



# 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



# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**  
Ústav automatizace a měřicí techniky

**Student:** Michal Mitrenga

**ID:** 186141

**Ročník:** 3

**Akademický rok:** 2017/18

## NÁZEV TÉMATU:

### Konvoluční neuronová síť pro segmentaci obrazu

#### POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s problematikou konvolučních neuronových sítí a segmentací obrazů.
2. Prostudujte použití konvolučních neuronových sítí pro segmentaci obrazu.
3. Ve spolupráci s firmou Artin s.r.o. vytvořte databázi obrazů.
4. Pro segmentaci obrazů použijte vybranou konvoluční neuronovou síť.
5. Dosažené výsledky zhodnoťte.

#### DOPORUČENÁ LITERATURA:

ZAPLETAL, Ondřej. Rozpoznávání obrazů konvolučními neuronovými sítěmi - základní koncepty. Brno 2017. Diplomová práce. VUT FEKT v Brně. Ústav automatizace a měřicí techniky. Vedoucí práce Karel HORÁK.

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 21.5.2018

**Vedoucí práce:** doc. Ing. Václav Jirsík, CSc.

**Konzultant:** Ing. Karel Horák, Ph.D.

**doc. Ing. Václav Jirsík, CSc.**  
*předseda oborové rady*

#### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Cílem bakalářské práce je seznámit se s problematikou konvolučních neuronových sítí pro segmentaci obrazu. Toto téma v sobě zahrnuje celý obor počítačového vidění. 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. Součástí práce je ukázka praktických aplikací segmentace obrazu. Důležitou částí je databáze obrazů SYNTHIA, kde jsou ukázány její vlastnosti. V závěru práce jsou podrobněji popsány podmínky a požadavky na výkon hardwaru a potřebný software pro dobrý chod sítě. Bylo využito frameworku Keras, který již obsahuje funkce pro práci s neuronovými sítěmi.

## **KLÍČOVÁ SLOVA**

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

## **ABSTRACT**

The aim of the bachelor thesis is to learn more about the problem of convolutional neural networks for image segmentation. This theme encompasses the whole field of computer vision. Particular 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. Part of the thesis is an example of practical applications of image segmentation. An important part is the SYNTHIA database of images, where its properties are shown. At the end of the thesis, the terms and requirements for hardware performance and software needed for good network performance are described in more detail. The Keras framework has already been used, which already includes functions for working with neural networks.

## **KEYWORDS**

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

# 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., konzultanta Ing. Karla Horáka, Ph.D. 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.....	6
1 Úvod.....	10
2 počítačové vidění.....	11
2.1 Snímání obrazu.....	11
2.2 Předzpracování obrazu .....	12
2.2.1 Bodové jasové transformace .....	12
2.2.2 Geometrická transformace.....	12
2.3 Lokální předzpracování .....	13
2.3.1 Filtrace obrazu .....	13
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 .....	33
5.3	PSPNet.....	35
5.4	RefineNet .....	35
5.5	Dilatační konvoluce.....	36
5.6	U-Net.....	36
6	Databáze obrazů.....	38
6.1	Vlastnosti .....	38
6.2	Popis databáze .....	38
7	Implementace sítě .....	41
7.1	Výběr grafické jednotky .....	41
7.2	Požadavky sítě .....	42
7.3	Popis programu pro klasifikaci obrazu .....	43
7.3.1	Rozdělení do vrstev .....	43
7.3.2	Rozdělení kódu na funkční celky .....	44
7.3.3	Compilace modelu .....	46
7.3.4	Proces učení .....	46
7.4	Struktura složek .....	46
7.5	Hlavní rozdíly mezi sítěmi.....	47
8	Závěr .....	48
	Literatura .....	49

# Seznam symbolů, veličin a zkratk

FEKT -	Fakulta elektrotechniky a komunikačních technologií
VUT -	Vysoké učení technické v Brně
NS	Neuronová síť
CNN	Convolutional neural network
CONV	Konvoluční vrstva
POOL	Sdružování
GPU	Grafická karta
CPU	Procesor
RAM	Random Access Memory
CV	Computer vision

## Seznam obrázků

Obr. 1 Řetězec zpracování obrazu.....	11
Obr. 2 Transformace jasové stupnice.....	12
Obr. 3 Geometrická transformace.....	13
Obr. 4 Příklad filtru pro průměrování.....	13
Obr. 5 Detekce hran.....	15
Obr. 6 Histogram a segmentace pomocí jednoho prahu.....	17
Obr. 7 Metoda spojování oblastí.....	18
Obr. 8 Spojování a štěpení.....	18
Obr. 9 Uspořádání neuronů do vrstev.....	20
Obr. 10 Matematický model neuronu.....	21
Obr. 11 Přenosové funkce.....	22
Obr. 12 Schéma architektury CNN.....	25
Obr. 13 Konvoluce.....	26
Obr. 14 Princip konvoluční vrstvy.....	27
Obr. 15 Sigmoida.....	28
Obr. 16 Tanh(x).....	28
Obr. 17 ReLU.....	28
Obr. 18 Leaky ReLU.....	28
Obr. 19 Zmenšování obrazu.....	29
Obr. 20 Možnosti sdružování.....	29
Obr. 21 Fotografie mobilního robota, pohled od stropu.....	31
Obr. 22 Segmentace satelitního snímku.....	32
Obr. 23 Architektura FCN – plně propojené vrstvy.....	34
Obr. 24 SegNet - rozšiřování .....	34
Obr. 25 Architektura sítě SegNet.....	34
Obr. 26 Architektura sítě PSPNet.....	35
Obr. 27 Ukázka architektury sítě RefineNet.....	35
Obr. 28 Dilatační sdružovací vrstva.....	36
Obr. 29 Architektura sítě U-Net.....	37
Obr. 30 RGB obraz Synthia.....	39
Obr. 31 GT/color obraz Synthia.....	39
Obr. 32 GT/labels rozdělení objektů do tříd .....	40
Obr. 33 Depth – zobrazení hloubky obrazu.....	40



## Seznam tabulek

Tabulka 5.1 Porovnání GPU.....	41
Tabulka 5.2 Parametry použitého Hardwaru.....	42

# 1 ÚVOD

Stále přetrvává trend, snahy ulehčovat si život. S čímž souvisí rychlost veškerého dění ať už ve vědě nebo i v způsobu normálního života. Proto se snažíme vyvíjet nové technologie, které nám pomohou. 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 více a více využívat. Z každé strany nás obklopuje velké množství dat, které musíme nějak zpracovat.

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. Jde o další projekt z dnes tak moderní oblasti samořízených aut na jejímž vývoji pracují také ve velkých firmách jako je Google nebo Tesla.

Druhá kapitola práce je zaměřená na počítačové vidění. Postupně bude popsán celý řetězec zpracování obrazu od nasnímání kamerou, přes předzpracování, segmentaci až po popis a klasifikaci obrazu. Největší důraz je kladen na část o segmentaci obrazu a její různé varianty. V části o klasifikaci bude popsáno, jak funguje neuronová síť. Její základní principy, architektura sítě, matematický popis neuronu a metody učení.

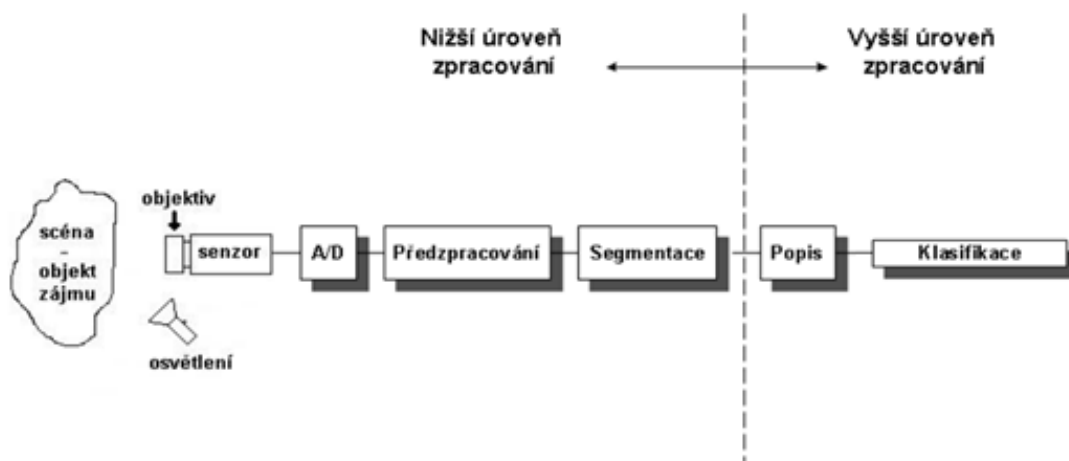
Při práci s obrazovými daty se využívá konvoluční neuronová síť. V kapitole čtyři je popis architektury sítě a na to navazují k jednotlivým vrstvám. Jsou to konvoluční vrstva, přenosová funkce a sdružování. Princip učení sítě, jehož základem je nastavování vah neuronu. Učit však celou síť od nuly je velmi pracné a časově náročné proto se využívají již před-učené sítě, které se upraví (doučí) na našich datech.

Velmi důležitou roli hraje databáze obrázků. Podle toho jak velkou a kvalitní máme databázi nám do značné míry ovlivní i výsledné naučení sítě. Ve své práci používám databázi obrázků, kterou mi dodala firma Artin.

Sedmá kapitola je přímo zaměřená na implementaci sítě. Je zde popsán celý postup mé práce, jenž začal výběrem hardwaru. Následovala instalace potřebného softwaru a popis zdrojového kódu, jenž stojí za práci celé konvoluční neuronové sítě.

## 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 [17].



Obr. 1 Řetězec zpracování obrazu [22]

### 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 žárovka nebo žárovka, opačným příkladem je LED osvětlení. Někdy se používá pouze sluneční záření, avšak je zde riziko velkého rozestupu intenzity.

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

Hlavní funkcí senzoru je převedení světelného signálu na elektrické, pro tuto funkci se využívá fotoelektrický jev v polovodičích. Senzory se zde dělí opět do několika kategorií.

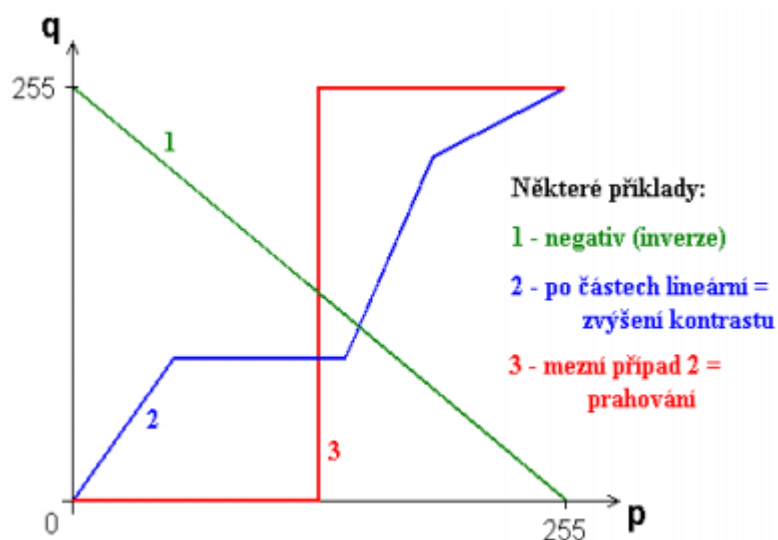
Převodník z analogových hodnot na digitální, v sobě zahrnuje proces vzorkování a kvantování. Většinou již součástí kamery [22].

## 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 tedy potlačit šum, odstranit zkreslení a potlačit či zvýraznit rysy obrazu. Popíšu čtyři základní metody pro předzpracování obrazu. Využívá se nadbytečnosti údajů v obraze (sousední obrazové body mají většinou téměř stejnou hodnotu) [22].

### 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 [22]. Příkladem operací (viz obr. 2), které se zde provádějí je: inverze, ekvalizace histogramu, kontrast, prahování.

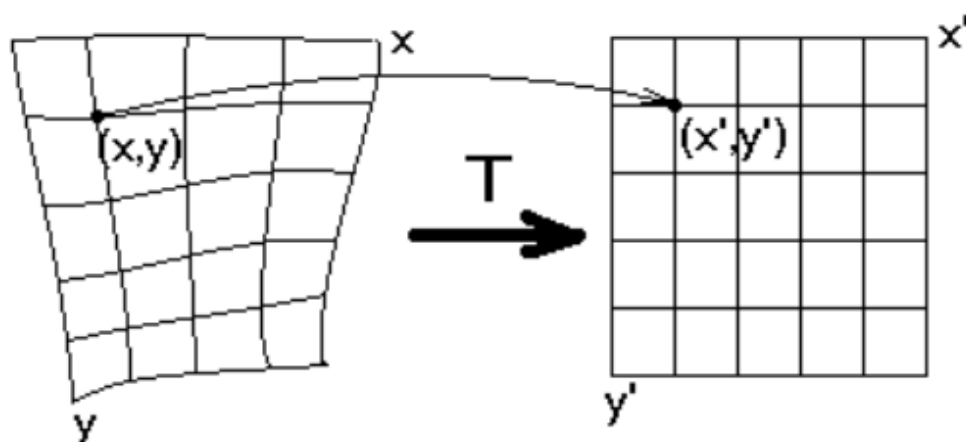


Obr. 2 Transformace jasové stupnice [22]

### 2.2.2 Geometrická transformace

Tato transformace se používá pro odstranění geometrického zkreslení, které vzniklo při špatné úhlu objektu vůči snímané ploše nebo širokoúhlým 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 [22].



Obr. 3 Geometrická transformace [22]

## 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 nový jas 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 [22]

Další možnosti, 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 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 jasů. Hrana je určena velikostí gradientu a směrem.

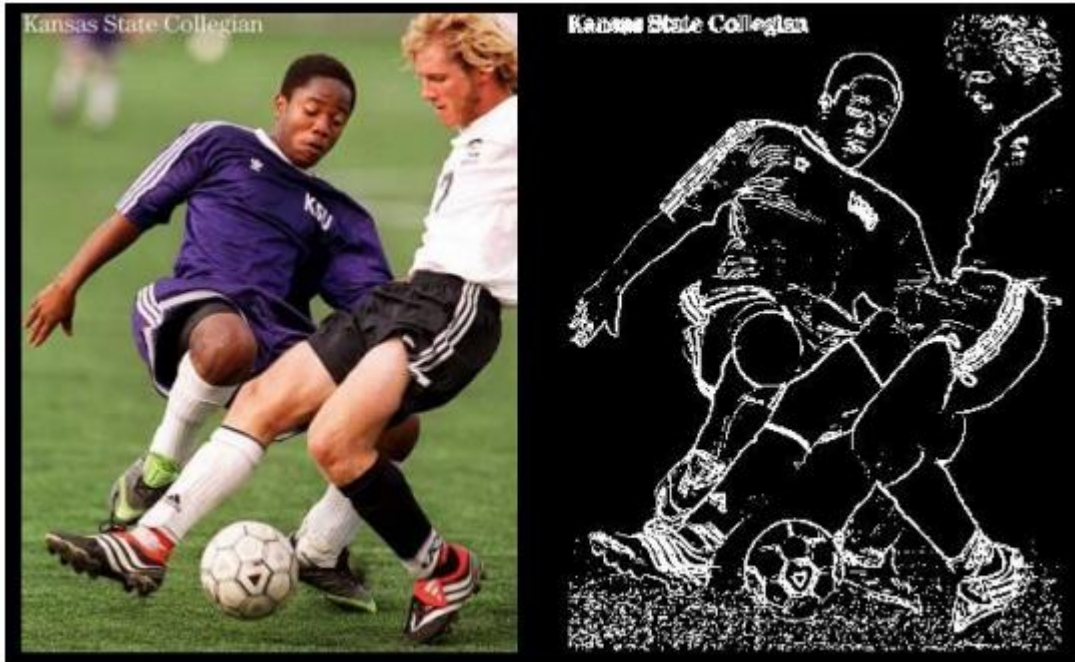
Hranice oblastí obrazu jsou tvořeny hranami, které můžeme najít použitím některého hranového operátoru:

- Robertsův operátor
  - využívá okolí 2x2
  - nevýhoda velká citlivost na šum, z důvodu použití malého okolí
- operátor Prewittové
  - hodnota gradientu odhadována z okolí 3x3
  - 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 tiskářském průmyslu pro zvýšení kontrastu obrazů [31].



Obr. 5 Detekce hran [26]

## 2.4 Segmentace obrazu

Detekce objektů ve snímku je velmi důležitým prvkem každé operace zahrnující práci s obrazem. Uplatňuje se v různých oblastech, od zpracování lékařských snímků, přes zabezpečení (př. kontrola, zda je vše na svém místě) až po využití u fotoaparátu (detekce obličeje nebo úsměvu) [36] [31].

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 [17]

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

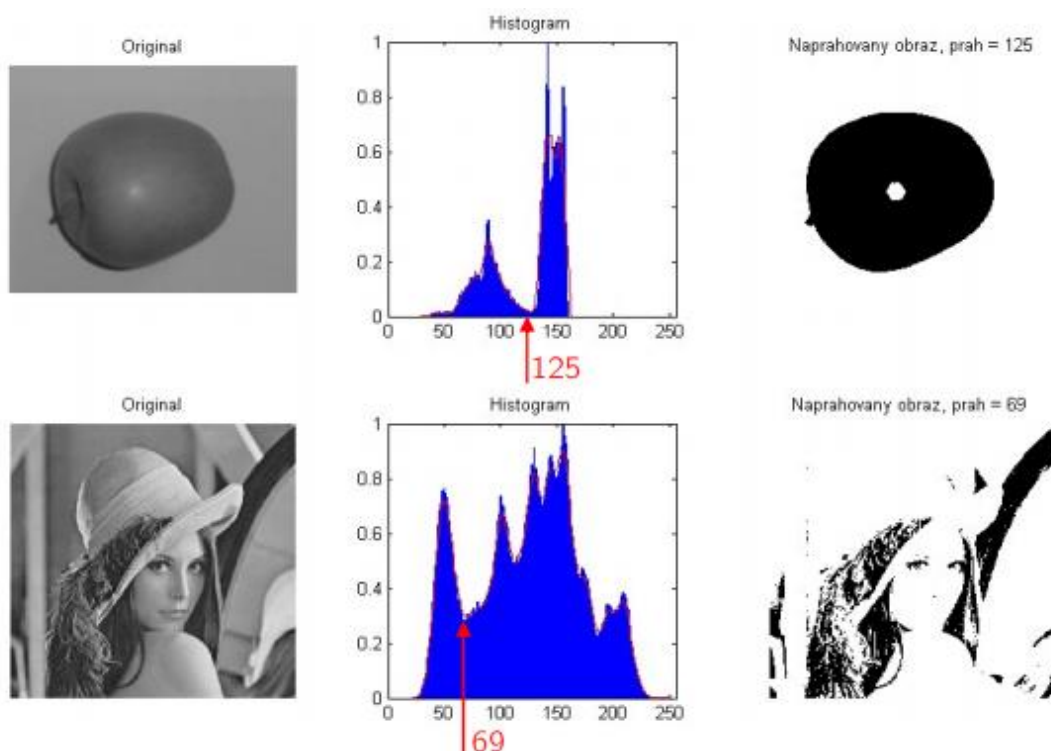
Mezi nejstarší a zároveň výpočetně nejjednodušší metody segmentace se řadí prahování (thresholding). Využívá se však většinou pouze u jednoduchých případů (předměty na bílém pozadí). Objekty jsou charakterizovány konstantní odrazivostí či pohltivostí svého povrchu. Využijeme hodnoty jasu k oddělení objektů 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. 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 bodů. 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 [3].



Obr. 6 Histogram a segmentace pomocí jednoho prahu [26]

## 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].

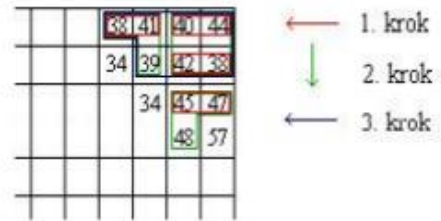
Mezi regionální metody tedy patří:

- 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 [26]. Celý proces je názorně zobrazen na obr. 7.

Příklad:

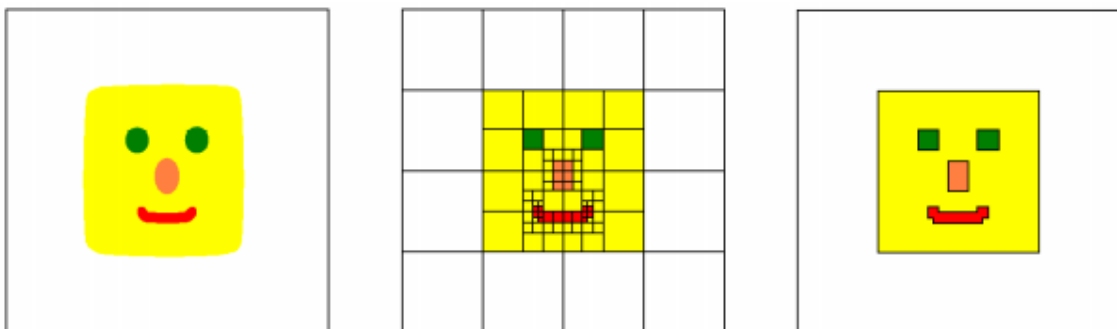
Pokud rozdíl jasů dvou sousedních oblastí je menší než 5, pak tyto oblasti spoj.



Obr. 7 Metoda spojování oblastí [26]

**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 [26]

**Watershed (vodní předěl)** – obraz je chápán jako topografický reliéf, kde jas vstupního obratu 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 se pokouší rozdělit obraz do sémanticky významných částí a klasifikovat každou část do jedné z předem určených kategorie (silnice, člověk, auto atd.) [19]. 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 nezávisle na změně jasů, 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 [16].

## 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. Popíšu Vám nyní jeden z základních klasifikačních algoritmu, a to umělou neuronovou síť.

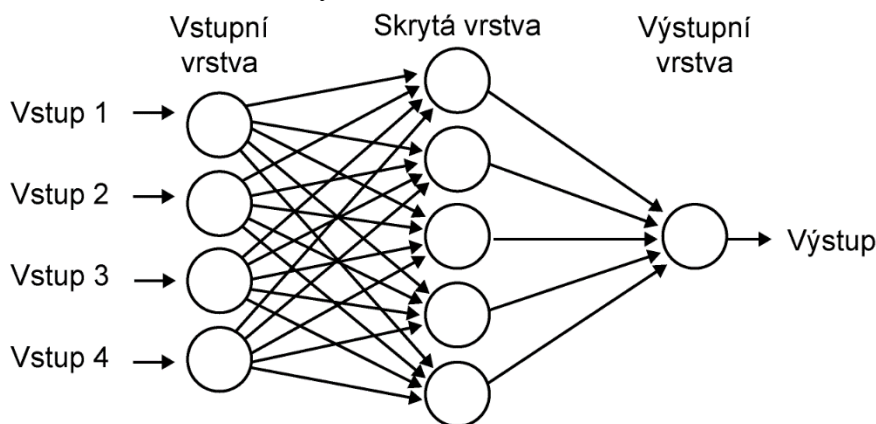
### 3 UMĚLÁ NEURONOVÁ SÍŤ

Umělá neuronová síť je paralelní distribuovaný systém výkonných prvků modelujících biologické neurony, které jsou účelně uspořádány tak, aby systém byl schopen požadovaného zpracování informací [21]. 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 [33].

#### 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 [23]

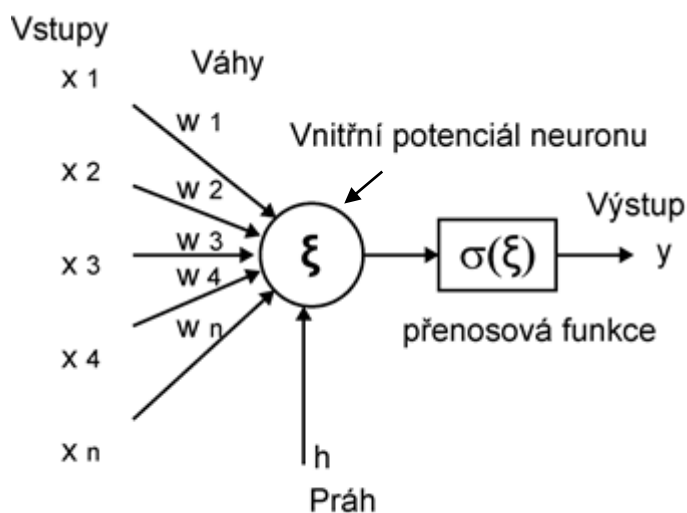
Každá NS má dvě fáze. První fází je vždy učící se, kdy síť na základě tréninkové množiny učí své parametry. Následně přechází do aktivní fáze, kdy na základě předložených vstupu, produkuje výstupy a parametry sítě se již nemění [23].

## 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:

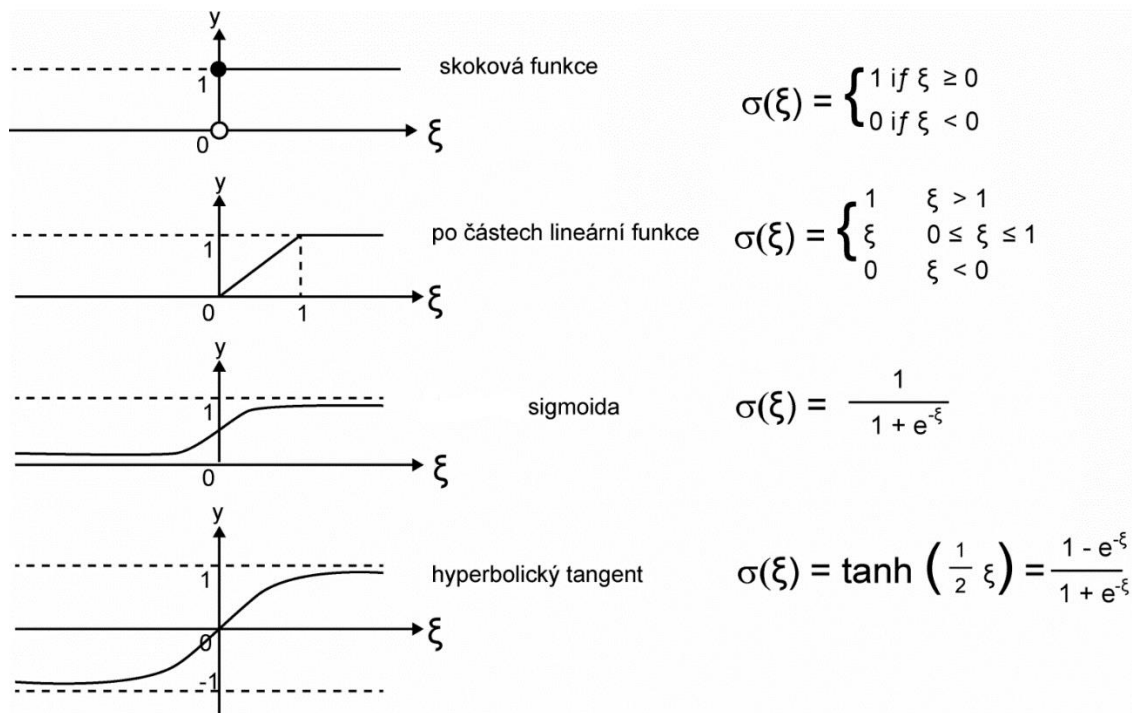
$$\zeta = \sum_{i=1}^n w_i x_i - h \quad (3.1)$$



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 [25].



Obr. 11 Přenosové funkce. [25]

### 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 těžší 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 [4].

### 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 [32].

### 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 70 %-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 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 množin [33].

### 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 [12].

### 4.1 Struktura konvoluční neuronové sítě

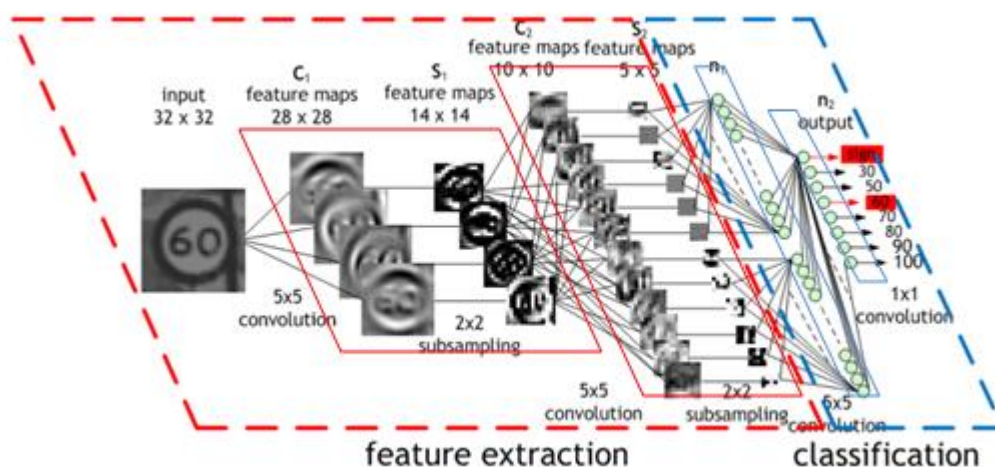
Základním rozdělením CNN je na extrakci příznaků, tedy konvoluce, a klasifikace pomocí NS. V kapitole číslo 3. byl popsán postup, kdy NS dostane na vstupu vektor hodnot, které projdou přes sérii skrytých vrstev, které byly vždy plně propojeny s předcházející vrstvou. Poslední vrstva se nazývá výstupní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

Jako vstupní data se používají obrázky s malým rozlišením. A to například jen 32x32x3 (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 [6].

### 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:

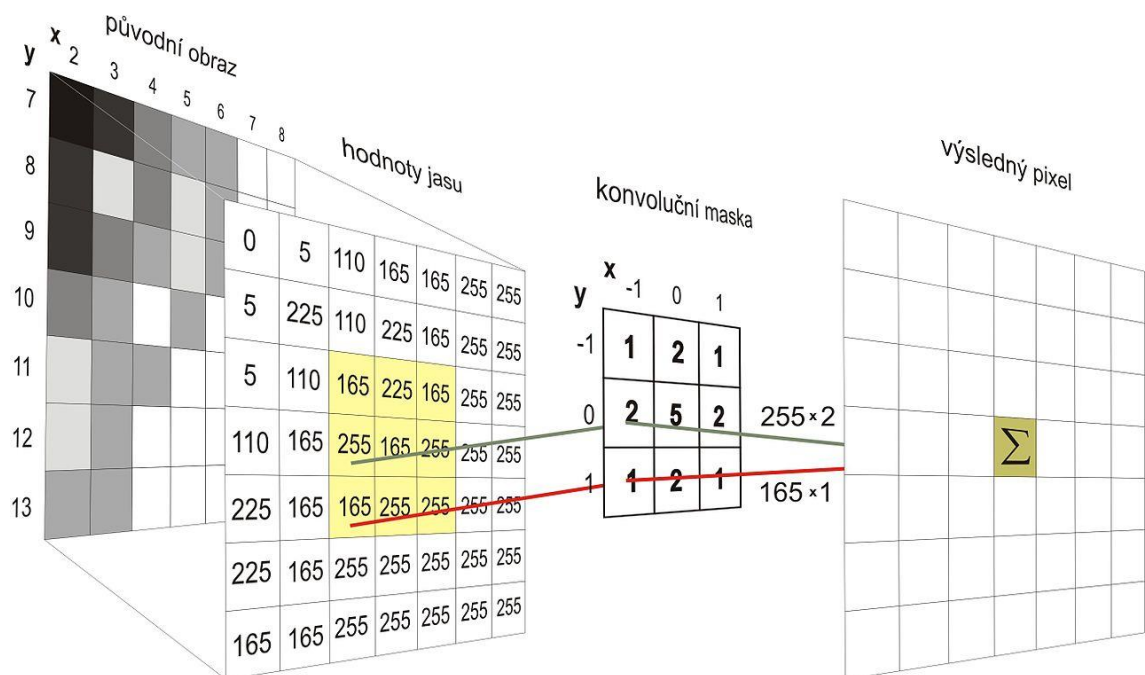
- konvoluční vrstvu,
- aktivační funkci,
- sdružení/pod vzorkování.



Obr. 12 Schéma architektury CNN [9]

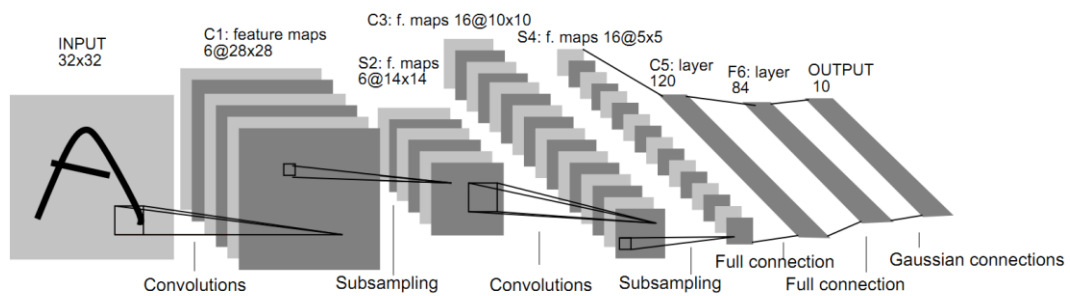
## 4.2.1 Konvoluční vrstva

Konvoluční vrstva (CONV) aplikuje filtr/masku na vstupní obraz. Velikost masky si můžeme zvolit, např. 5x5x3. 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 velikost, dojde ke zmenšení obrázku [9] [10].



Obr. 13 Konvoluce

Ve většině případu se však obrázek prochází více filtry najednou. Máme tedy například 6 filtrů a každý z nich vytvoří dvou rozměrnou 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 použitím sružovací vrstvy [6][7].



Obr. 14 Princip konvoluční vrstvy [10]

Volba velikosti masky také souvisí s množstvím parametrů, které bude daná funkce muset mít. Jako příklad můžeme uvést filtr 5x5x3 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 [24].

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

Kde:

N... velikost obrázku

F ... velikost masky

Po dosazení:  $(32-5) / 1 + 1 = 28$  (3.2)

Výsledný obraz bude velikosti 28x28x6

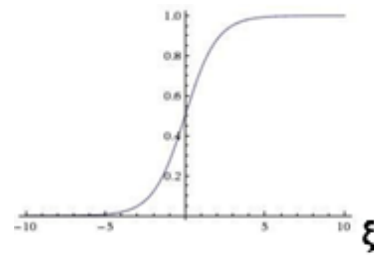
V případech, kdy volíme větší krok než jedna a nevycházelo by nám tak obsáhnout celý obraz, aniž bychom nevynechali žádný obrazový bod, používá se funkce zero-padding. Tato funkce dokáže zvětšit obraz o jeden bod. Na hrany obrazu vloží nulové pixely, což se projeví jako nepatrný černý rámeček.

Pokud máme více konvolučních vrstev za sebou, což se v praxi používá můžeme již hovořit jako o hlubokém učení (deep learning) [20].

## 4.2.2 Přenosová funkce

V kapitole 3.2 již bylo popsáno, jak funguje matematický model neuronu. Nyní se dostáváme k jeho přenosové funkci. Základní jsou tyto čtyři funkce:

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

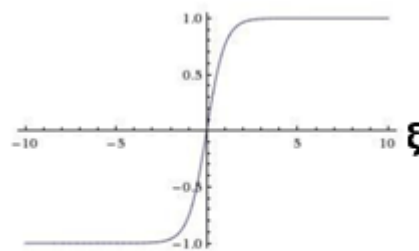


**Sigmoid**

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

Obr. 15 Sigmoida [14]

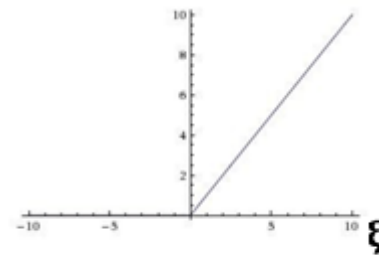
- hodnoty v rozsahu [-1,1]
- souměrný kolem nuly
- stále nevýhoda saturace



**tanh(ξ)**

Obr. 16 Tanh(ξ) [14]

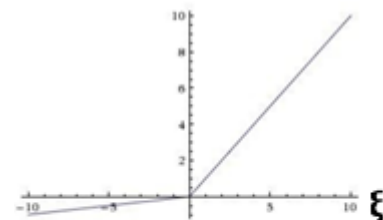
- není zde problém se saturací
- výpočetně efektivní
- konverguje až 6x rychleji než sigmoida nebo tanh
- výstup není centrován kolem nuly
- může však během tréninku „zemřít“



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

Obr. 17 ReLU [14]

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



**Leaky ReLU**

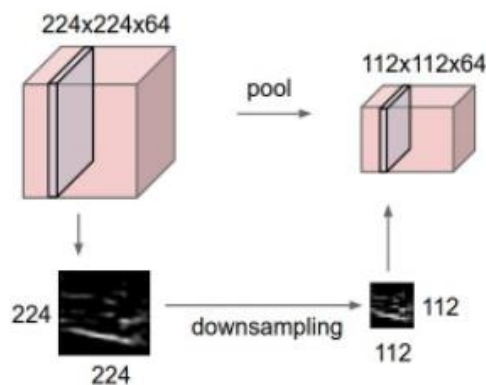
$$f(\xi) = \max(0.01\xi, \xi)$$

Obr. 18 Leaky ReLU [14]

Mezi další funkce ještě patří Maxout a ELU. Doporučuje se používat ReLU, a rozhodně při učení CNN nepoužívat sigmoida [14].

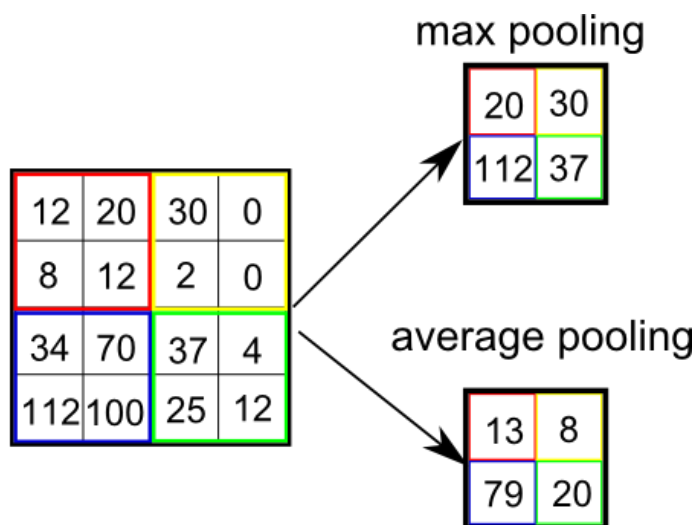
### 4.2.3 Sdružení

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 [5]. Ukázka zmenšení velikosti obrazu je na obr. 19.



Obr. 19 Zmenšování obrazu [6]

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 [20].



Obr. 20 Možnosti sdružování [14]

### 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 [10]. Dalším krokem je nastavení vah na vstupech neuronů, bylo již vysvětleno v kapitole 3.3.

### 4.4 Princip vybavování

Vybavování neboli aktivní režim sítě, využíván k vlastnímu použití naučené sítě na daný výstup. Může se jednat o jednorázový proces nebo iterační [21].

### 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 [6].

**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. A to 3x3 a pro sdružování masku 2x2. Jejich model je k použití v knihovně Caffe.

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

## 4.6 Praktické aplikace CNN pro segmentaci obrazu

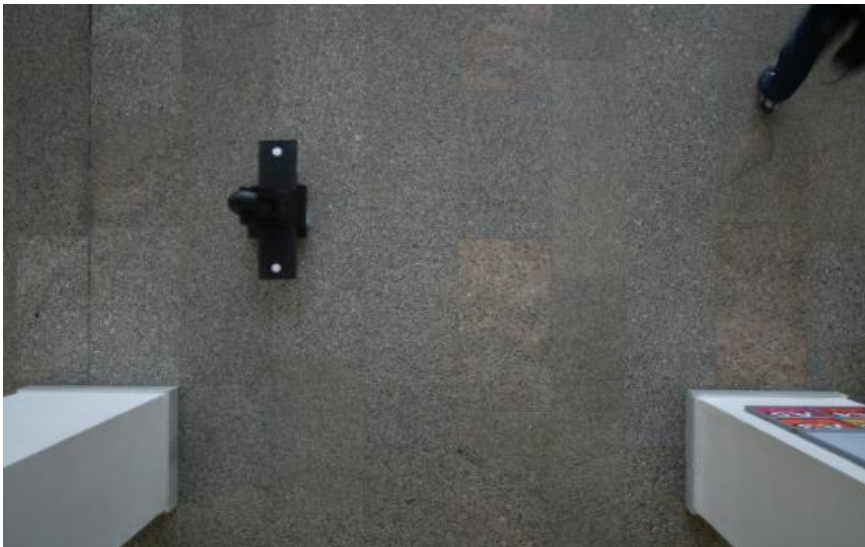
V praxi se CNN používají v mnoha různých odvětvích techniky, od systému strojového vidění přes vyhledávací, diagnostické a asistenční systémy až po autonomní vozidla. Uvedu čtyři příklady, kde je využití těchto sítí velmi přínosné a posunulo vývoj v některých oblastech kupředu.

### 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 [13].

### 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, byly tedy na robota 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]

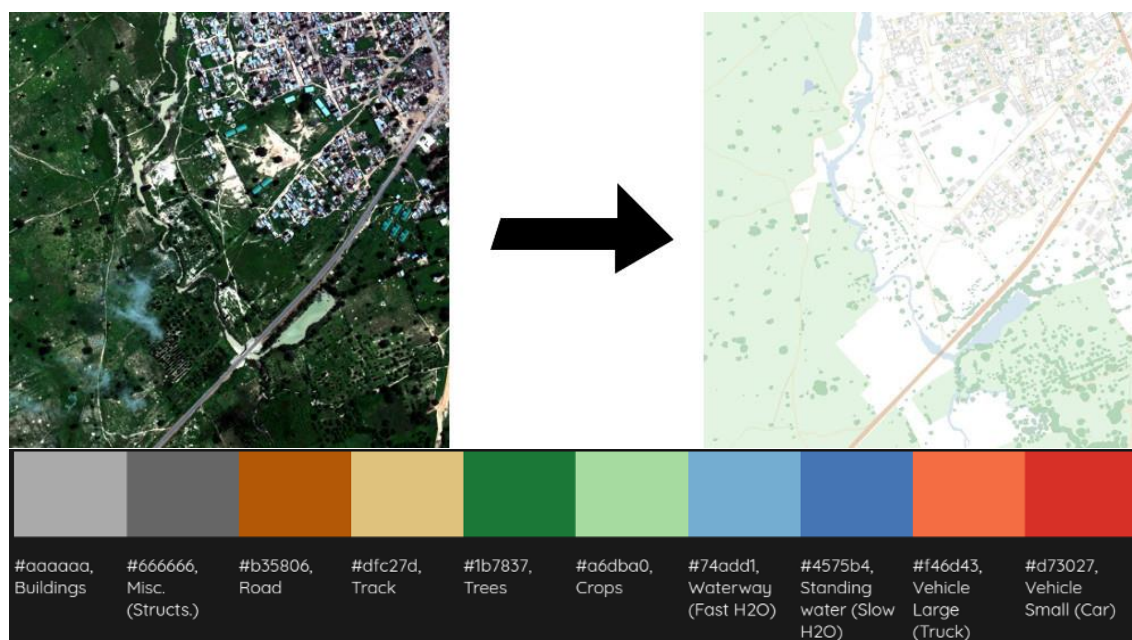
### 4.6.3 Analýza zdravotnických obrazů

Konvoluční neuronové sítě v posledních několika letech přetvořili lékařskou analýzu obrazu. Architektura U-Net je jednou z neznámějších 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-organová segmentace CT obrazů břicha a segmentace mozkových nádorů na MR obraze [17].

### 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 [18].



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 sémantická segmentace / 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 sítě (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 sítě 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 sémantická segmentace 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 [19] [28].

### 5.1 FCN

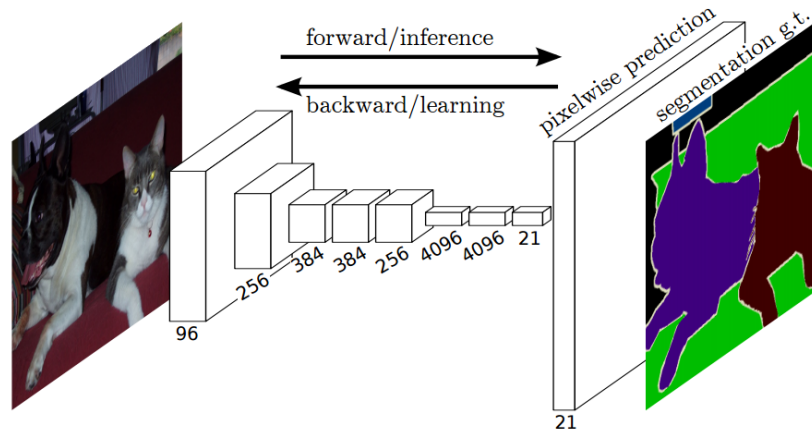
Plně propojené vrstvy (fully connected layers - FCN) v klasifikačních sítích lze považovat za konvoluční s jádry, které pokrývají celé vstupní oblasti. To odpovídá vyhodnocení původní klasifikační sítě na překrývajících se vstupních oblastech, ale je to mnohem efektivnější, protože výpočet je sdílen 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 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 [28].

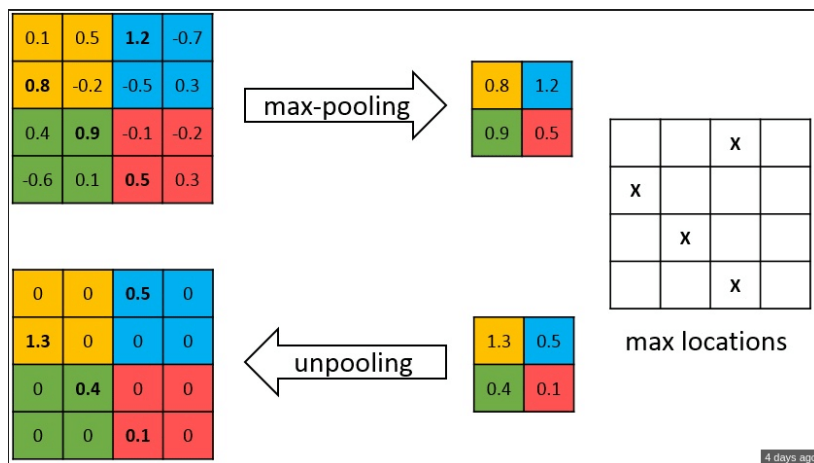
### 5.2 SegNet

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

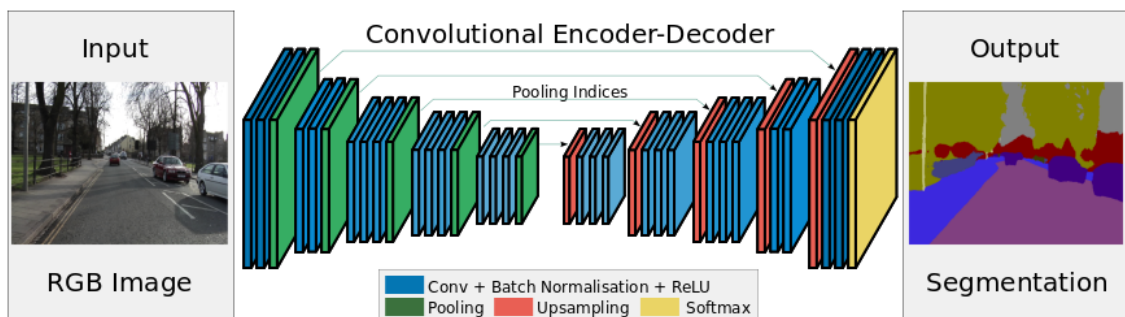
pro provádění nelineárního vzorkování, obr. 24. Eliminuje se tím potřeba učení se převzorkování. Vzorkované mapy jsou řídké, poté však projdou přes konvoluční vrstvu s tréninkovými filtry, které vytvářejí husté mapové vrstvy. Celý princip vidíme na obr. 25. Avšak tato architektura dosahuje horších výsledků než FCN nebo PSPNet [19].



Obr. 23 Architektura FCN – plně propojené vrstvy [28]



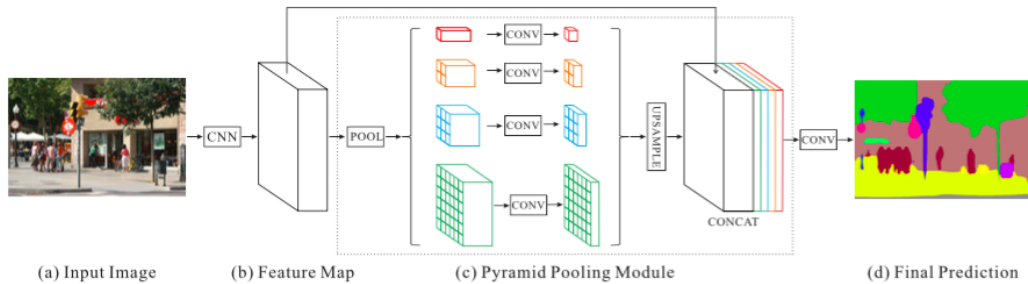
Obr. 24 SegNet - rozšiřování [28]



Obr. 25 Architektura sítě SegNet [19]

## 5.3 PSPNet

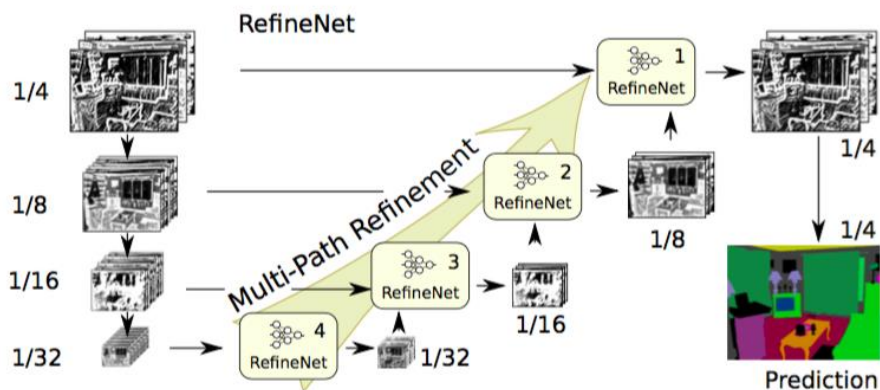
PSPNet modifikuje základní architekturu ResNet přidáním dilatační konvoluce a funkcí po inicializační sružovací vrstvou. Síť analyzované pyramidové scény (pyramid scene parsing network - PSPNet) využívá schopnost globálního spojování informací z různě velkých regionů (viz obr. 26). Proto se to nazývá pyramidová scéna jdeme od velkých části obrazu k menší, kde se již jen doladují detaily [28].



Obr. 26 Architektura sítě PSPNet [28]

## 5.4 RefineNet

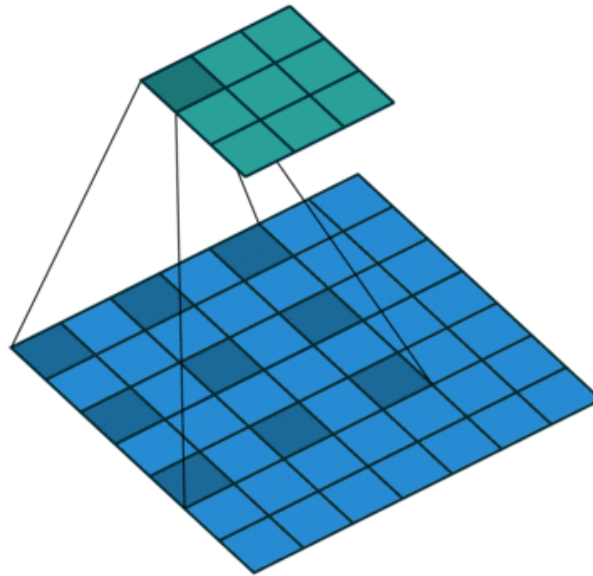
Obecná zdokonalovací síť s více cestami, to je RefiNet. Využívá všechny vstupy při více rozlišeních v průběhu procesu odběru vzorků, spojí extrahované funkce a přenese je do další fáze, aby umožnila předpověď s vysokým rozlišením. Tímto způsobem mohou hlubší vrstvy, které zachycují sémantické prvky vyšší úrovně, být přímo použity k vyhlazení jemně zrnitého obrazu z předchozích konvolucí (obr. 27). Zavádí řetězové sružení, které dokáže zachytit kontext pozadí z větší oblasti obrazu. Jednotlivé části sítě 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 [19].



Obr. 27 Ukázka architektury sítě RefineNet [19]

## 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.



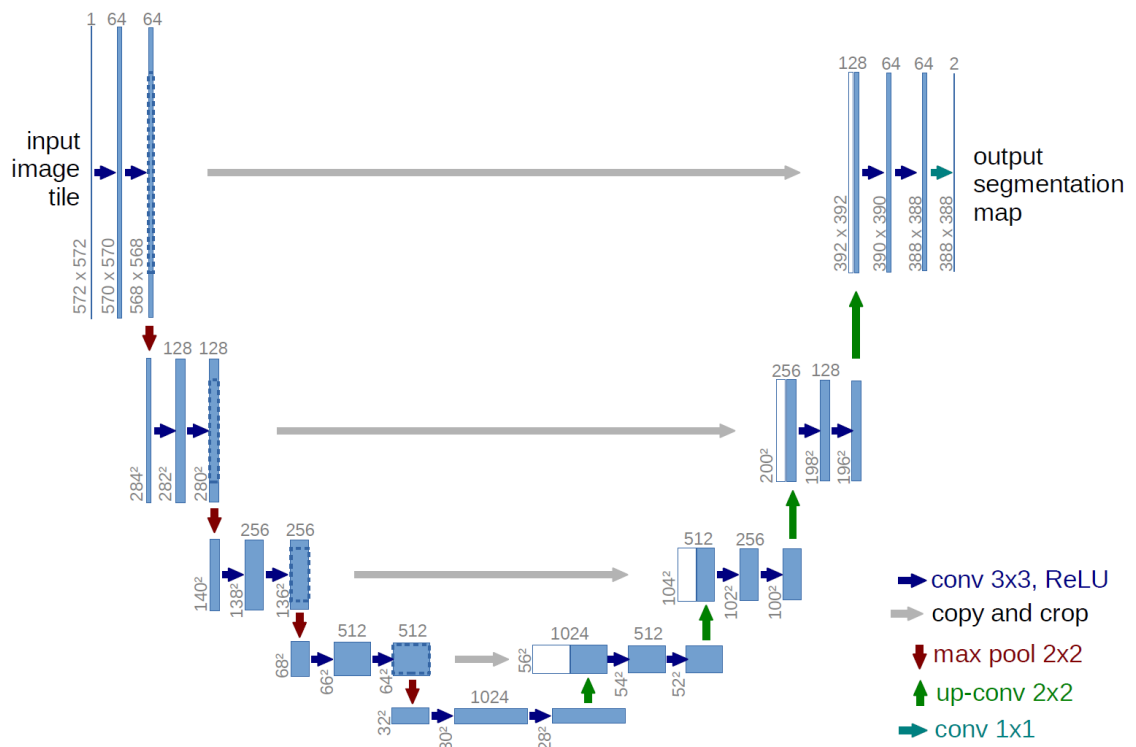
Obr. 28 Dilatační sdružovací vrstva [28]

Poslední dvě sdružovací vrstvy jsou odstraněny a nahrazeny dilatační konvoluční vrstvou. Konvoluce mezi 3 a 4 vrstvou má dilataci dva, poté již je velikost dilatační mezery čtyři. Tímto způsobem získáme husté mapy předpovědí bez zvýšení počtu parametrů [28].

## 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 sdružovací vrstvě v kodéru [28].



Obr. 29 Architektura sítě U-Net [28]

## 6 DATABÁZE OBRAZŮ

V rámci bakalářské práce pracuji s databází obrazu SYNTHIA. Jedná se o syntetickou sbírku snímků s anotací. Byla vytvořena za účelem učení sémantické segmentace a pro pochopení scény v kontextu scénářů řízení. Skládá se ze sbírky fotorealistických snímků vykreslených z virtuálního města. Navíc přichází s přesnými sémantickými anotacemi na úrovni pixelů [30]. Podrobně jsem prozkoumal vlastnosti této databáze a zjistil jaké možnosti nabízí.

### 6.1 Vlastnosti

Představení vlastností databáze obrazů a co vše nám může nabídnout. Je k dispozici velká variace možností ať už rozdílné prostředí změnou denní doby nebo jiným počasím [30].

- velké množství dat: více než 200 000 obrázků v rozlišení HD z video streamu a na 20 000 snímků z nezávislých scén,
- rozmanitost scén: Evropská města, dálnice, zeleň,
- mnoho druhů dynamických objektů: auta, chodci, cyklisté,
- více témat pro všechna roční období,
- podmínky osvětlení a počasí: dynamické stíny, několik denních režimů (východ slunce, poledne, západ slunce), noční režim, déšť, sníh, mlha,
- snímače: 8 kamer RGB, obraz z celých 360°, 8 snímačů hloubky.

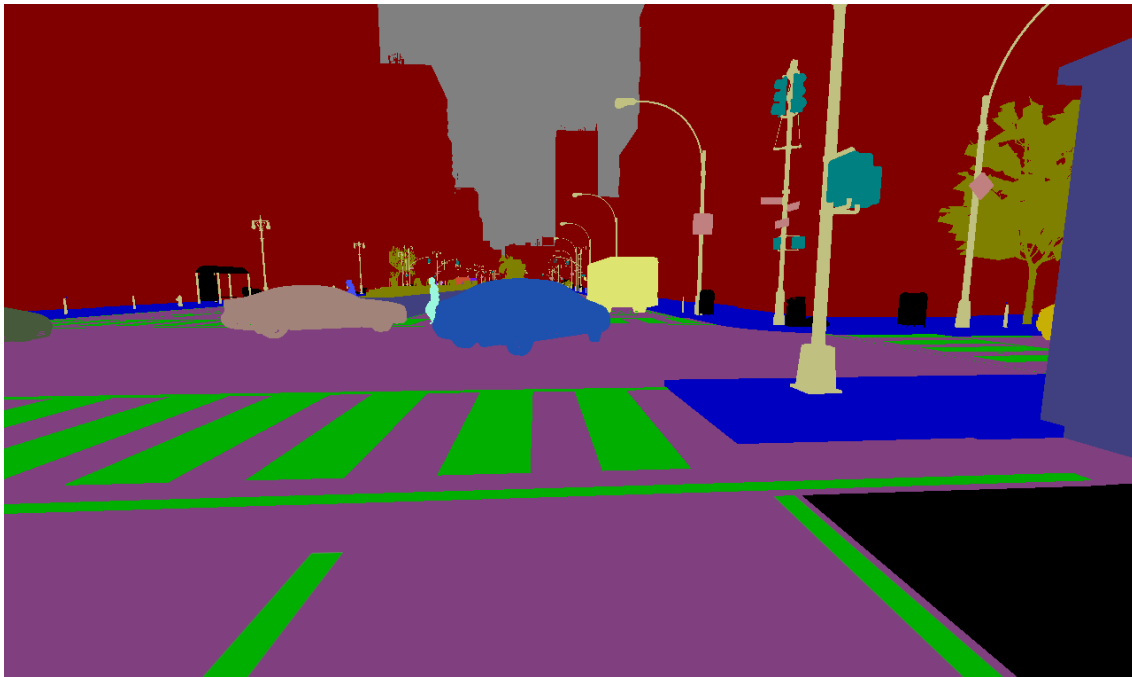
### 6.2 Popis databáze

Můžeme si vybrat jednotlivé typy scén z velké nabídky možností, Scéna, kterou teď uvidíte je ze západu slunce (slunečný den, poloha město), konkrétně se jedná o SYNTHIA-SEQS-05-SUNSET. Při otevření databáze objevíte 4 podsložky:

- a) RGB – obsahuje standartní barevné obrázky 1280x760 používané pro trénování, názornou ukázkou vidíme na obr. 30,
- b) depth – obsahuje obrázky v HD rozlišení, kde hloubka je zakódovaná do 3 kanálu v centimetrech, viz obr. 33,
- c) GT/color – obsahuje png soubory (jeden pro každý obraz), anotace je dána pomocí barevné reprezentace a slouží především pro vizualizace, nikoliv pro učení, viz obr. 31,
- d) GT/labels – obsahuje opět png soubory, ovšem zde jsou dány pomocí 2 kanálů, první obsahuje třídu pixelů, ve druhém je dáno unikátní ID pro dynamické objekty, viz obr 32,
- e) CameraParams – složka obsahuje textový soubor s globální pozicí kamery a dalšími parametry pro zpětnou 3D rekonstrukci.



Obr. 30 RGB obraz Synthia [30]



Obr. 31 GT/color obraz Synthia [30]

Class	R	G	B	ID
Void	0	0	0	0
Sky	128	128	128	1
Building	128	0	0	2
Road	128	64	128	3
Sidewalk	0	0	192	4
Fence	64	64	128	5
Vegetation	128	128	0	6
Pole	192	192	128	7
Car	64	0	128	8
Traffic Sign	192	128	128	9
Pedestrian	64	64	0	10
Bicycle	0	128	192	11
Lanemarking	0	172	0	12
Reserved	-	-	-	13
Reserved	-	-	-	14
Traffic Light	0	128	128	15

Obr. 32 GT/labels rozdělení objektů do tříd [30]

Ve složkách s obrazy jsou další podsložky, kde první vrstva rozděluje obrazy na stereo\_left/right. V další vrstvě je ještě rozdělení na 4 pohledy: vpřed, vzad, vlevo, vpravo. To odpovídá 8 kamerám, o kterých jsem se již zmínil.



Obr. 33 Depth – zobrazení hloubky obrazu [30]



## 7 IMPLEMENTACE SÍTĚ

Mým úkolem v této části bylo znovu zprovoznit a pochopit, jak celá již navrhnutá síť funguje. Vycházel jsem z diplomové práce Ing. Ondřeje Zapletala [35]. Měl jsem k dispozici text jeho práce a zdrojový kód.

### 7.1 Výběr grafické jednotky

V kapitole 4 již bylo popsáno, že konvoluční neuronové sítě mají velké požadavky na výkon výpočetní techniky. A to hlavně na grafickou jednotku (GPU), ale samozřejmě nesmíme opomenout procesor nebo paměť RAM. V rámci výběru GPU jsem zpracoval tabulku, která porovnává dvě velmi výkonné karty.

	<b>Tesla C2075</b>	<b>GeForce GTX 1080</b>
CUDA cores	448	2560
Core clock	1150 MHz	1721 MHz
Bus Type	PCI-E 2.0 x16	PCI-E 3.0 x16
Memory size	6 GB	8 GB
Memory Clock	1500 MHz	10010 MHz
Memory Bus	384 bit	256 bit
ROPs	48	64
Memory Type	GDDR5	GDDR5X
Process Technology	40 nm	16 nm
<i>Pixel fill rate</i>	27.6 gigapixels/s	110.91 gigapixels/s
<i>Texture fill rate</i>	32.2 gigatexels/s	277.28 gigatexels/s
<i>Texture units</i>	56	160
<i>Single Precision performance</i>	1030.4 GFLOPS	8872.96 GFLOPS
<i>Double Precision performance</i>	515.2 GFLOPS	277.28 GFLOPS
<i>Memory bandwidth</i>	144 GB/s	320.32 GB/s
Outup ports	DVI	Dual-link DVI-D
		HDMI-2.0b
		Display Port-1.4 *3

Tabulka 1 Porovnání GPU

V tabulce 1 je porovnání dvou karet: GPU Tesla C2075 a GeForce GTX 1080. U většiny parametrů byl lepší karta GeForce, avšak u dvou parametrů byla lepší karta

Tesla. Když jsem následně studoval, jak fungují výpočty a co vše se v GPU děje během učení sítě, zjistil jsem, že právě parametr Double precision performance, je jedním z nejdůležitějších parametrů pro urychlení výpočtu. Což vlastně znamená, že procesor grafické karty zvládne pracovat lépe s 64 bitovými daty. A jelikož většina obrázku je 64 bitová, je to pro nás velmi prospěšné. Když ještě přihlédnu k ceně, karta Tesla je asi 4x dražší ale jelikož byla tato karta ve škole k dispozici, nebylo již se třeba dále rozhodovat a zvolil jsem GPU Tesla C2075. Všechny hlavní parametry použitých komponentů počítače jsou uvedeny v tabulce 2.

GPU	Tesla C2075
CPU	Intel Core i5 650, 3,2GHz
RAM	4 GB

Tabulka 2 Parametry použitého Hardwaru

## 7.2 Požadavky sítě

Zdrojový kód na klasifikaci obrazu, se kterým jsem měl pracovat byl napsán panem Zapletalem v programovacím jazyce Python. Aby bylo možno kód spustit je potřeba nainstalovat program, který zvládne přeložit jazyk Python do strojového kódu, aby jej mohl procesor vykonat. Já jsem použil program PyCharm od společnosti JetBrains. Začal jsem instalací softwaru *Anaconda*. Což je distribuce s otevřeným zdrojovým kódem jazyku Python pro rozsáhle zpracování dat, prediktivní analýzu a vědecké výpočty. Jehož cílem je zjednodušení práce [9]. Nyní bylo potřeba otevřít *Anaconda Prompt*, což je verze příkazového řádku, ve kterém jsem zadal toto pro aktualizaci [1]:

```
conda update conda
conda update --all
```

Následně bylo zapotřebí, stáhnout a nainstalovat dva programy pro podporu a správu GPU (aby bylo možno efektivně používat GPU při učení sítě). A to CUDA Toolkit 9.0 a cuDNN. Nyní jsem musel nastavit cestu v uživatelských proměnných ve vlastnostech systému k cuDNN. Poté jsem vytvořil prostředí pro python 3.6 a aktivoval Tensorflow následující příkazem:

```
conda create -n tensorflow python=3.6 numpy scipy matplotlib spyder
activate tensorflow
```

Instalace Tensorflow a Keras

```
Pipinstall--ignore-installed--upgrade
https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-1.0.1-cp35-
cp35m-win_amd64.whl
pip install keras
```

Keras je vlastně rozhraní pro programování vysokorychlostní neuronové sítě [2]. Instalace je hotová, nyní již stačí spustit dané skripty buď přes konzoli v příkazovém řádku nebo za pomoci programu PyCharm.

## 7.3 Popis programu pro klasifikaci obrazu

Pro implementaci CNN byl použit sekvenční model Keras, což je koncept který je vhodný pro modelování neuronových sítí. Celá síť je postavena na vrstvách. Je potřeba vytvořit ekvivalentní topologii, aby bylo možno využít frameworky. Model je vytvořen voláním sekvenčního konstruktora **Sequential()**, jednotlivé vrstvy jsou přidávány funkcí **add** [35].

```
Model = Sequential()
model.add(layer),
```

kde **layer** je definice vrstvy.

### 7.3.1 Rozdělení do vrstev

Všechny vrstvy byly vytvořeny na základě této struktury (vychází z knihovny Keras), která byla již teoreticky vysvětlena v kapitole 4

#### Konvoluční vrstva

Konvoluce má architekturu následující struktury

```
Conv2D(filters=n, kernel_size=(z, z), strides=(s, s), padding='valid',
input_shape=shape),
```

kde **n** je počet filtrů, **z** je velikost jádra filtru, **s** je počet pixelů v kroku a **input\_shape** definuje velikost vstupní matice.

#### Přenosová (aktivační) vrstva

Chcete-li přidat přenosové funkce na výstup vrstvy, nesmíme ještě zapomenout na uživatele, který si může vybrat, jakou funkce použije.

**Activation(activation\_function),**

kde *activation\_function* je 'softmax' nebo 'ReLU'.

Obě specifikace jsou ekvivalent, protože Keras automaticky používá lineární přenosovou funkci pro každou vrstvu.

**Sdružovací**

tato vrstva je specifikovaná jako

**MaxPooling2D(pool\_size=(z, z), strides=(s, s)),**

kde *pool\_size* definuje velikost filtru a *strides* udává velikost kroku

**Plně propojená vrstva**

Poslední plně propojenou vrstvou, která již provádí rozdělení do kategorií.

**Dense(num\_of\_units),**

kde *num\_of\_units* je počet neuronů v poslední plně propojené vrstvě. Dává tak počet možných výsledných klasifikací [35].

## 7.3.2 Rozdělení kódu na funkční celky

V kapitole 7.3.1 byly popsány jednotlivé vrstvy teoreticky, tato kapitola však již rozděluje kód na funkční celky, které se plně shodují s reálným rozdělením kódu v mé aplikaci.

**Application.py**

Hlavní část pro přístup ke knihovně Keras. Čeká v nekonečné smyčce na konfigurační soubor v databázi. Co 60 sekund se dotazuje, zda je nějaký nový model na učení. Když je nalezen nový zahájí učení, voláním funkcí a podprogramů. začne trénovací proces. V závislosti na nastavení aplikace běží učení CNN na GPU.

**Dataset\_prepare.py**

Skript připravuje data na proces učení. V prvním kroku načte a převede obrázky na vhodnou velikost, aby byly všechny obrazy stejné, již přednastavené, velikosti a poté je uloží do více rozměrného pole. Data jsou následně náhodně rozdělena mezi trénovací a testovací množinu v určitém poměru. Jako poslední část se ještě provede převod z n rozměrného pole na formát *float32*.

## **Evaluation\_model\_creator.py**

V tomto skriptu se vytváří konfigurační soubor pro testování modelu namísto učení.

## **Evaluator.py**

Model slouží k vyhodnocování a trénování vytvořených modelů.

## **Extracto.py**

Testování knihovny Keras

## **Loggers.py**

Tímto skriptem jsou vytvářeny protokoly o průběhu trénování, testování, formátu a dalších důležitých událostech, které nastaly během procesu učení. Tyto data jsou uloženy, a lze tak zpětně zjistit, zda vše proběhlo správně a popřípadě, kde nastala chyba

## **Model\_creator.py**

Vytváří modely modelům knihovny Keras. Vytváří model celé CNN, přes který budou procházet všechny vstupní data. Nastavuje vlastně jednotlivé vrstvy a jejich parametry. Kolik bude konvolučních vrstev, jak velká bude maska, zda bude následovat aktivační funkce a zda se použije sduřovací vrstva.

Konkrétní nastavení parametrů modelu:

```
model = Sequential()
layer1 = Conv2D(48, (11, 11), strides=(4), padding='same',
input_shape=shape)

model.add(layer1)
model.add(Activation('relu'))

model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

layer2 = Conv2D(64, (5, 5), padding='same')
model.add(layer2)
model.add(Activation('relu'))

model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

layer4 = Conv2D(95, (3, 3), padding='same')
model.add(layer4)
model.add(Activation('relu'))

layer5 = Conv2D(64, (3, 3), padding='same')
model.add(layer5)
model.add(Activation('relu'))
```

### 7.3.3 Kompilace modelu

Pokud je struktura modelu specifikována ještě před tím, než může být naučena. Potřebujeme funkci, která optimalizuje procedury a nadefinuje matici. Tato metoda je nazvána *compile*.

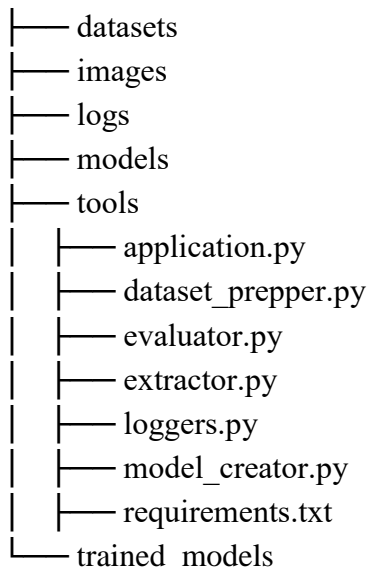
### 7.3.4 Proces učení

Proces učení se v kerasu nazývá modelování (model fitting). Nabízí dva typy metod *fit* a *fit\_generátor*. První metoda načte datovou sadu jen jednou a použije ji k trénování sítě. Avšak tuto metodu nelze použít na všechny typy dat. Při použití druhé možnosti použijeme vlastnost jazyka Python nazvanou generátor. Tento generátor bere data z databáze obrázků v nekonečné smyčce. V závislosti na nastavení může generovat jak trénovací data nebo testovací.

Další velkou výhodou použití *fit\_generator* metody je, že během toho, jak se model učí v GPU, je možno připravovat další várku dat paralelně. Toto je velmi užitečná funkce, protože nedochází k zpomalování procesu učení [35].

## 7.4 Struktura složek

Aby program správně fungoval je potřeba vytvořit si strukturu složek, jak lze vidět níže. Do složky images dáme obrázky a do složky tools vložíme samotný program.



Je potřeba spustit programy v tomto pořadí:

- a. dataset\_prepare.py
- b. model\_creator.py
- c. application.py

## 7.5 Hlavní rozdíly mezi sítěmi

Rozdíly mezi použitím konvoluční neuronové sítě pro klasifikaci a segmentaci obrazu, jsou v některých bodech dost zásadní. Zde je vypsány odlišnosti mezi architekturami sítí.

- a) Problém při použití CNN pro segmentaci jsou sdružovací vrstvy. Mají totiž vlastnost, že odstraní informaci o poloze bodu. Nicméně sémantická segmentace vyžaduje přesné zarovnání třídících map, a proto potřebují informace o poloze zachovat.
- b) U klasifikace je výsledkem hluboká síť (jde pouze o procentuální vyjádření, do které třídy bude objekt zařazen), segmentace však vyžaduje nejen rozlišení na úrovni obrazových bodů (což běžná CNN dokáže nabídnout) ale navíc také mechanismus projektu v rozlišení prvků naučených v různých fázích kodéru na prostor pixelu.
- c) Při sdružování dochází ke zmenšování obrazu, proto je třeba u segmentace používat navíc dekódovací mechanismus, který dokáže zvětšit obraz opět do původní velikosti.
- d) Potřebujeme FCN, tedy plně propojené vrstvy, kdežto klasická CNN nemá již všechny vrstvy spojené.
- e) Pro trénování segmentace potřebujeme jiné databáze obrazu a způsob učení je také odlišný.

## 8 ZÁVĚR

Při výběru mé semestrální práce mne zaujalo téma neuronových sítí a s tím spojená klasifikace obrazů pomocí konvolučních neuronových sítí. Na bakalářské práci jsem spolupracoval s firmou Artin, která mi dodala data na zpracování. Po konzultaci mi firma doporučila přeformulovat zadání z klasifikace obrazu na segmentaci. Jelikož v Artinu právě pracovali na projektu Roboauta a zrovna na zmíněné segmentaci, rozhodl jsem se toto téma přijmout s vidinou uplatnění mé práce v reálném projektu. Jedná se o rozvíjející se oblast autonomního řízení, které mě velmi zaujalo.

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. Popsal jsem celý řetězec zpracování obrazu v počítačovém vidění, kde jsem se zaměřil hlavně na část segmentace. V této části jsem zpracoval základní metody segmentace obrazu a velkou část této kapitoly věnoval sémantické segmentaci, kterou jsem měl použít pro vlastní aplikaci.

Mým zadáním bylo znovu rozchodit diplomovou práci Ondřeje Zapletala, která se věnovala klasifikaci objektů do předem daných tříd pomocí konvoluční neuronové sítě. Při znovu zprovoznování sítě jsem řešit problémy s výběrem správného hardwaru, bylo nutno zjistit jaký je třeba přesně software abych mohl naplno využívat výkon GPU. V textu diplomové práce O. Zapletala nebyla tahle část ohledně softwaru popsána, zjišťoval jsme tedy všechny informace znovu. Práce byla napsána v jazyce Python, s tímto jazykem jsem se doposud nesetkal, a to bylo další ztížení při řešení mé bakalářské práce

Dalším bodem zadání bylo upravit síť O. Zapletala pro segmentaci obrazu. Po určitém čase studování dané problematiky jsem zjistil, že změnit síť z klasifikace obrazu na segmentaci nepůjde tak lehce. Segmentace vyžaduje velmi rozdílnou architekturu, než jakou jsem měl nyní v kódu k dispozici, a nejen tuto rozdílnost by bylo potřeba změnit. Musel bych tedy provést radikální změny sítě, dokonce až takové, že by stálo za úvahu začít psát celý program znovu. Hlavní rozdíly mezi sítěmi jsou vypsány v kapitole 7.5.

Nyní již vím, jaké jsou architektury pro sémantickou segmentaci obrazu. V dalším pokračování by bylo potřeba tyto architektury podrobněji prozkoumat a vybrat tu nejvhodnější a následně ji implementovat, což by mohlo být pro mě následným zadáním diplomové práce.



# Literatura

- [1] *Anaconda Cloud* [online]. 2018 [cit. 2018-05-14]. Dostupné z: <https://anaconda.org/anaconda/python>
- [2] ANTONIOSEHK. *Keras-TensorFlow-GPU-Windows-Installation: 10 easy steps on the installation of TensorFlow-GPU and Keras in Windows* [online]. In: . 16.4.2017 [cit. 2018-01-02]. Dostupné z: <https://github.com/antoniosehk/keras-tensorflow-windows-installation>
- [3] 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>
- [4] 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/>
- [5] *Convolutional Neural Network (CNN) basics* [online]. [cit. 2018-01-02]. Dostupné z: <https://pythonprogramming.net/convolutional-neural-network-cnn-machine-learning-tutorial/>
- [6] *CS231n Convolutional Neural Networks for Visual Recognition* [online]. 2017 [cit. 2017-12-30]. Dostupné z: <http://cs231n.github.io/convolutional-networks/>
- [7] *Deep learning pro segmentaci obrazu* [online]. HUMUSOFT, 5.11.2017 [cit. 2017-12-30]. Dostupné z: <http://sciencemag.cz/deep-learning-pro-segmentaci-obrazu/>
- [8] 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/](http://www.automa.cz/cz/casopis-clanky/deep-learning-v-prostredi-matlab-2017_05_0_10482/)
- [9] 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](http://www.automa.cz/Aton/FileRepository/pdf_articles/10482.pdf)
- [10] 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/>
- [11] *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>

- [12] *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/>
- [13] 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.
- [14] HORÁK, Karel. *Introduction to Convolutional Neural Networks* [pdf]. VUT, 2017 [cit. 2018-01-03]. Dostupné z: [http://midas.uamt.feec.vutbr.cz/ROZ/roz\\_cz.php](http://midas.uamt.feec.vutbr.cz/ROZ/roz_cz.php)
- [15] 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](http://midas.uamt.feec.vutbr.cz/POV/pov_cz.php)
- [16] 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](http://midas.uamt.feec.vutbr.cz/POV/pov_cz.php)
- [17] 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/>
- [18] 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/>
- [19] CHILAMKURTHY, Sasank. *A 2017 Guide to Semantic Segmentation with Deep Learning* [online]. July 5, 2017 [cit. 2018-05-09]. Dostupné z: <http://blog.qure.ai/notes/semantic-segmentation-deep-learning-review>
- [20] 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](http://www2.humusoft.cz/www/papers/tcc17/TCC2017_Prednaska_Deep_Learning.pdf)
- [21] JIRSÍK, Václav. *Umělé neuronové sítě - základy: Moderní prostředky v automatizaci* [online]. In: . Březen 2018 [cit. 2018-05-06].
- [22] 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>

[23] *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>

[24] 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.

[25] *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>

[26] *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>

[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 strojíního inženýrství. Vedoucí práce Ing. JIŘÍ KREJSA, Ph.D.

[28] *Semantic Segmentation using Fully Convolutional Networks over the years* [online]. 01.06.2017 [cit. 2018-05-09]. Dostupné z: [https://meetshah1995.github.io/semantic-segmentation/deep-learning/pytorch/visdom/2017/06/01/semantic-segmentation-over-the-years.html#sec\\_fcn](https://meetshah1995.github.io/semantic-segmentation/deep-learning/pytorch/visdom/2017/06/01/semantic-segmentation-over-the-years.html#sec_fcn)

[29] 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.

[30] *Synthia dataset* [online]. Barcelona, 2018 [cit. 2018-05-08]. Dostupné z: <http://synthia-dataset.net/>

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

[32] *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

[33] *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](http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--adaptacni-dynamika-neuronu--uceni-s-ucitelem)

[34] Úvod do neuronových Úvod do neuronových sítí. *StatSoft* [online]. [cit. 2018-01-  
02]. Dostupné z:  
[http://www.statsoft.cz/file1/PDF/newsletter/2013\\_02\\_05\\_StatSoft\\_Neuronove\\_site\\_link  
y.pdf](http://www.statsoft.cz/file1/PDF/newsletter/2013_02_05_StatSoft_Neuronove_site_linky.pdf)

[35] ZAPLETAL, Ondřej. *Rozpoznávání obrazů konvolučními neuronovými sítěmi -  
základní koncepty* [online]. 2017 [cit. 2018-01-03]. Diplomová práce. VUT FEKT.  
Vedoucí práce Ing. Karel Horák, Ph.D.: