neurony a jejich činnost v lidském mozku [34].

3.1 Princip sítě

Neuronová síť je tvořena matematickými modely neuronu, kde každý zpracovává váhové uskupení a generuje výstup. Neurony jsou topologicky uspořádány do struktury komunikující přes orientované ohodnocené spoje. Každá síť může být tedy jiná. Záleží na typu neuronu (resp. typ přenosové funkce), topologickém uspořádání a strategii učení. Z obrázku č. 9 vidíme, že jsou neurony uspořádány do jednotlivých vrstev. Neurony v jedné vrstvě nejsou propojené, kdežto mezi vrstvami je plné propojení. Každý spoj je ohodnocen váhou, která představuje intenzitu procházejícího signálu. V tomto uspořádání je vždy jedna vstupní vrstva a jedna výstupní, ale skrytých vrstev může být více.

Dopředná neuronová síť představuje dynamický systém, kde každý neuron pracuje samostatně. Dopřednou je nazývána, protože se signál šíří pouze jedním směrem, a to od vstupu k výstupu. Taková síť se vyznačuje robustností a odolností vůči poškození. Dokáže tedy poskytnout relativní výstupy i při poškození některých obrazových elementů. Tato topologie je však jen jednou z mnoha možností uspořádání neuronů. Existuje celá řada dalších konceptů, ať už zavedení zpětné vazby nebo propojení každého neuronu s každým.



Obr. 9 Uspořádání neuronů do vrstev [22]

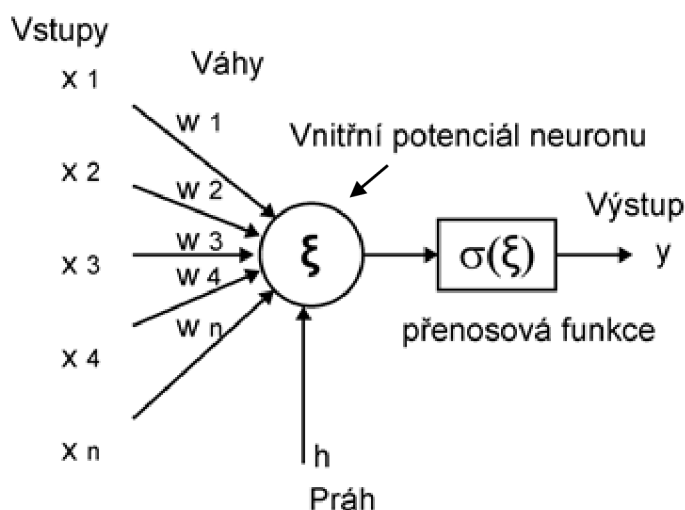
Každá NS má dvě fáze. První fází je vždy učení, kdy síť na základě tréninkové množiny učí své parametry. Následně přechází do fáze vybavování, kdy na základě předložených vstupů, produkuje výstupy a parametry sítě se již nemění [22].

3.2 Matematický model neuronu

Vstupem do neuronu je vektor o n prvcích. Každý vstupní prvek má svou váhu, která udává jeho propustnost (důležitost). Vnitřní potenciál neuronu je hodnota, která vzniká porovnáním všech vstupních váhových prvků sečtených s prahem, viz obr. 10.

Každý vstupní prvek je tedy násoben svou váhou, kde:

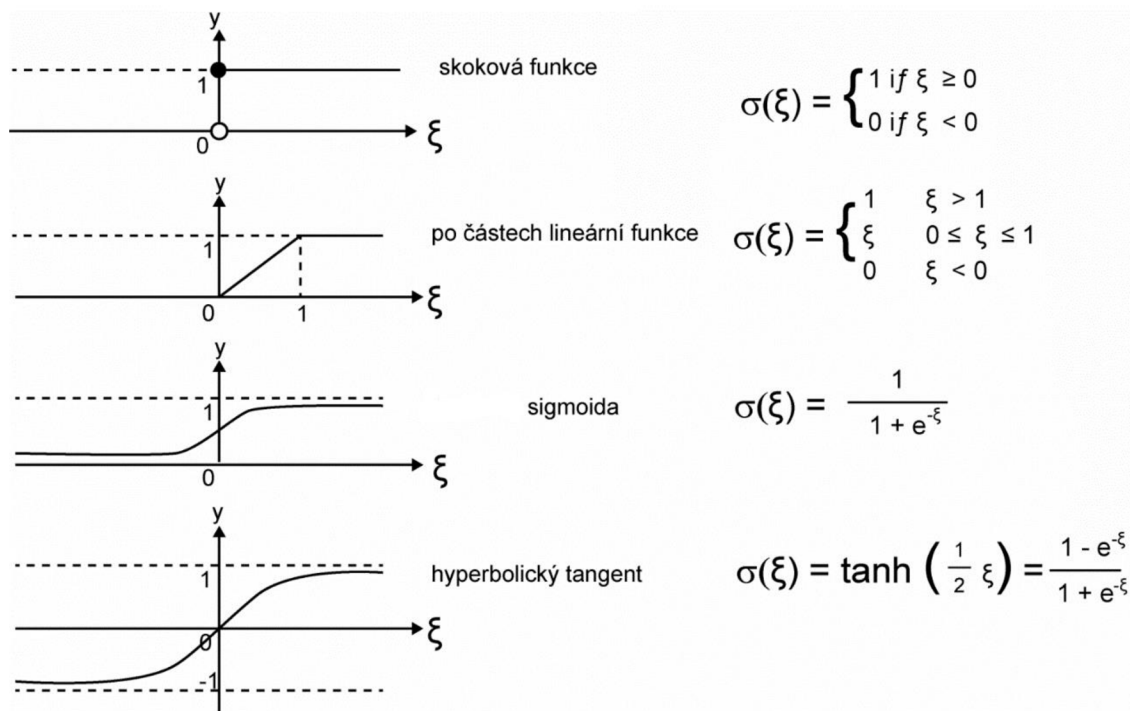
$$\xi = \sum_{i=1}^n w_i x_i - h \tag{3.1}$$



Obr. 10 Matematický model neuronu [25]

Přenosová funkce pak provádí obecnou nelineární transformaci. Základní přenosové funkce jsou na obr. 11. Obvyklou podmínkou pro výběr funkce je její diferencovatelnost, což nám dává výhodu při využití učících algoritmu na základě gradientní metody.

Existují i další koncepce neuronů, kde je výstup vyčíslen vzdáleností vektoru x od vektoru vah w . Další možností jsou vlnkové sítě, vyčísľují rozdíl mezi vstupním signálem a posunutou vlnovou funkcí každého neuronu. Vlnkové sítě se hojně využívají při analýze signálů a obrazů [24].



Obr. 11 Přenosové funkce. [24]

3.3 Principy učení

Vstupní váhy w představují paměť neuronu. Učení obstarává adaptační algoritmus, který nastavuje váhy. Tento proces probíhá iterativně, kdy algoritmus má k dispozici množinu vstupních dat a k tomu odpovídající výstupní (správná řešení). Při řešení postupuje na základě analogie, s již známými vzory. Nastavení vah tedy odpovídá nalezení nejvýstižnější transformace vstupních vektorů na výstup.

Velkou výhodou NS je schopnost nalezení správné transformace i v těch případech, kdy to analyticky je řešitelné jen velmi stěží nebo vůbec. Postačuje k tomu velké množství tréninkových dat, na kterých se může daná síť správně naučit. Musíme si však také uvědomit, že ačkoliv jsou NS výborným nástrojem pro řešení obtížných úloh, není zde zaručeno, že vždy dojdeme k požadovanému výsledku. Učení může skončit i neúspěchem [2].

3.3.1 Učení bez učitele

Učení bez učitele je metoda strojového učení, která hledá skrytou strukturu v datech. Pracuje na principu *shlukové analýzy*, tudíž hledá ve vstupních datech podobné elementy, které jsou tříděny do skupin (počet skupin může být znám předem), ve kterých mají tyto objekty podobné vlastnosti. Tedy není zde v průběhu učení žádná kontrola, zda učení postupuje správným směrem. Do učení nijak nezasahujeme, tedy celé učení vychází pouze z informací získaných ze vstupní množiny dat. Učení je tedy trvalého charakteru, jelikož nevíme, kdy je již naučen správně. Jde o tzv. Kohonenovu síť neboli se jí také říká samo organizující mapa [33].

3.3.2 Učení s učitelem

V tomto případě má adaptační algoritmus dostatečnou množinu dvojic vstup ku odpovídajícímu výstupu. Má tedy konkrétní příklady správné transformace, které využívá k adaptaci vah, naučení sítě do požadovaného tvaru. Množina dat je rozdělena na dvě části tréninková a testovací. Poměr není přesně dán, je třeba jej volit vzhledem k povaze úlohy. Obvykle tréninková množina obsahuje zhruba 60 %-80 % dat z celkové množiny dat.

Algoritmus předkládá postupně jednotlivé prvky a zjišťuje odchylku od očekávaného výstupu a následně provádí korekci vah. Tento postup, než se projde celá testovací množina se nazývá epocha. K naučení sítě je obvykle potřeba stovky až tisíce epoch. Okamžik zastavení učení je nejčastěji volen dosažením určité hodnoty celkové chyby, kdy prohlásíme síť za naučenou. K zastavení učení také může dojít, pokud se chyba ustálí na nějaké hodnotě a již dále neklesá.

Výkonnost naučené sítě pak ověříme pomocí testovací množiny. Kritérii správného naučení je více, nejčastěji je to však střední kvadratická odchylka mezi nalezenými výstupy a očekávanými. Pokud je výkonnost nad testovací množinou dobrá, předpokládáme, že bude přibližně stejně dobrá i u vstupů mimo naši množinu [34].

3.4 Zpracování obrazu

Neuronové sítě (např. s učením Backpropagation) založené pouze na vrstvách neuronů, které bychom chtěli využít pro rozpoznávání a klasifikaci obrazů, nemusí být tím nejlepším řešením. Aby totiž bylo možné efektivně pracovat s invariantními vstupními obrazy (rotace, změna jasu atd.), bylo by nutné síť udělat velmi hlubokou (mnoho vrstev, mnohdy dosahující 20 až 50 vrstev) a i počet neuronů by musel řádově odpovídat počtu pixelů. To je sice skutečně možné zařídit (ostatně zaplatíme za to „jen“ strojovým časem), ovšem stále zde narážíme na principiální omezení klasických vrstvených neuronových sítí – jednotlivé neurony se učí izolovaně od ostatních neuronů, zatímco na vstupu máme „plovoucí“ obrázek. Bylo by tedy výhodnější se zaměřit na vylepšení samotné architektury neuronové sítě specializované právě na to, že na vstupu bude mít bitmapu, a tudíž by sousední neurony měly nějakým způsobem sdílet své váhy na vstupech. Moderní trendy jsou v použití konvoluční neuronové sítě, které se používají primárně pro práci s obrazovými daty.

4 KONVOLUČNÍ NEURONOVÉ SÍTĚ

Konvoluční neuronové sítě (CNN) jsou nástrojem pro práci s rozpoznáváním obrazu nebo zvuku. Konvoluční část se stará o předzpracování obrazu, které následně předá ve formě vektoru do NS na vstup. Jinak řečeno, konvoluční vrstva extrahuje užitečné informace z obrazu automaticky, aby s nimi následně klasifikátor mohl pracovat. Dokáže tedy lépe redukovat počet vstupních parametrů [11].

4.1 Struktura konvoluční neuronové sítě

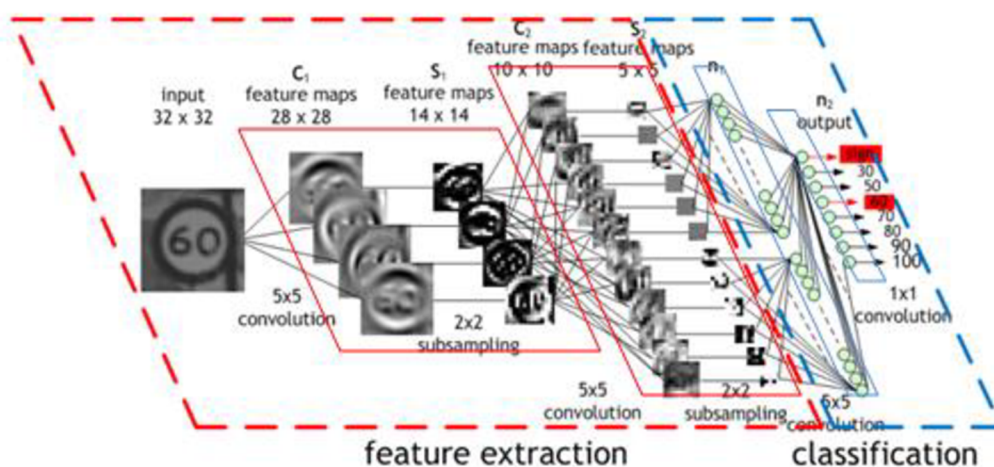
Základním rozdělením CNN je na extrakci příznaků, tedy konvoluce, a klasifikace pomocí NS. V kapitole číslo 3. byl popsán postup, kdy NS dostane na vstupu vektor hodnot, které projdou přes sérii skrytých vrstev, které byly vždy plně propojeny s předcházející vrstvou. Poslední vrstva se nazývá výstupní a reprezentuje již rozdělení do kategorií.

Jako vstupní data se používají obrázky s malým rozlišením. A to například jen $32 \times 32 \times 3$ (32 výška, 32 šířka, 3 vrstvy – barevný obraz RGB). Tedy každý neuron při plném propojení by musel mít $32 \times 32 \times 3 = 3072$ vah. Což ještě není až tak velké číslo, avšak pokud bychom měli větší obraz, již se budeme dostávat na desítky tisíc vah. Z tohoto důvodu již není možné mít plně propojeny všechny neurony [4].

4.2 Architektura sítě

Z obrazu č. 12 vidíme, že konvoluční síť se skládá z několika vrstev, kdy každá má jinou funkci. Používáme tři hlavní typy funkcí:

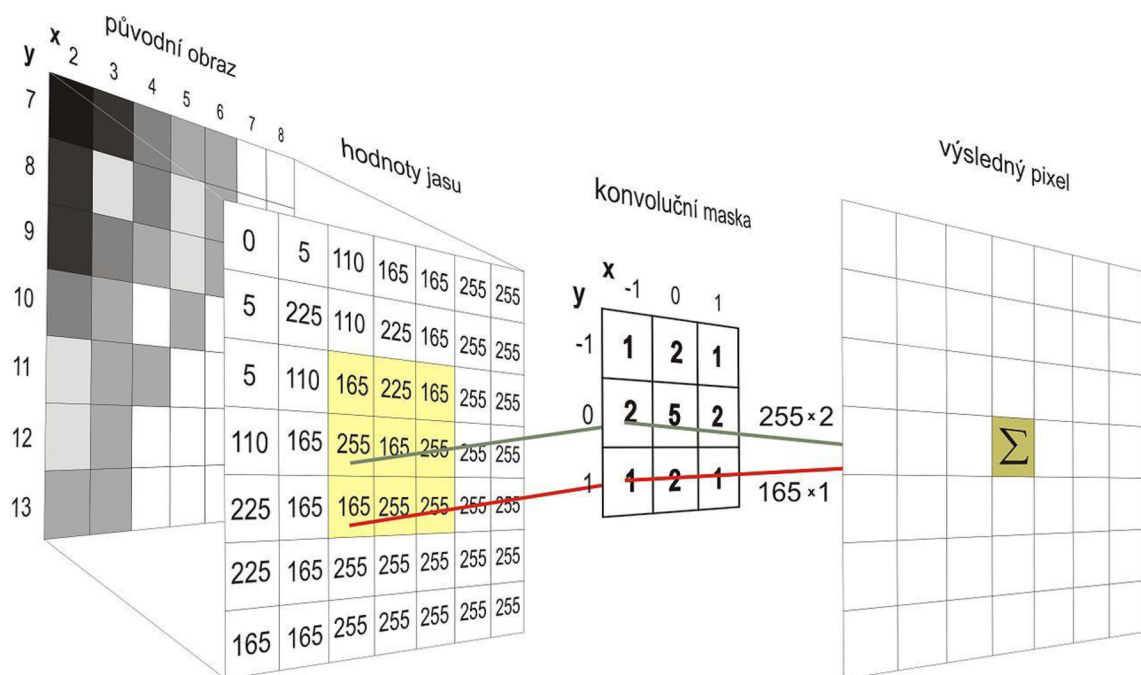
- konvoluční vrstvu,
- aktivační funkci,
- sdružení / podvzorkování.



Obr. 12 Schéma architektury CNN [7]

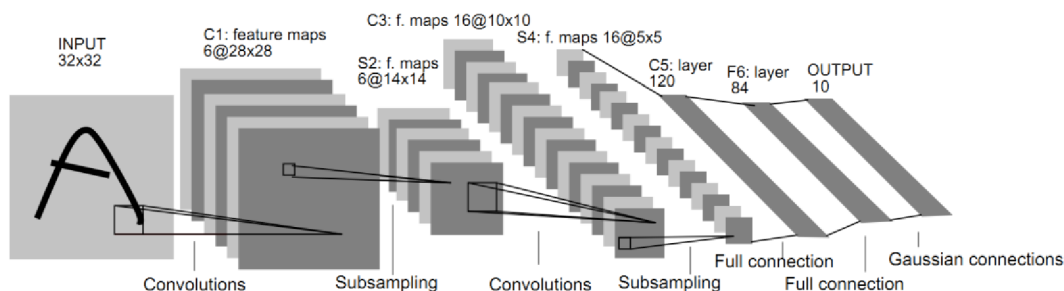
4.2.1 Konvoluční vrstva

Konvoluční vrstva (CONV) aplikuje filtr/masku na vstupní obraz. Velikost masky si můžeme zvolit, např. $5 \times 5 \times 3$. Opět zde máme výšku, šířku masky a taky její hloubku (tři barevné vrstvy RGB). Každý pixel má svou barvu, která je popsána číslem. Maskou se postupně prochází celý obraz a násobíme jednotlivé pixely s hodnotami masky. Tyto výsledky sečteme a výsledkem bude jedno číslo. Celý tento postup je názorně vykreslen na obr. 13. Masku se však nemusí posouvat pouze po jednom kroku, může se také posouvat o větší počet kroků. Tím, že používáme masku o určité velikosti, dojde ke zmenšení obrázku [7] [8].



Obr. 13 Konvoluce

Ve většině případu se však obrázek prochází více filtry najednou. Máme tedy například 6 filtrů a každý z nich vytvoří dvourozměrnou aktivační mapu. Na obr. 14 vidíme, jak nám roste počet aktivačních map v jednotlivých vrstvách, ale zároveň se mapy zmenšují. Jedním důvodem je velikost masky a druhým použití sružovací vrstvy [4][5].



Obr. 14 Princip konvoluční vrstvy [8]

Volba velikosti masky také souvisí s množstvím parametrů, které bude daná funkce muset mít. Jako příklad můžeme uvést filtr 5x5x3 (trojka vyjadřuje, že se jedná o 3 barevné kanály, tedy RGB obraz), který se obvykle používá na vstupu. Celkově to tedy bude 5x5x3, což se rovná 75 váhám, které budou muset být jako parametr. Vždy je nutno ještě připočítat práh, tudíž ještě plus jedna. Praktický příklad výpočtu: máme obraz o velikosti 32x32x3 a aplikujeme na něj 6x filtr velikosti 5x5x3. Na tento výpočet použijeme vzorec 3.1 [23].

$$(N - F)/(krok + 1) \tag{3.1}$$

N... velikost obrázku

F ... velikost masky

Po dosazení:
$$\frac{32-5}{1+1} = 28 \tag{3.2}$$

Výsledný obraz bude velikosti 28x28x6

V případech, kdy volíme větší krok než jedna a nevycházelo by nám tak obsáhnout celý obraz, aniž bychom nevynechali žádný obrazový bod, používá se funkce zero-padding. Tato funkce dokáže zvětšit obraz o jeden bod. Na hrany obrazu vloží nulové pixely, což se projeví jako nepatrný černý rámeček.

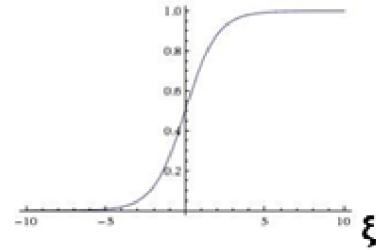
Pokud máme více konvolučních vrstev za sebou, což se v praxi používá, můžeme již hovořit jako o hlubokém učení (deep learning) [19].

4.2.2 Přenosová funkce

V kapitole 3.2 již bylo popsáno, jak funguje matematický model neuronu. Nyní se dostáváme k jeho přenosové funkci. Doporučuje se používat ReLU, a rozhodně při učení CNN nepoužívat sigmoidu [13]. Základní jsou tyto čtyři funkce

Sigmoida

- Rozsah hodnot [0,1)
- Historicky nejpoužívanější
- při saturaci již nemůžeme detekovat gradienty
- není souměrný kolem nuly
- exponenciální funkce je výpočetně náročná



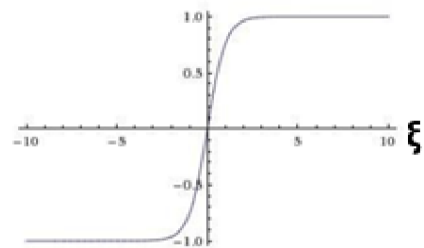
Sigmoid

$$\sigma(\xi) = 1/(1 + e^{-\xi})$$

Obr. 15 Sigmoida [13]

Tanh(ξ)

- hodnoty v rozsahu [-1,1]
- souměrný kolem nuly
- stále nevýhoda saturace

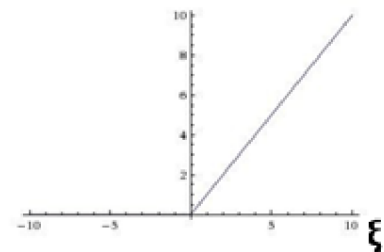


tanh(ξ)

Obr. 16 Tanh(ξ) [13]

ReLU

- není zde problém se saturací
- výpočetně efektivní
- konverguje až 6x rychleji než sigmoida nebo tanh
- výstup není centrován kolem nuly
- může však během tréninku „zemřít“

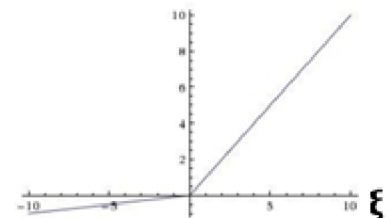


ReLU $f(\xi) = \max(0, \xi)$

Obr. 17 ReLU [13]

Leaky ReLU

- nedochází k saturaci
- efektivita výpočtu
- rychle konverguje
- nemůže „zemřít“



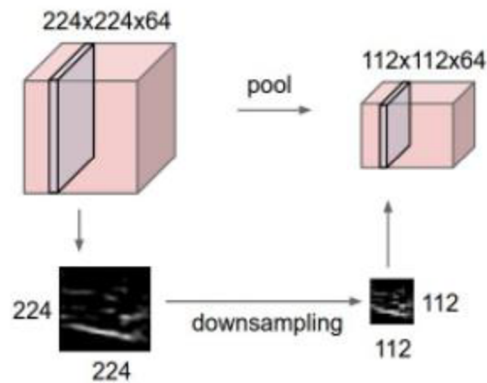
Leaky ReLU

$$f(\xi) = \max(0.01\xi, \xi)$$

Obr. 18 Leaky ReLU [13]

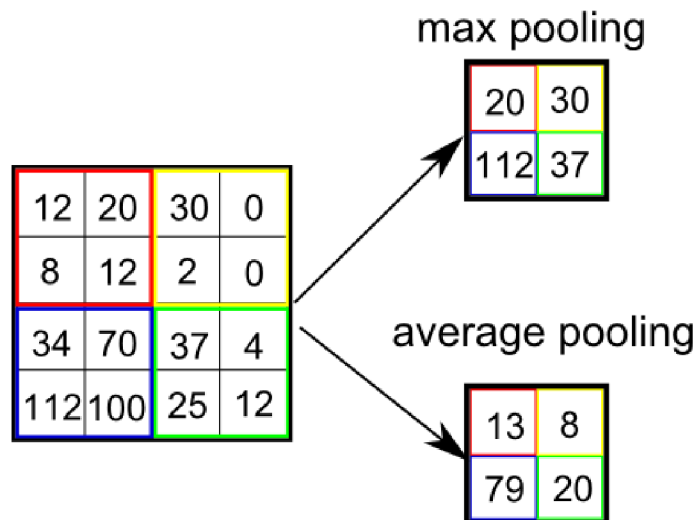
4.2.3 Sdružení

Sdružení (*pooling*) má za úlohou progresivně redukovat velikost reprezentovaných dat a tím redukovat množství parametrů předávaných do sítě. Tato vrstva pracuje nezávisle na hloubce vstupu. Nejpoužívanější je tato vrstva při použití masky 2x2 aplikované s krokem 2. Redukuje tak 75 % ze vstupních dat. Tedy na výstup se dostane jedna čtvrtina vstupních dat [3]. Ukázka zmenšení velikosti obrazu je na obr. 19.



Obr. 19 Zmenšování obrazu [4]

Máme dva typy této funkce, a to výběrem maxima z dané masky nebo průměru, viz obr. 20, při použití MAX, se vždy aplikuje maska, a z té se vybere nejvyšší hodnota. Pro příklad, kdy je použita funkce výběrem průměru se z daných hodnot pod maskou vypočítá průměr [19].



Obr. 20 Možnosti sdružování [13]

4.3 Učení CNN

Učení spočívá v nastavení hodnot filtrů, kterými se prochází obraz v konvoluční vrstvě. Před zahájením učení jsou hodnoty filtru náhodně rozděleny. Nejlepší možností na začátku učení, je použití Gaussova rozložení se směrodatnou odchylkou $\sqrt{2/n}$, kde n je počet výstupních neuronů. Špatnou volbou jsou příliš malá nebo naopak velká čísla. Síť je schopna filtry nastavit prostřednictvím procesu učení, ten probíhá tak, že CNN dostane velké množství obrazů, kde ke každému obrazu je přiřazen popis. Prochází se tedy postupně celá tréninková sada, kdy dáváme na vstup postupně obrazy a na výstupu hodnotíme, jestli byl přiřazen správný popis k danému obrazu [8]. Dalším krokem je nastavení vah na vstupech neuronů bylo již vysvětleno v kapitole 3.3.

4.4 Princip vybavování

Vybavování neboli aktivní režim sítě, se využíván k vlastnímu použití naučené sítě na daný výstup. Může se jednat o jednorázový proces nebo iterační [20].

4.5 Před-učené sítě

Naučit celou neuronovou síť je i pro výkonné počítače velmi časově náročné. Takové učení zabere řádově dny, někdy až týdnech při opravdu velkých sítích. V praxi se tedy kvůli snížení časové náročnosti používá, již částečně před-učená síť, kterou stačí následně jen doladit na naše požadovaná data. Sítě jsou pojmenované většinou po svých tvůrcích, ve kterých je implementována i jejich originální architektura. Z literární rešerše mne nejvíce zaujalo těchto šest sítí [4].

AlexNet – první síť, která v roce 2012 ve větší míře zpopularizovala CNN v oblasti počítačového vidění. Síť má podobnou architekturu jako LeNet, ale má hlubší a větší konvoluční část (po jedna CONV vrstvě ihned následuje POOL vrstva).

LeNet – první úspěšná aplikace CNN již v roce 1990. Byla používána pro čtení zazipovaných souborů nebo čtení číslic a další aplikace.

ZFNet – jedná se o vylepšenou síť AlexNet. Zejména rozšíření velikostí středních konvolučních vrstev, zmenšení kroku a velikosti filtru na vstupní vrstvě.

GoogLeNet – síť z roku 2014. Dramaticky snížili počet parametrů (na 4 miliony, v porovnání s 60 mil od AlexNet).

VGGNet – ukázali, že hloubka sítě má kritické parametry pro dobrý výkon. Používá stále stejnou masku konvoluce 3x3 a pro sdružování masku 2x2. Jejich model je k použití v knihovně Caffe. Existují dva typy Vgg16 a Vgg19, které se liší pouze počtem vrstev. Na začátku je vždy vstupní vrstva poté skryté vrstvy a jako poslední výstupní vrstva, která klasifikuje objekty do 1000 kategorií.

ResNet – obsahuje speciální přeskokovací spojení. Nemá plně propojenou výstupní vrstvu.

4.6 Praktické aplikace CNN pro segmentaci obrazu

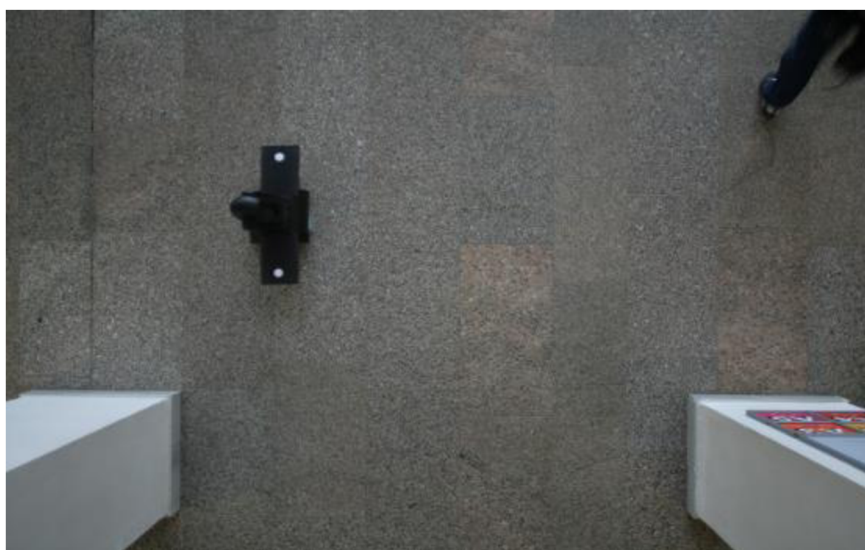
V praxi se CNN používají v mnoha různých odvětvích techniky, od systému strojového vidění přes vyhledávací, diagnostické a asistenční systémy až po autonomní vozidla. Uvedu čtyři příklady, kde je využití těchto sítí velmi přínosné a posunulo vývoj v některých oblastech kupředu.

4.6.1 Automatické rozpoznání stavu elektroměru

Fotografie je snímána pomocí mobilního telefonu, kde byla následně implementována aplikace pro Android na rozpoznání číslic. Na obrázku je detekována plocha číselníku a následně jednotlivá čísla, která jsou rozpoznávána za pomoci neuronové sítě. Je zde využito segmentace obrazu, pro následnou klasifikaci výstupu bylo použito vektorové pole (SVN). Byla použita knihovna OpenCV pro problematiku segmentace [12].

4.6.2 Detekce mobilního robota zpracováním obrazu

Pomocí metody segmentace prahováním byl hledán mobilní robot, který se pohyboval po místnosti. Obrázky byly foceny kamerou umístěnou na stropě kolmo k zemi (obr. 21) Aby bylo možno použít tuto metodu segmentace musel být mezi robotem a pozadím dostatečný kontrast. Byl zde ovšem problém odlišit pozadí od snímaného objektu, na robota tedy byly umístěny kontrastní značky, které pomáhají najít objekt v obraze. Konkrétně se jedná o bílé body na černém robotovi [27].



Obr. 21 Fotografie mobilního robota, pohled od stropu [27]

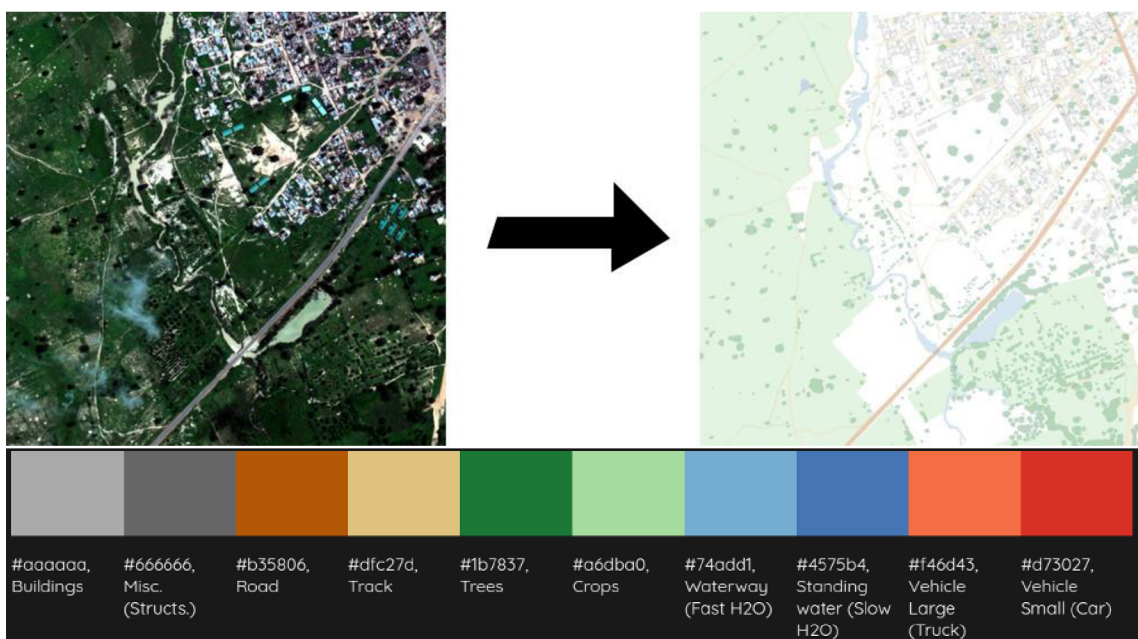
4.6.3 Analýza zdravotnických obrazů

Konvoluční neuronové sítě v posledních několika letech přetvořili lékařskou analýzu obrazu. Architektura U-Net je jednou z nejznámějších architektur CNN pro sémantickou segmentaci a dosáhla pozoruhodných úspěchů v mnoha aplikacích segmentace medicínských obrazů.

V této aplikaci však byla architektura U-Net porovnávaná s architekturou DriNet, která se skládá ze tří hlavních bloků. Jmenovitý konvoluční blok s hustými spoji, dekonvoluční blok se zbytkovými počátečními moduly a sdružovací blok. Navržená architektura DriNet překonala U-Net ve třech náročných úlohách, a to třídění cerebrospinálních tekutin v CT mozku, multi-orgánová segmentace CT obrazů břicha a segmentace mozkových nádorů na MR obraze [16].

4.6.4 Segmentace satelitních snímků

Segmentace obrazu je téma strojového učení, kde je potřeba nejen kategorizovat to, co je vidět v obraze, ale také pracovat až na úrovni pixelů. Vycházíme ze satelitního snímku a cílem je segmentovat na reprezentaci mapy, kde se věci v obrázku automaticky kategorizují (obr. 22). Nejprve jsou satelitní snímky vybaveny 20 použitelnými pásmy. Pokud červená, zelená a modrá představují 3 pásma (RGB), 20 pásem obsahuje mnohem více informací pro neuronovou síť, a tím ulehčí učení a kvalitu předpovědí. Pro naučení bylo k dispozici pouze 25 satelitních snímků, avšak jednotlivé části byly navíc trénovány pomocí databáze obrazů CIFAR-10 a CIFAR-100. Byla použita architektura U-Net a to hlavně kvůli schopnosti naučit se i z malého množství obrazů [17].



Obr. 22 Segmentace satelitního snímku [18]

5 SÉMANTICKÁ SEGMENTACE

Sémantická segmentace obrazu je porozumění obrazu na úrovni pixelů, tj. chceme každému pixelu přiřadit třídu objektů. Zatímco SS / analýza scény byla od roku 2007 součástí komunity počítačového vidění, hlavní průlom přichází až v roce 2014, kdy byly poprvé použity plně konvoluční neuronové sítě. Plně konvoluční síť (FCN) se používají pro sémantickou segmentaci přirozených obrazů, pro multimodální lékařskou analýzu obrazu a segmentaci multispektrálního družicového obrazu.

Jedním z populárních přístupů k hlubokému učení byla klasifikace oblastí se stejnými vlastnostmi, kde každý pixel byl odděleně zařazen do tříd pomocí okolí obrazového bodu kolem něj. Hlavním důvodem použití těchto blízkých oblastí bylo, že klasifikační síť mají obvykle plně připojené vrstvy, a proto vyžadují obrazy s pevnou velikostí.

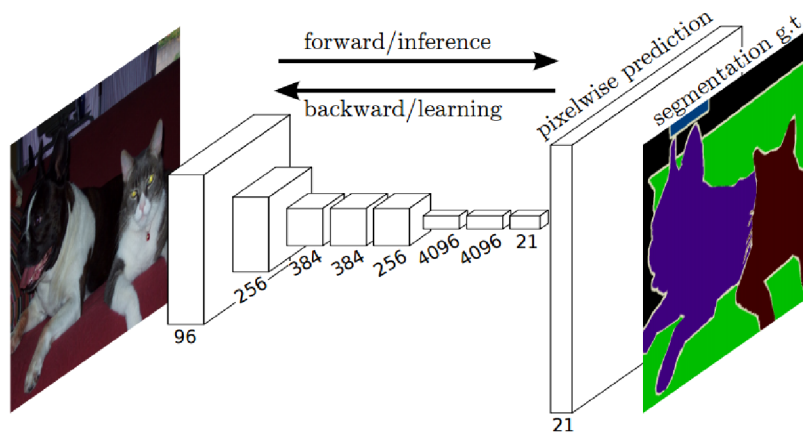
Až v roce 2014 byla popularizována architektura CNN pro husté předpovědi bez jakýchkoli plně propojených vrstev. To umožnilo vytvářet mapy segmentace pro obraz libovolné velikosti a bylo také mnohem rychlejší ve srovnání s přístupem klasifikace stejných oblastí. Kromě plně propojených vrstev je jedním z hlavních problémů při použití CNN pro segmentaci spojování vrstev. Sdružování vrstvy mají vlastnost, že vyřazují informaci o poloze bodu. Nicméně SS vyžaduje přesné zarovnání třídících map, a proto potřebuje informaci o poloze zachovat.

Tento problém je řešen architekturou dekodéru - kodéru. Kodér postupně snižuje prostorový rozměr spojováním vrstev a dekodér postupně obnovuje detaily objektu a prostorové rozměry. V následující části bude popsáno 6 základní architektur, které ukazují postupný vývoj segmentačních architektur [18] [29].

5.1 FCN

Plně propojené vrstvy (fully connected layers - FCN) v klasifikačních sítích lze považovat za konvoluční s jádry, které pokrývají celé vstupní oblasti. To odpovídá vyhodnocení původní klasifikační sítě na překrývajících se vstupních oblastech, ale je to mnohem efektivnější, protože výpočet je sdílen přes překrývající se oblasti se stejnými vlastnostmi. Funkce jsou sloučeny z různých fází kodéru, které se liší v hrubosti sémantických informací.

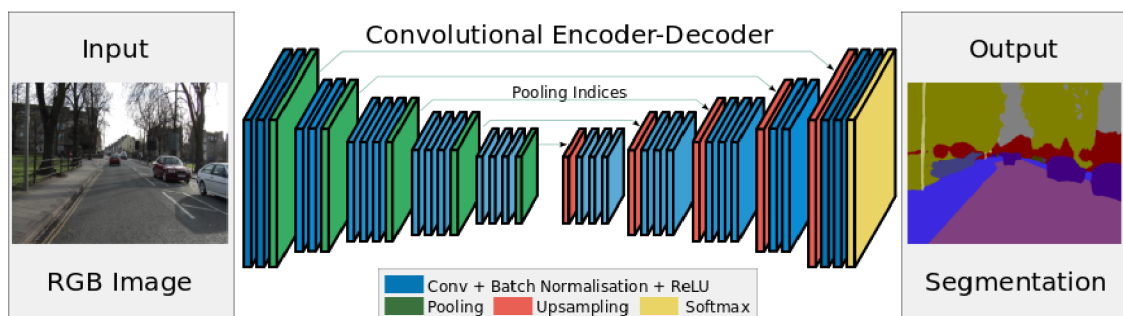
Po konvoluci plně propojených vrstev v před učené síti jako VGG, je mapy třeba stále vzorkovat kvůli sloučení operací v systémech CNN. Důležitým aspektem pro architekturu sémantické segmentace (obr. 23) je mechanismus používaný pro převzorkování segmentačních map s nízkým rozlišením pro vstupní rozlišení obrazu pomocí naučených dekonvolucí (jsou inicializovány pomocí bilineárních interpolačních filtrů) nebo částečné zamezení snížení rozlišení v kodéru pomocí dilatačních konvolucí [29].



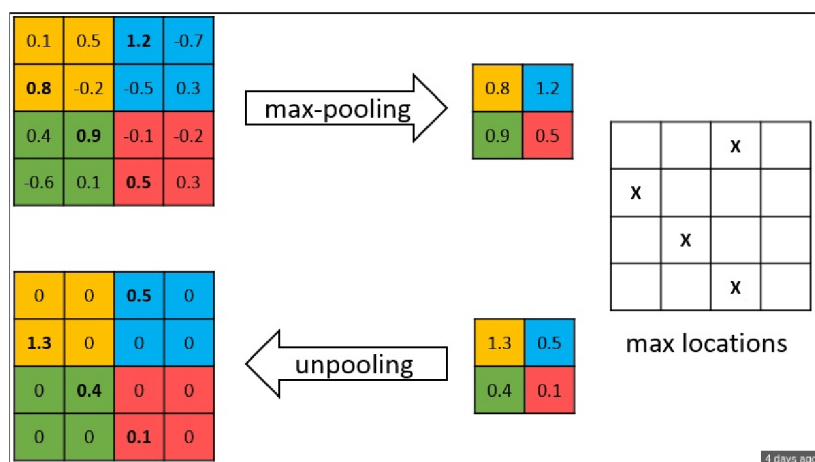
Obr. 23 Architektura FCN – plně propojené vrstvy [29]

5.2 SegNet

Změna u SegNet, oproti FCN, spočívá ve způsobu, jakým dekodér vzorkuje vstupní mapu s nízkým rozlišením a taky již zde nejsou plně propojené vrstvy. Konkrétně dekodér používá indexy sdružování vypočítané v “max-pooling“ odpovídajícího kodéru pro provádění nelineárního vzorkování, obr. 24. Eliminuje se tím potřeba učení se převzorkování. Vzorkované mapy jsou řídké, poté však projdou přes konvoluční vrstvu s tréninkovými filtry, které vytvářejí husté mapové vrstvy. Celý princip vidíme na obr. 25. Avšak tato architektura dosahuje horších výsledků než FCN nebo PSPNet [18].



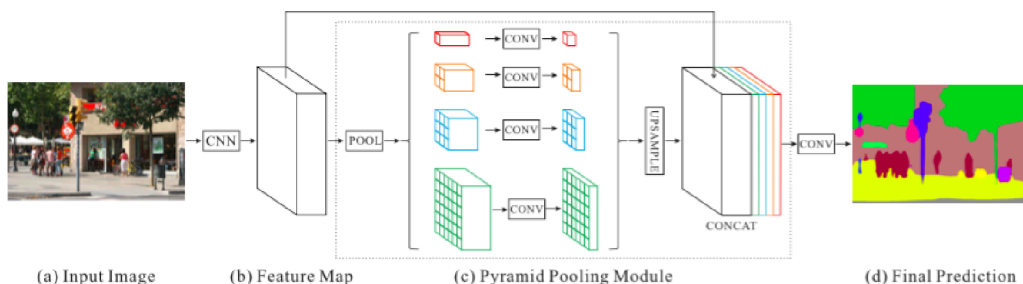
Obr. 24 Architektura sítě SegNet [18]



Obr. 25 SegNet - rozšiřování [29]

5.3 PSPNet

PSPNet modifikuje základní architekturu ResNet přidáním dilatační konvoluce a funkcí po inicializační sruřovací vrstvou. Síť analyzované pyramidové scény (pyramid scene parsing network - PSPNet) využívá schopnost globálního spojování informací z různě velkých regionů (viz obr. 26). Proto se to nazývá pyramidová scéna jdeme od velkých částí obrazu k menší, kde se již jen doladují detaily [29].

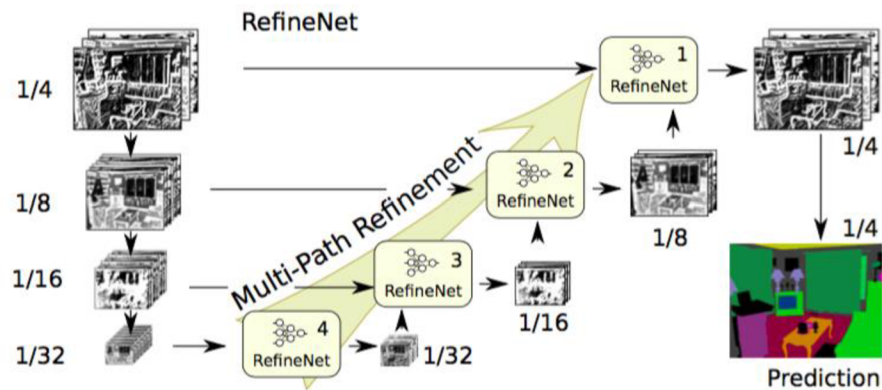


Obr. 26 Architektura síť PSPNet [29]

5.4 RefineNet

Obecná zdokonalovací síť s více cestami, to je RefiNet. Využívá všechny vstupy při více rozlišeních v průběhu procesu odběru vzorků. Spojí extrahované funkce a přenese je do další fáze, aby umožnila předpověď s vysokým rozlišením. Tímto způsobem mohou hlubší vrstvy, které zachycují sémantické prvky vyšší úrovně, být přímo použity k vyhlazení jemně zrnitého obrazu z předchozích konvolucí (obr. 27). Zavádí řetězové sruření, které dokáže zachytit kontext pozadí z větší oblasti obrazu. Jednotlivé části síť využívají zbytkové připojení k identifikování myšlenkových map, které umožňují efektivní konečné učení. Dekodér tedy spojuje bloky snímače s vysokým rozlišením

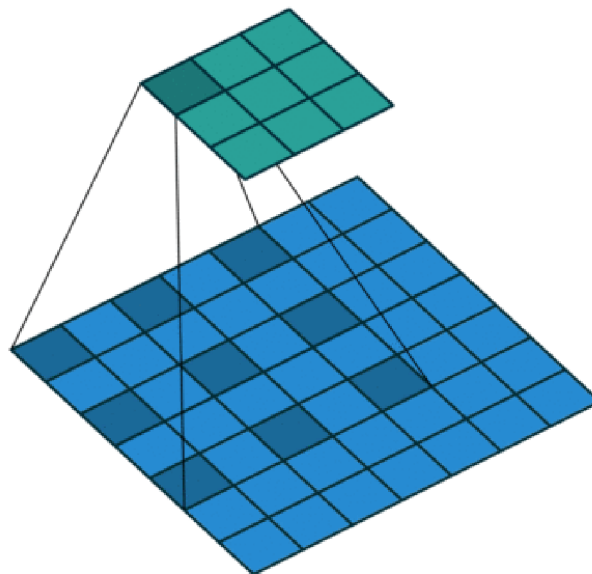
s bloky s nízkým rozlišením z předchozího bloku. Každý blok tedy propojuje více funkcí pro zachycení kontextu na základě shodných vrstev [18].



Obr. 27 Ukázka architektury sítě RefineNet [18]

5.5 Dilatační konvoluce

Sdružování snižuje rozlišení obrazu při segmentaci, proto tato architektura používá dilatační vrstvu konvoluce, která umožňuje exponenciální zvětšení zorného pole bez snížení velikosti obrazu, viz obr 28.



Obr. 28 Dilatační sdružovací vrstva [29]

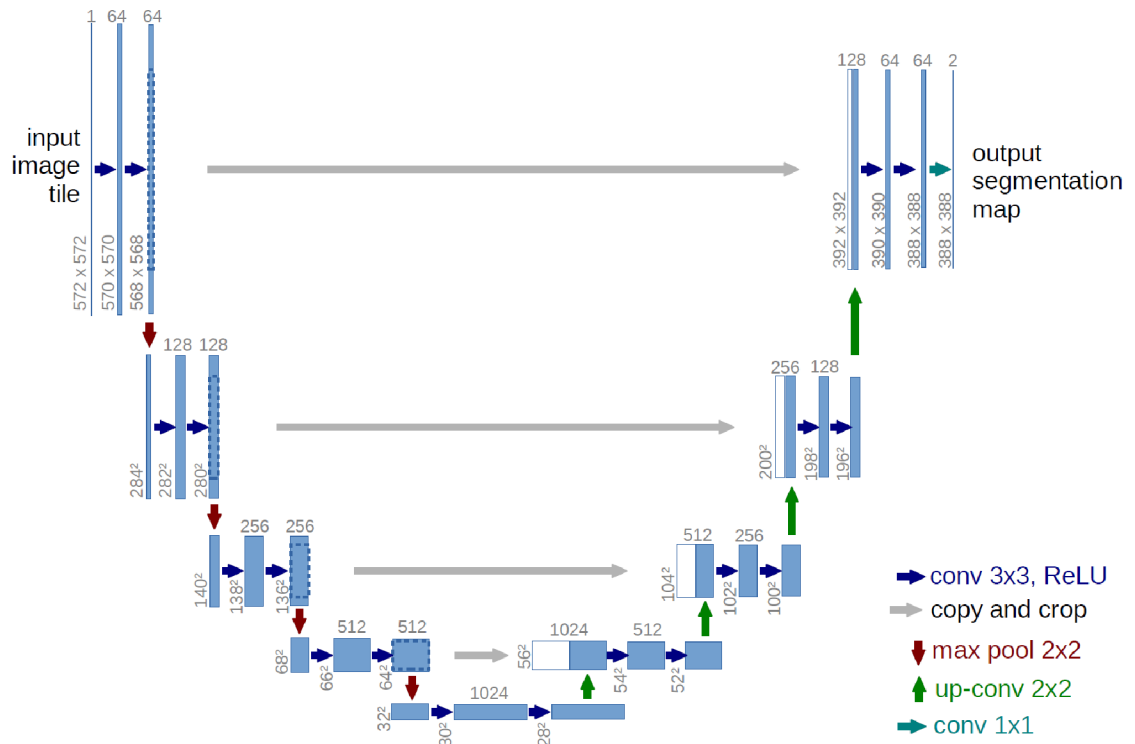
Poslední dvě sdružovací vrstvy jsou odstraněny a nahrazeny dilatační konvoluční vrstvou. Konvoluce mezi 3 a 4 vrstvou má dilataci 2, poté již je velikost dilatační

mezery čtyři. Tímto způsobem získáme husté mapy předpovědí bez zvýšení počtu parametrů [29].

5.6 U-Net

Architektura sítě U-Net se skládá ze zadávací cesty k zachycení kontextu a symetrické rozšiřovací cesty, která umožňuje přesnou lokalizaci obrazových bodů. Této síti stačí k naučení menší počet obrazů než předchozím sítím. Je také velmi rychlá, segmentace obrazu o velikosti 512x512 pixelů trvá méně než jednu sekundu na výkonné GPU.

U-Net jednoduše zřetězuje mapovací prvky kodéru s mapovacími vlastnostmi dekodéru v každé fázi, čímž vytváří strukturu podobnou žebříku, viz obr. 29. Architektura díky zkratkovým spojením umožňuje dekodéru v každé fázi učení získat relativní vlastnosti, které jsou ztraceny při sdužovací vrstvě v kodéru [29].



Obr. 29 Architektura sítě U-Net [29]

6 DATABÁZE OBRAZŮ

Zásadní částí pro naučení CNN jsou databáze obrazů. Podle toho jak velkou a kvalitní máme databázi nám do značné míry ovlivní i výsledné naučení sítě. Databáze obrazu určená k učení musí být rozdělena na dvě části. První části jsou originální obrazy, a ke každému obrazu je přiřazen jeden obraz kde je každý pixel zařazen do určité kategorie (anotace). Databáze CamVid byla použita jak pro trénink sítě, tak i testování. Sadu vlastních obrazů MTech, jsem použil pro testování naučených sítí.

6.1 Databáze CamVid

Databáze CamVid (Cambridge-driving Labeled Video Database) obrazy z reálného provozu. Tato databáze je k dispozici na webových stránkách [26].

Data byla pořízena kamerou umístěnou za čelním sklem auta, které se pohybovalo po silnicích anglického města. Deseti minutový video záznam byl převeden na jednotlivé obrazy v rozlišení 960x720 pixelů. Databáze obsahuje snímky v denních a večerních hodinách s jasným i oblačným počasím.

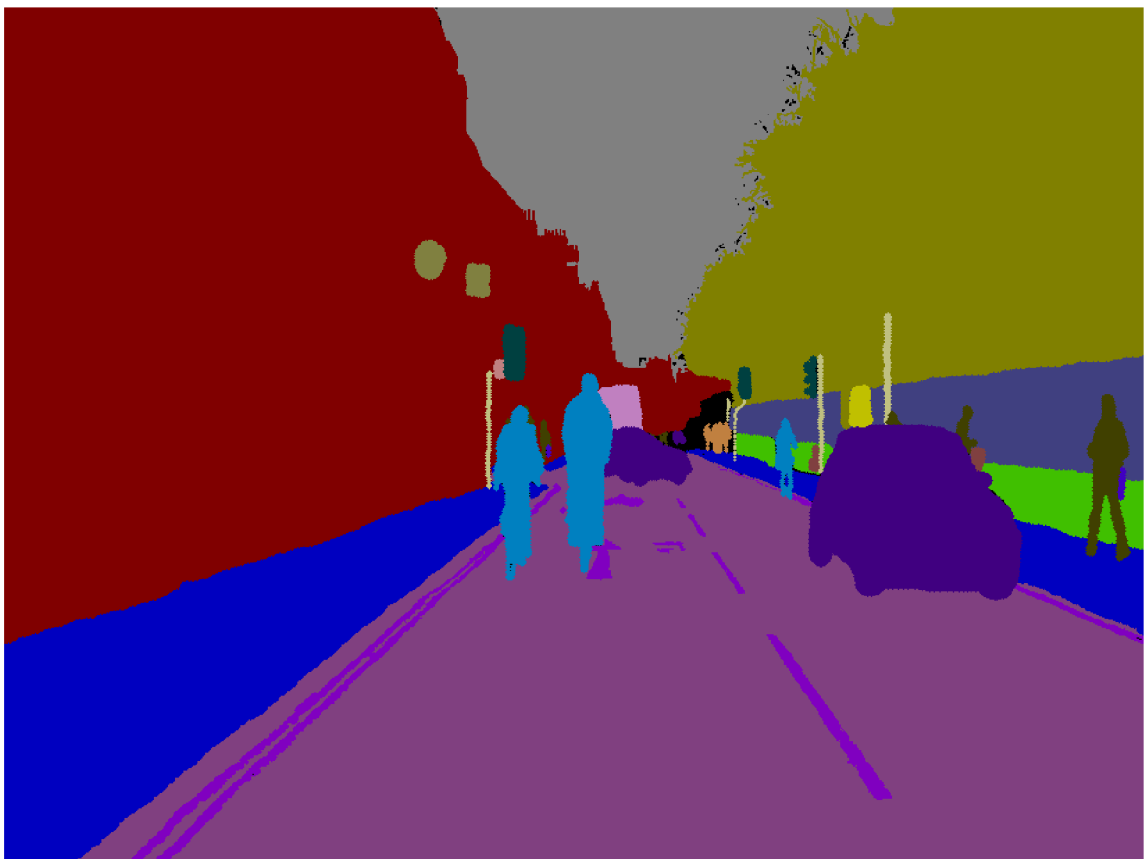
Databáze obsahuje celkem 701 obrazů získaných z videozáznamu (obr. 34) a k tomu odpovídající stejný počet anotovaných obrazů ve stejné velikosti (obr.35). Anotované obrazy řadí jednotlivé pixely do následujících 32 kategorií a byly vytvořeny ručně s velkou přesností.

Void	Building	Wall	Tree	VegetationMisc	Fence
Sidewalk	ParkingBlock	Column_Pole	TrafficCone	Bridge	SignSymbol
Misc_Text	TrafficLight	Sky	Tunnel	Archway	Road
RoadShoulder	LaneMkgsDriv	LaneMkgsNonDriv	Animal	Pedestrian	Child
CartLuggagePram	Bicyclist	MotorcycleScooter	Car	SUVPickupTruck	Truck_Bus
Train	OtherMoving				

Obr. 30 Rozdělení CamVid do 32 kategorií [26]



Obr. 31 CamVid – RGB [26]



Obr. 32 CamVid – Labels [26]

6.2 Databáze MTech

Pro účely testování naučených sítí jsem po dohodě s firmou Artin vytvořil vlastní sadu obrazu. Ověření funkčnosti i na snímcích které nebyly součástí tréninkové sady. Snímky byly pořízeny pomocí mobilního telefonu v rozlišení 3264x2448 pixelů. Tato velikost snímků byla samozřejmě příliš velká, proto jsem obrazy následně zmenšil. Vyfotil jsem 50 snímků v okolí FEKTu. Se zaměřením jednak na fakultu a její blízké okolí, tak i snímky z rychlostní silnice, která vede pod ní. Ukázka snímku je na obr. 33. Všechny snímky jsou v příloze na CD.



Obr. 33 Vlastní data MTech

7 IMPLEMENTACE SÍTĚ

Implementace sítě v této části představuje výběr a oživení CNN jako celku, kdy na vstup předložíme barevný obraz a na výstupu dostaneme segmentovaný obraz. Pojmem **celek** chápeme následující čtyři části, které bylo potřeba na začátku zvolit:

- a) databázi obrazů,
- b) před-učenou síť,
- c) architekturu CNN,
- d) skript, který dokáže upravit obrazy a provést proces učení.

Jelikož každá PT CNN má již vlastní architekturu, na které byla naučena, jeden výběr odpadá. Vybral jsem si vhodný skript, který zajišťoval jak proces učení i testování vhodné PT CNN tak i kompatibilní databázi obrazů. A to vše od společnosti Matlab. Tento skript i s odkazem na PT CNN je k nalezení na jejich webových stránkách [28]. Jako softwarový nástroj jsem tedy použil program Matlab R2017b, který byl k dispozici na školních počítačích. Mezi další důvody, proč jsem použil právě Matlab byla skutečnost, že obsahuje nástroje pro práci s obrazy a také UNS. Bylo potřeba upravit pár základních částí jako byly adresa uložení databáze obrazů, zvolit správné rozlišení vstupních dat, upravit počet kategorií, upravit vstupní a výstupní vrstvu sítě. Popis těchto úprav bude lépe objasněn v kapitole 7.2

7.1 Popis vybrané architektury

V teoretické části mé bakalářské práce popisují mnoho metod a architektur pro segmentaci obrazu. Pro praktickou aplikaci jsem si nakonec vybral sémantickou segmentaci obrazu s architekturou SegNet (viz kapitola 5.2).

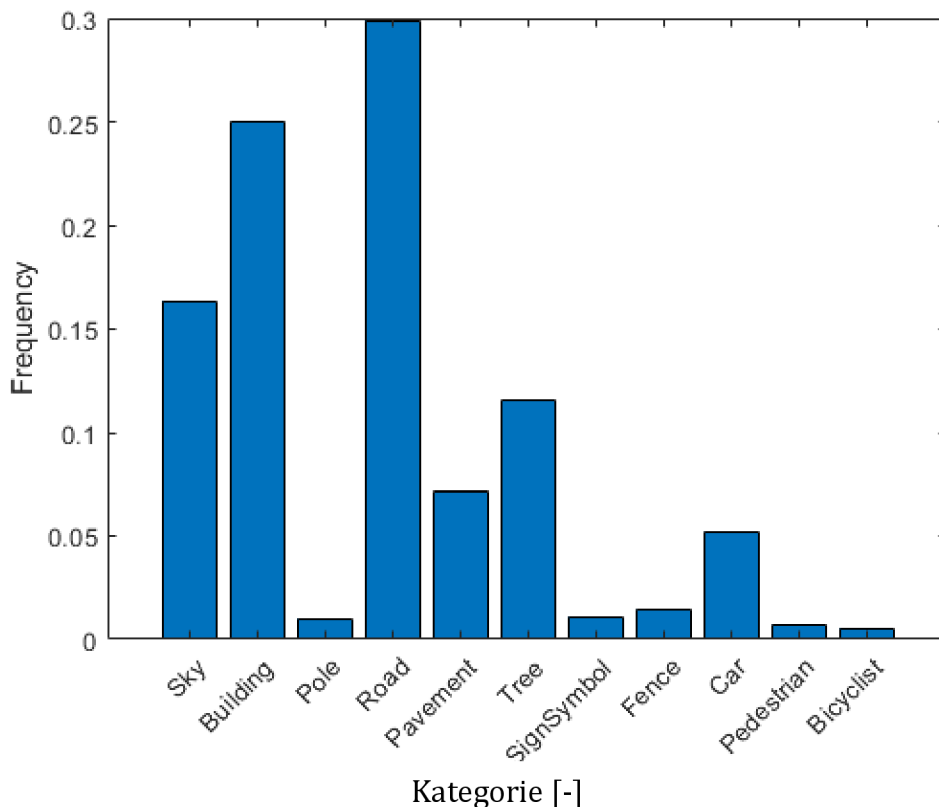
Jak již bylo řečeno v kapitole 4.5 učit celou síť od začátku je velmi časově náročné. Používají se tedy PT CNN které stačí doučit podle požadovaných dat. Ve své práci jsem použil před-učenou síť Vgg16 (viz kapitola 4.5 - VGGNet).

7.2 Rozdělení skriptu

Celý skript je rozdělen do několika částí, které na sebe navazují a postupně vykonávají načtení obrazových dat, jejich úpravu a předložení CNN pro učení. Na závěr také vykonají test úspěšnosti. Podrobný popis jednotlivých částí následuje zde:

- a) Nastavení cesty k umístění databáze obrazů a následné načtení obrazů do proměnné. Na snímky se použije funkce *histeq*, která vylepší kontrast pomocí vyrovnání histogramu.
- b) Vytvoření daného počtu kategorií a načtení popisů (labels) z databáze pro každý originální obraz. Každá datová sada obsahuje jiný počet kategorií. Cam Vid původně rozděluje obraz až do 32 kategorií, což je pro naše účely a vzhledem k malému výkonu GPU zbytečné, proto je zde pomocná funkce *camvidPixelLabelIDs*. Tato funkce zmenší počet kategorií z 32 na 11 a to tak, že například nerozlišuje již autobusy, nákladní auta, osobní auta, SUV atd. ale je to pouze jedna kategorie **auta**.
- c) Statistika celkového počtu pixelů rozdělených do kategorií v jednom testovacím obraze (viz obr. 34). Tento výpočet rozložení pixelů do jednotlivých kategorií je využit dále k přednastavení vah.
- d) Zmenšení velikosti obrazů, za použití pomocné funkce *resizeCamVidImages*. Tato funkce vytvoří adresář, do kterého se budou zmenšené obrazy ukládat. Následně načítá postupně snímky, zmenší jejich velikost pomocí funkce *imresize* na předem nastavenou hodnotu a uloží zpět do počítače již zmenšený obraz. Obdobně pracuje i funkce *resizeCamVidPixelLabels*, která navíc převede obraz do odstínu šedé (256 odstínů), tedy do bitové hloubky 8 Bytu.
- e) Rozdělení obrazů do tréninkové a testovací množiny funkcí *partitionCamVidData*, která náhodně rozdělí obrazy v daném poměru a uloží do nových proměnných.
- f) Vytvoření sítě načtením vybrané před-učené sítě, jako parametr se volí počet kategorií, do kterých budeme obraz rozdělovat. Dalším parametrem je velikost vstupní vrstvy, která udává velikost obrazů na vstupu. Výpočet váhy pomocí mediánu ze statistiky celkového počtu pixelů v jednotlivých kategoriích (viz bod 7.2.c)). Odstranění výstupní vrstvy, která sloužili pro klasifikaci objektů do 1000 skupin a nahrazení vrstvou, která bude klasifikovat pouze do menšího počtu kategorií jenž jsme si nastavily.
- g) Nastavení parametrů sítě – počet epoch, počáteční rychlost učení, velikost dávky, rozložení vah, hybnost (příspěvek gradientového kroku z předchozí iterace na aktuální iteraci tréninku), ukládání mezivýsledků (nikdy/ jednou/ po každé epoše), frekvence tisku výsledků do příkazového okna (Verbose Frequency), nastavené posunutí a rotace snímku při učení (augmenter) atd.

- h) Vytvoření tréninkové množiny (datové sady) pro jednoduché načítání dat při trénování. Jedná se o spojení tréninkové sady obrazů a popisů, plus nastavení možných rotací či posunutí obrazu.
- i) Zahájení samotného trénování. Jsou zde dvě možnosti, první je doučení sítě s našimi daty (proces učení, který trvá X hodin). Druhou je použití před-učené sítě tak jak je a pouhé vyhodnocení na testovací sadě obrazů.
- j) Vyhodnocení segmentace na jednom obraze. Načtení jednoho náhodného snímku z testovací sady, aplikace naučené sítě přes funkci *semanticseg*. Vykreslení sémantické mapy (podle předem zvolených barev jednotlivých kategorií) a nastavení úrovně průhlednosti, kdy při hodnotě 0 se zobrazí plná sémantická mapa. Porovnání aktuálního a očekávaného výsledků. Zobrazení, ve kterých pixelech se hodnota od správného výsledku liší.
- k) Vyhodnocení celé testovací množiny obrazů opět s pomocí funkce *semanticseg*, které se nastaví další parametry (ukládání mezivýsledků, velikost dávky atd.). Porovnání těchto segmentovaných obrazů s danými anotacemi funkcí *evaluateSemanticSegmentation*, která vypočítá různé metriky hodnocení výsledků. Máme k dispozici jak metriky celé testovací sady, tak i úspěšnost zařazení pixelů do jednotlivých kategorií. Zde lze očekávat, že pokud se např. kolo v datové sadě vyskytuje minimálně nebude dosahovat dobrých výsledků, protože síť neměla dostatek těchto obrazů k učení.



Obr. 34 Statistika počtu pixelu v kategoriích

7.3 Výpočetní výkon

K zahájení učení sítě je zapotřebí dostatečný výkon počítače. Hlavním parametrem pro měření výkonu počítače je v oblasti CNN paměť GPU. Máme však několik možností, jak výpočetní výkon zvýšit, aniž bychom si museli pořídit lepší GPU.

- a) zmenšit rozlišení obrazů,
- b) použít výkon výpočetního serveru, který je ve škole,
- c) pronajmout si část výkonu na serverech Amazonu (či jiném poskytovateli),
- d) propojit více počítačů do clusteru – jeden počítač bude řídit a rozdělovat úkoly (výpočty) dalším strojům v síti,

Měl jsem k dispozici laboratorní počítač, na kterém byla k dispozici pouze běžnou GPU (viz tabulka 1), vybral jsem si možnost a) zmenšit vstupní obrázky. Z původního rozlišení 960x720px bylo potřeba zmenšit až na velikost 240x180px, aby se daná dávka obrazů vůbec vešla do paměti GPU.

GPU	NVIDIA GeForce GT 630 Paměť 1024 MB, takt 700MHz
CPU	AMD Fx-6350, 6 CPUs – 3,9GHz
RAM	8 GB

Tabulka 1 Parametry použitého Hardwaru

Učení sítě má i velké časové nároky. Při učení na GPU, kterou jsem měl k dispozici ve škole a velikosti dávky 3 (Mini Batch Size) trvá jedna epocha 19,5 minuty. Jednotlivé epochy se můžou časovou náročností lehce odlišovat, ale celková doba učení při počtu 100 epoch trvá přibližně 30 hodin. Musíme vzít v potaz, že učení probíhá pouze na 500 tréninkových snímcích a snažíme se pouze doučit již naučenou síť. Kdybychom chtěli začít učit síť od nuly bylo by k tomu potřeba až několik desítek tisíc obrazů, na tom můžeme vidět, jak velmi časově náročný to je proces. Proto tak velmi záleží na výkonnosti HW a především GPU. I při velmi výkonném HW trvá učení celé sítě od nuly až několik dnů ne-li týdnů.

7.4 Výsledky učení

Ve své bakalářské práci jsem provedl desítky učení různých sítí s rozdílnými parametry, to vše kvůli vytvoření statistiky a vyhodnocení, při jakých parametrech dostaneme nejlepší výsledné obrázky. Cílem celé sémantické segmentace obrazu je naučit síť tak, aby ve výsledném obraze bylo co nejméně pixelu zařazeno do špatné kategorie a aby toto zpracování obrazu bylo zároveň co nejrychlejší.

7.4.1 Vysvětlení pojmů

Poté co byl proces učení úspěšně dokončen, bylo provedeno testování sítě. Předložil jsem testovací množinu obrazů, kde se zjišťovalo, jak dobře se neurony naučily segmentovat obraz do jednotlivých kategorií. Výsledkem tohoto testování je 5 hodnot v rozmezí 0-1, kdy 1 znamená naprostou shodu a 0 žádnou shodu (neúspěch). Tyto hodnoty jsou následně podrobně popsány [9].

- a) **Global Accuracy** - poměr správně zařazených pixelů, bez ohledu na kategorii, ku celkovému počtu pixelů. Používá se pro rychlý odhad procenta správně zařazených pixelů.
- b) **Mean Accuracy** – průměr správně identifikovaných pixelů ze všech kategorií. Přesnost se počítá podle vzorce

$$TP/(TP + FN) \quad (7.1)$$

TP ... správně identifikováno (True positive)

FN ... chybně zamítnuto (False negative)

FP ... chybně identifikováno (False positive)

- c) **Mean IoU** – Průměrné skóre průniku nad množinou v jednotlivých kategoriích, je jednou z nejčastěji používaných metrik. Používá statistické měření přesnosti, které postihuje chybně identifikované pixely. Pro každou třídu IoU (Interaction over Union) je poměr správně zařazených pixelů na celkovém počtem správně předpovězených pixelů v dané třídě. Pokud je hodnota IoU větší nebo rovna 0,5 je výsledek považován za správný.

$$IoU = TP/(TP + FP + FN) \quad (7.2)$$

- d) **Weighted IoU** – průměr IoU nad každou kategorií vážený podílem počtu pixelů. Pokud mají obrazy nepřiměřené rozdíly ve velikosti mezi kategoriemi, snažíme se snížit dopad chyb v malých třídách na hodnocení celkové kvality, použijte tedy tuto metriku.
- e) **Mean BF score** – Průměrná hodnota obrysu hranic objektů ukazuje, jak dobře předpovídané hranice souhlasí se skutečnými hranicemi. Toto vyhodnocení se používá, pokud chceme metriku, která lépe koreluje s lidským hodnocením kvality než IoU metrika.

7.4.2 Statistiky výsledků učení

Mým prvním krokem bylo zjistit, jak velká je závislost velikosti obrazu na výslednou přesnost sítě. Již v kapitole 7.3 jsem popsal nutnost zmenšení rozlišení vstupních dat. Začal jsem opravdu od velmi malého rozlišení 45x60 a postupně zvyšoval až na 180x240. Jak lze vidět v grafu č.1. Další dvě hodnoty (tedy rozlišení 360x480 a 450x600), již nepochází přímo z učení sítě, ale pouze testu před-učené sítě, kterou jsem již nijak nedoučoval. Tyto hodnoty tedy nelze brát moc v potaz. Z výsledku lze usoudit, že tato před-učená síť je opravdu dobrá pro tuto úlohu.

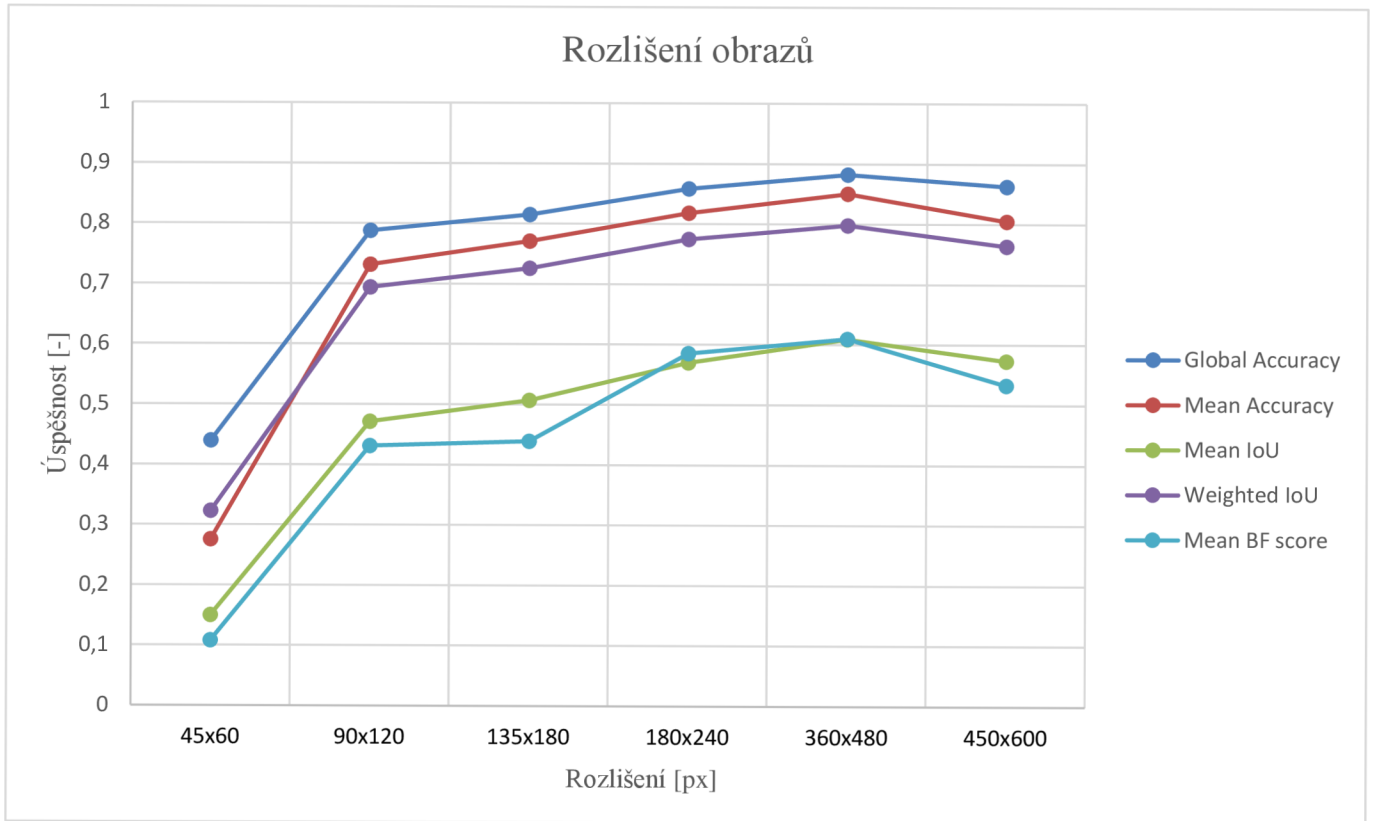
Zkoumal jsem vliv počtu epoch (viz 3.3.2) na přesnost výsledku. Sledoval jsem tedy jak dlouho je zapotřebí učit síť. Začal jsem od 25 opakování a postupně zvyšoval až na 200 opakování. Z grafu č. 2 vidíme, že rostoucí tendenci mají výsledky stále, jak lze očekávat, ale od 75 epoch již není mezi body takový rozdíl. Tedy jako výsledek tohoto testu bych konstatoval, že 100 opakování je dostatečné a není potřeba již dalších průchodů.

Věděl jsem již, že na větší rozlišení, než je 180x240 se nedostanu s aktuálně používanou GPU. Začal jsem upravovat parametry učení, abych zjistil, při jakých hodnotách lze dosáhnout nejlepších výsledků. Ovšem kvůli ušetření času jsem nastavoval pouze 50 epoch. Jak již bylo vysvětleno v kapitole 7.3 učení sítě je velice náročné na čas. I těchto 50 epoch trvalo průměrně 15 hodin. Provedl jsem test se 3 parametry, které jsem upravoval.

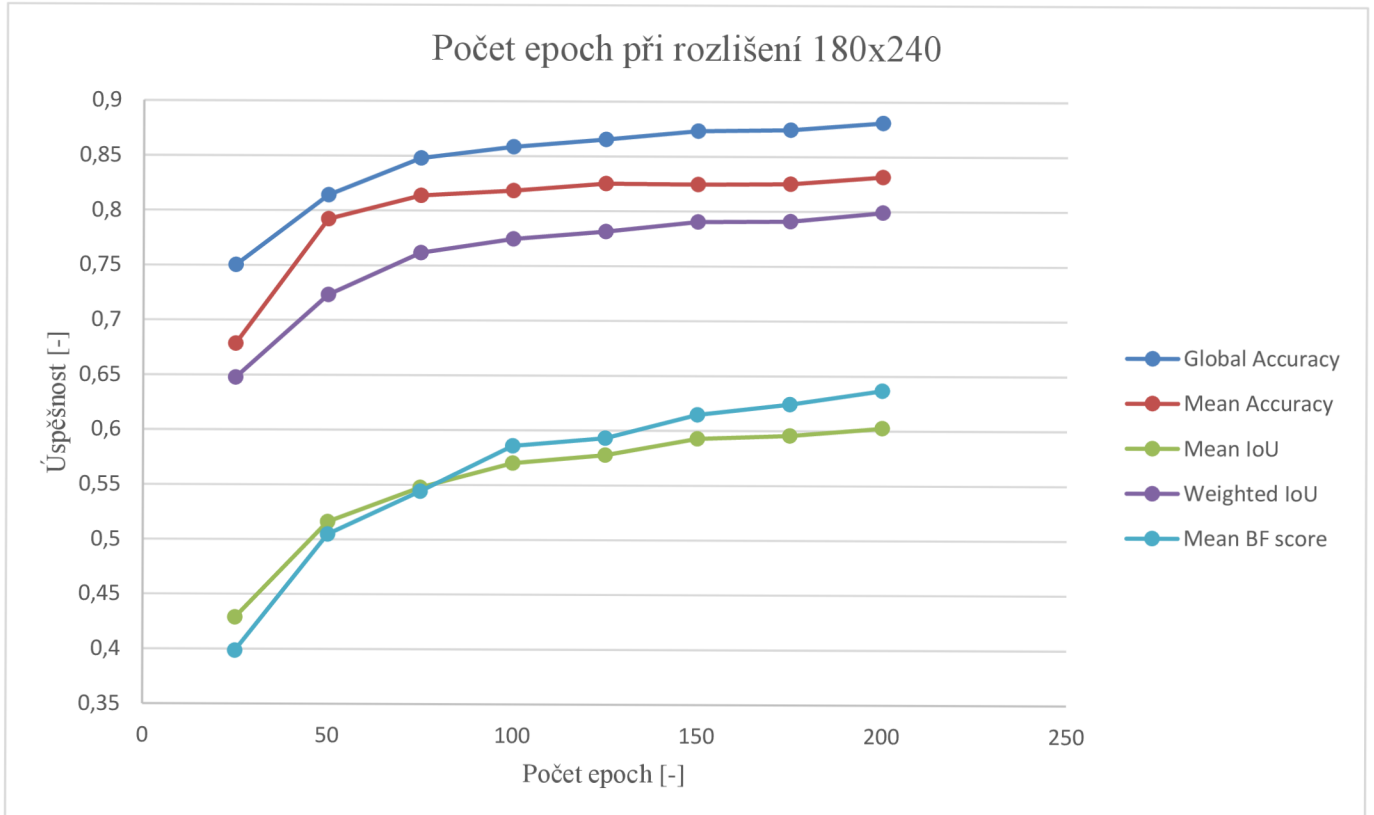
Prvním z testu bylo nastavování parametru **rozložení vah** (L2 Regularization). Faktor L2 (hmotnostní rozklad), specifikovaný jako dvojice hodnot. Můžeme určit jako násobitel pro prvky s parametry. [32]. Výchozí přednastavená hodnota byla $5e-4$, upravoval jsem tedy tuto hodnotu a hledal nejlepší výsledek. Ten jsem našel hned při hodnotě $5e-3$ a se zvětšováním hodnoty začala úspěšnost prudce klesat (viz graf č.3).

Druhým testem byl na závislost parametru **počáteční rychlosti učení** (Initial Learn Rate). Vycházel jsem z hodnoty $1e-3$, při zvýšení na hodnotu $1e-2$ byla úspěšnost největší a při další zvyšování této hodnoty již úspěšnost prudce klesá (viz graf č.4). Pokud je míra učení příliš nízká trénink trvá moc dlouhou dobu. Je-li míra učení naopak příliš vysoká nemusí trénink dosáhnout správných výsledků [32].

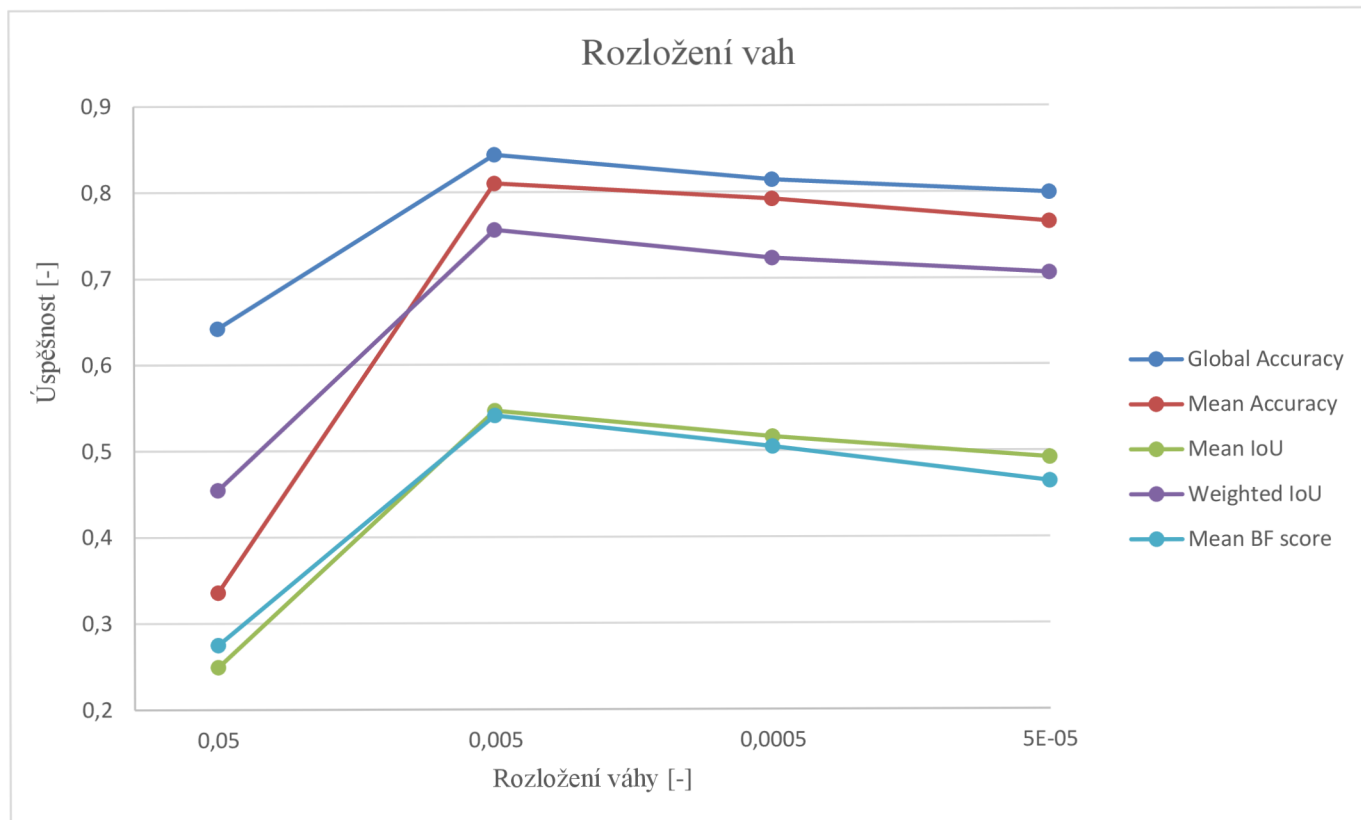
Třetím a posledním parametrem, jenž jsem testoval pro zjištění nejlepšího nastavení při učení sítě, byl poměr rozložení obrazů mezi tréninkovou a testovací množinu. Hodnota na ose X udává procentuální vyjádření množství obrazů pro tréninkovou množinu. Výchozí hodnotou bylo 60 což znamená, že 60 % obrazu z celkové sady je náhodně vybráno do tréninkové množiny a zbylých 40 % je přiřazeno do testovací množiny. Výsledky jednotlivých testů se od sebe moc neliší, avšak testoval jsem pouze na 3 hodnotách (viz graf č.5). Pokud by se udělalo více testu šlo by určit lepší hodnotu rozložení. Na závěr bych konstatoval, že tento parametr nemá moc velký vliv na učení v těchto středních hodnotách. Pokud bychom však nastavili okrajové hodnoty, úspěšnost by klesala rychle dolů.



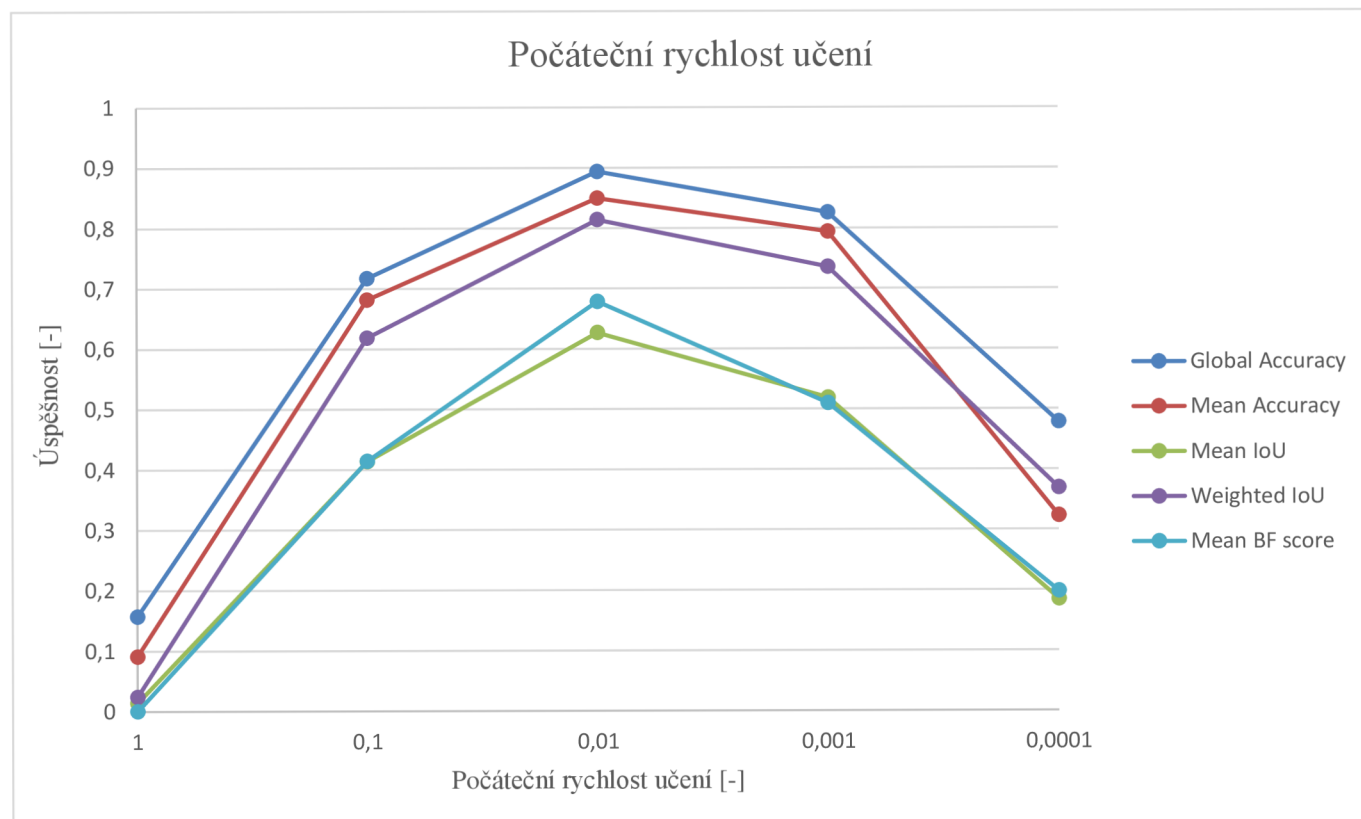
Graf 1 Rozlišení obrazů



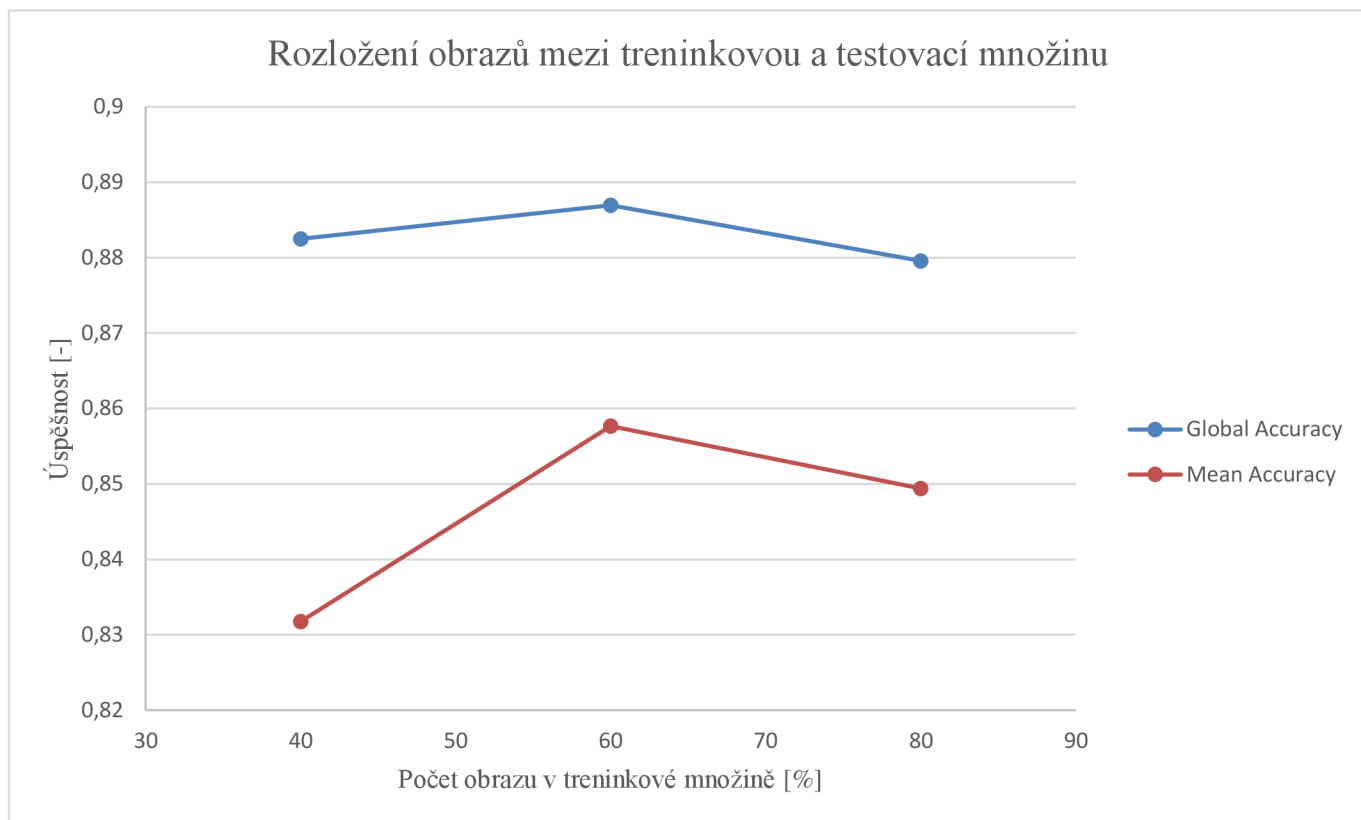
Graf 2 Závislost počtu Epoch



Graf 3 Rozložení vah

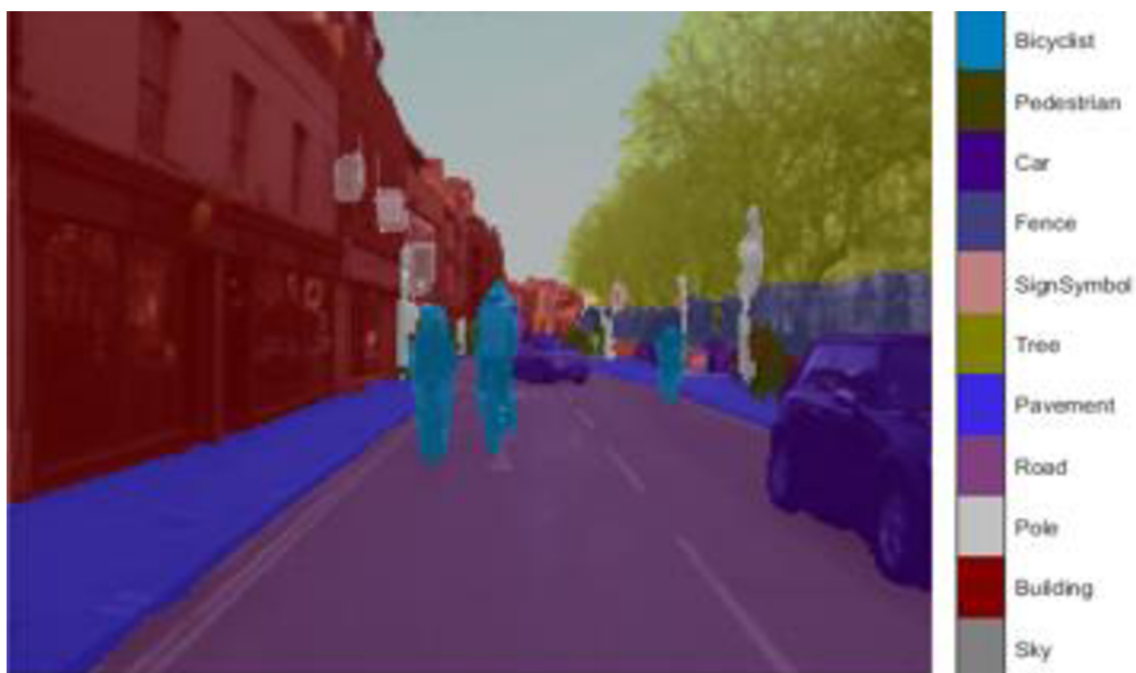


Graf 4 Počáteční rychlost učení



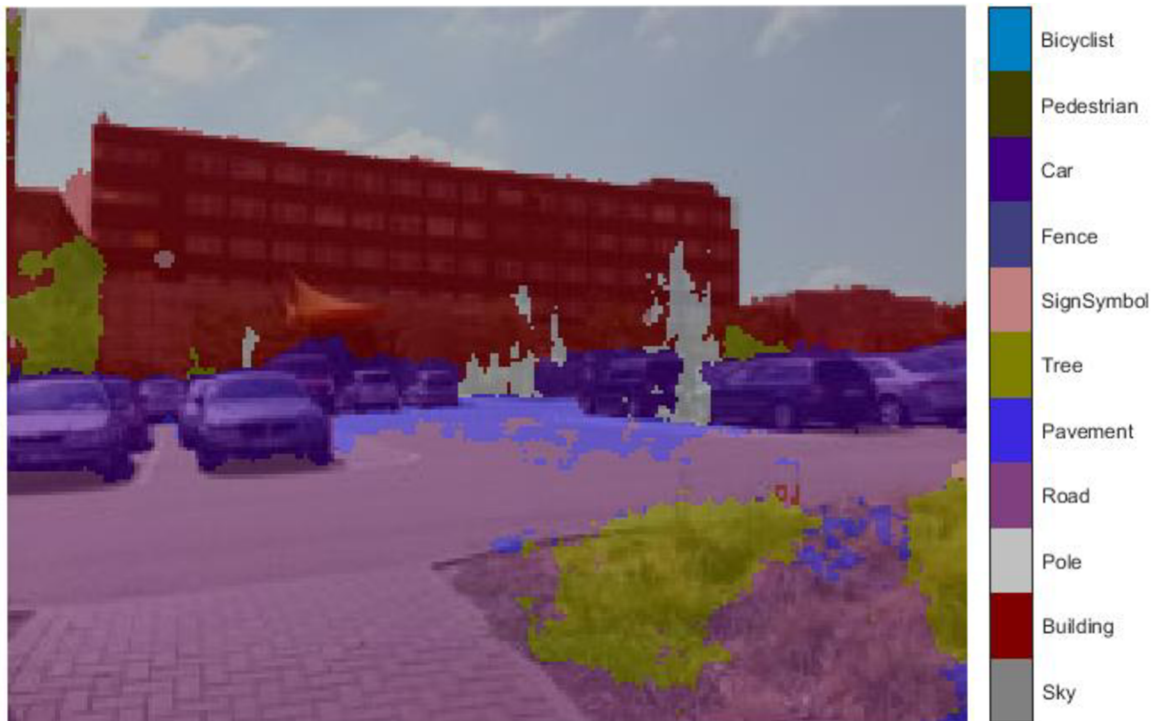
Graf 5 Rozdělení obrazů

7.4.3 Segmentované obrazy



Obr. 35 CamVid - segmentace

Segmentovaný obraz 37 je z databáze CamVid. Segmentace dopadla lépe než u obrazu z databáze MTEch. Důvodem lepšího výsledku je fakt, že se na obrazech z databáze CamVid se CNN doučovala a poté i testovala.



Obr. 36 MTEch - segmentace

Na obr. 38 je má fotografie, která prošla přes naučenou síť a toto je výsledná segmentace. Jsou na ní vidět jisté chyby ale vzhledem k tomu, že se síť nemohla učít ze žádného velmi podobného obrazu je výsledek velmi dobrý. Touto segmentací jsem si otestoval funkčnost a kvalitu naučení mé CNN. Potvrdil tak, že zvládne segmentovat i odlišné obrazy, než na kterých se prováděl proces učení.

8 ZÁVĚR

Při výběru mé semestrální práce mne zaujalo téma konvolučních neuronových sítí a jejich využití při segmentaci obrazu. Jedná se o rozvíjející se oblast autonomního řízení, která mne velmi zajímá. Během mého studia jsem se ve škole nesetkal s předmětem, který by se zabýval podobným tématem. Z tohoto důvodu jsem si musel získat informace samostudiem a následnou konzultací s vedoucím práce.

V práci jsem popsal celý řetězec zpracování obrazu v počítačovém vidění, kde jsem se zaměřil hlavně na část segmentace (kapitola 2). Klasifikaci zajišťuje NS, ale jelikož pracuji s obrazovými daty zpracoval a pochopil jsem také funkci CNN (kapitola 4). Kapitola 5 jsem věnoval architekturám sémantické segmentaci, jelikož jsem si následně jednu vybral a použít pro vlastní aplikaci.

Měl jsem k dispozici běžnou GPU, proto jsem tento problém vyřešil zmenšením obrazů. Při testování maximální velikosti jsem se dostal na 180x240 pixelů větší obrazy se již do paměti GPU nevlezly (viz kapitola 7.3).

Cílem práce bylo vybrat a použít vhodnou CNN pro segmentaci obrazu. Vybral jsem si před-učenou CNN Vgg16 s architekturou SegNet (kapitola 5.2). Abych upravil vstupní obrazy, nastavil parametry sítě a započal proces učení, využil jsem skriptu od Matlabu.

Navíc jsem provedl soubor testů, kdy jsem nastavoval parametry učení a hledal optimální hodnoty pro získání co nejlepších výsledků. Při tomto testování jsem zjistil, které parametry mají na výsledky největší vliv a jakou nastavit jejich hodnotu. Všechny výsledky těchto testů jsou v kapitole 7.4.2. Největší vliv na úspěšnost učení mají parametry **rozložení vah** a **počáteční rychlost učení**.

Výsledkem mé práce je CNN pro sémantickou segmentaci, kterou jsem doučil pomocí databáze CamVid. Otestoval na testovací sadě obrazů (část z databáze CamVid, která nebyla použita pro trénink). Navíc jsme ověřil funkčnost CNN i na vlastních obrazech z databáze MTech kterou jsem za tímto účelem vytvořil. Výsledky jsou na dobré úrovni vzhledem k výkonnosti HW a počtu obrazu pro doučení (viz kapitola 7.4.3).

Nyní již mám teoretické a částečně i praktické základy v oblasti sémantické segmentace. V diplomové práci bych na toto téma mohl navázat praktickým využitím např. v automobilu a segmentovat obraz v reálném čase.

Literatura

- [1] BERAN, Vít'a. *Co a jak dnes počítače vidí?* [online]. VUT FIT Brno, 2015 [cit. 2018-01-02]. Dostupné z: <https://www.superlectures.com/openalt2015/co-a-jak-dnes-pocitace-vidi>
- [2] Biologické algoritmy (5) - Neuronové sítě. *ROOT.cz* [online]. 14.6.2012 [cit. 2018-01-02]. Dostupné z: <https://www.root.cz/clanky/biologicke-algoritmy-5-neuronove-site/>
- [3] *Convolutional Neural Network (CNN) basics* [online]. [cit. 2018-01-02]. Dostupné z: <https://pythonprogramming.net/convolutional-neural-network-cnn-machine-learning-tutorial/>
- [4] *CS231n Convolutional Neural Networks for Visual Recognition* [online]. 2017 [cit. 2017-12-30]. Dostupné z: <http://cs231n.github.io/convolutional-networks/>
- [5] *Deep learning pro segmentaci obrazu* [online]. HUMUSOFT, 5.11.2017 [cit. 2017-12-30]. Dostupné z: <http://sciencemag.cz/deep-learning-pro-segmentaci-obrazu/>
- [6] Deep Learning v prostředí Matlab. *AUTOMA: časopis pro automatizační techniku* [online]. (Humusoft s. r. o.), 2017(05) [cit. 2017-12-30]. Dostupné z: http://www.automa.cz/cz/casopis-clanky/deep-learning-v-prostredi-matlab-2017_05_0_10482/
- [7] Deep Learning v prostředí Matlab: systémy strojového vidění; identifikace zboží a osob. *AUTOMA* [online]. Humusoft s. r. o, 2017, (5), 12-14 [cit. 2018-01-02]. Dostupné z: http://www.automa.cz/Aton/FileRepository/pdf_articles/10482.pdf
- [8] DESHPANDE, Adit. *A Beginner's Guide To Understanding Convolutional Neural Networks* [online]. 20.6.2016 [cit. 2018-01-02]. Dostupné z: <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>
- [9] *Evaluate Semantic Segmentation* [online]. 2018 [cit. 2018-08-08]. Dostupné z: <https://nl.mathworks.com/help/vision/ref/evaluatesemanticsegmentation.html>
- [10] *Framework Torch: problematika rozpoznávání a klasifikace obrázků* [online]. 7.12.2017 [cit. 2018-01-02]. Dostupné z: <https://www.root.cz/clanky/framework-torch-problematika-rozpoznavani-a-klasifikace-obrazku/#k17>
- [11] *Framework Torch: využití konvolučních sítí pro rozpoznávání a klasifikaci obrázků* [online]. 14.12.2017 [cit. 2018-01-02]. Dostupné z:

<https://www.root.cz/clanky/framework-torch-vyuziti-konvolucnich-siti-pro-rozpoznavani-a-klasifikaci-obrazku/>

[12] HANZLÍK, Ondřej. *Automatické rozpoznávání stavu elektroměru z fotografie* [online]. České Budějovice, 2015 [cit. 2018-04-15]. Dostupné z: <https://theses.cz/id/i7tzbq/dp-hanzlik.pdf>. Diplomová práce. Jihočeská univerzita v Českých Budějovicích Přírodovědecká fakulta. Vedoucí práce Ing. Miroslav Skrbek, Ph.D.

[13] HORÁK, Karel. *Introdustion to Convolutional Neural Networks* [pdf]. VUT, 2017 [cit. 2018-01-03]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ROZ/roz_cz.php

[14] HORÁK, Karel, JANÁKOVA, Ilona. *Počítačové vidění: optika v počítačovém vidění* [online]. 2017 [cit. 2018-03-14]. Dostupné z: http://midas.uamt.feec.vutbr.cz/POV/pov_cz.php

[15] HORÁK, Karel, JANÁKOVA, Ilona. *Počítačové vidění: popis objektů* [online]. 2017 [cit. 2018-03-14]. Dostupné z: http://midas.uamt.feec.vutbr.cz/POV/pov_cz.php

[16] CHEN, Liang. *DRINet for Medical Image Segmentation* [online]. 10.05.2018 [cit. 2018-05-13]. DOI: 10.1109/TMI.2018.2835303. Dostupné z: <https://ieeexplore.ieee.org/document/8357580/>

[17] CHEVALIER, GUILLAUME. *Satellite Image Segmentation: a Workflow with U-Net* [online]. 24 August 2017 [cit. 2018-05-13]. Dostupné z: <https://vooban.com/en/tips-articles-geek-stuff/satellite-image-segmentation-workflow-with-u-net/>

[18] CHILAMKURTHY, Sasank. *A 2017 Guide to Semantic Segmentation with Deep Learning* [online]. July 5, 2017 [cit. 2018-05-09]. Dostupné z: <http://blog.quare.ai/notes/semantic-segmentation-deep-learning-review>

[19] JIRKOVSKÝ, Jaroslav. *TCC 2017: Deep Learning (a Computer Vision)* [online]. In: . Brno, 7.9.2017 [cit. 2018-01-02]. Dostupné z: http://www2.humusoft.cz/www/papers/tcc17/TCC2017_Prednaska_Deep_Learning.pdf

[20] JIRSÍK, Václav. *Umělé neuronové sítě - základy: Moderní prostředky v automatizaci* [online]. In: . Březen 2018 [cit. 2018-05-06].

[21] KALOVÁ, Ilona. *Předzpracování obrazu: Řetězec zpracování obrazu* [online]. [cit. 2018-03-14]. Dostupné z: <http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/02%20-%20Predzpracovani%20obrazu.pdf>

- [22] *Koncept umělé neuronové sítě* [online]. [cit. 2018-01-03]. Dostupné z: <http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--uvod-do-neuronovych-siti--koncept-umele-neuronove-site>
- [23] KRAJČOVIČOVÁ, Mária. *Konvoluční neuronová síť pro zpracování obrazu* [online]. 2015 [cit. 2018-01-02]. Dostupné z: <https://core.ac.uk/download/pdf/30299309.pdf>. Diplomová práce. VUT FEKT Brno. Vedoucí práce Doc. Ing. Radim Burget, Ph.D.
- [24] *Matematický model a aktivní dynamika neuronu* [online]. [cit. 2018-01-02]. Dostupné z: <http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--jednotlivy-neuron--matematicky-model-a-aktivni-dynamika-neuronu>
- [25] *Metody rozpoznání objektů v obraze* [online]. In.: s. 7 [cit. 2018-03-25]. Dostupné z: <http://www.fbmi.cvut.cz/files/predmety/3528/public/Metody%20rozpozn%C3%A1n%C3%AD%20objekt%C5%AF%20v%20obrazu.pdf>
- [26] *Motion-based Segmentation and Recognition Dataset: CamVid Database* [online]. University of Cambridge, 2008 [cit. 2018-08-07]. Dostupné z: <http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/#ClassLabels>
- [27] NOVOTNÝ, Stanislav. *Detekce mobilního robotu zpracováním obrazu* [online]. Brno, 2015 [cit. 2018-04-15]. Dostupné z: <https://core.ac.uk/download/pdf/30308035.pdf>. Diplomové práce. VUT Fakulta strojního inženýrství. Vedoucí práce Ing. JIŘÍ KREJSA, Ph.D.
- [28] *Semantic Segmentation Using Deep Learning* [online]. [cit. 2018-08-07]. Dostupné z: <https://nl.mathworks.com/help/vision/examples/semantic-segmentation-using-deep-learning.html>
- [29] *Semantic Segmentation using Fully Convolutional Networks over the years* [online]. 01.06.2017 [cit. 2018-05-09]. Dostupné z: https://meetsah1995.github.io/semantic-segmentation/deep-learning/pytorch/visdom/2017/06/01/semantic-segmentation-over-the-years.html#sec_fcn
- [30] SONKA, Milan, Václav HLAVAC a Roger BOYLE. *Image Processing, Analysis, and Machine Vision*. 3. edice. Toronto: Thomson, 2008. ISBN 13: 978-0-495-08252-1.

[31] ŠONKA, Milan a Václav HLAVÁČ. *Počítačové vidění*. Praha: Grada a.s, 1992. ISBN 80-85424-67-3.

[32] *Training Options: Matlab* [online]. 2016 [cit. 2018-08-05]. Dostupné z: <https://nl.mathworks.com/help/nnet/ref/trainingoptions.html>

[33] *Učení bez učitele* [online]. [cit. 2018-01-02]. Dostupné z: <http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--adaptacni-dynamika-neuronu--uceni-bez-ucitele>

[34] *Učení s učitelem* [online]. [cit. 2018-01-02]. Dostupné z: <http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--adaptacni-dynamika-neuronu--uceni-s-ucitelem>