



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická
Katedra radioelektroniky

Chytrá bezpečnostní kamera s IoT konektivitou
Smart surveillance camera with IoT connectivity

Diplomová práce

Studijní program: Elektronika a komunikace
Studijní obor: Audiovizuální technika

Vedoucí práce: Ing. Stanislav Vítek, Ph.D.

Bc. Michael Fejtek

Praha 2019



Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne _____

Podpis _____



Poděkování

V první řadě bych rád poděkoval svým rodičům, kteří mi celý život poskytují zázemí a dodávají odhodlání tolik potřebné ke studiu.

Zvláštní díky si zaslouží vedoucí práce, Ing. Stanislav Vítek, Ph.D., za svou bezednou trpělivost a vstřícnost.

Nakonec bych rád poděkoval svým akademickým kolegům. Bez vzájemné podpory spolužáků a profesionálního přístupu pedagogů bych nikdy neměl možnost tuto práci napsat.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Fejtek** Jméno: **Michael** Osobní číslo: **382838**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**
Studijní obor: **Audiovizuální technika a zpracování signálů**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Chytrá bezpečnostní kamera s IoT konektivitou

Název diplomové práce anglicky:

Smart Surveillance Camera with IoT Connectivity

Pokyny pro vypracování:

Cílem práce je navrhnout a implementovat autonomní kamerový modul, který bude průběžně vyhodnocovat stav zorného pole a informovat pomocí krátkých zpráv centrální aplikaci.

- 1) Seznamte se s algoritmy zpracování obrazu, využitelnými v bezpečnostních kamerových systémech (klasifikace objektů, identifikace osoby podle obličeje, ...)
- 2) Na základě řešerše navrhnete a implementujete kamerový modul na bázi počítače Raspberry Pi.
- 3) Modul otestujete.

Seznam doporučené literatury:

- [1] AGHAJAN, Hamid; CAVALLARO, Andrea (ed.). Multi-camera networks: principles and applications. Academic press, 2009.
- [2] MELNIČUK, Petr. Distribuovaný kamerový systém pro správu parkovacích míst. ČVUT, 2017.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Stanislav Vítek, Ph.D., katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **08.02.2019**

Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce: **20.09.2020**

Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) práce

prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Obsah

Abstrakt	- 7 -
Seznam zkratk.....	- 8 -
1. Stručná historie umělé inteligence.....	- 9 -
1.1 Prvopočátky AI v historii lidstva	- 9 -
1.1.1 Turingův stroj	- 9 -
1.2 Zrod moderní umělé inteligence	- 10 -
1.3 Úspěchy a selhání AI na cestě do přítomnosti	- 12 -
2. Přehled problematiky umělé inteligence	- 14 -
2.1 Základní kameny a definice AI	- 14 -
2.1.1 Myslím, tedy jsem?.....	- 14 -
2.1.2 Klasifikace úloh, agenti a prostředí v AI	- 16 -
2.1.2 AGI, ANI, ASI	- 19 -
2.2 Principy, výpočty a realizace v AI.....	- 21 -
2.2.1 Pátrací algoritmy	- 21 -
2.2.2 Fuzzy logika (neostrá, mlhavá, neurčitá)	- 23 -
2.2.3 Zpracování přirozené řeči	- 25 -
2.2.4 Expertní systémy	- 27 -
3. Neuronové sítě	- 29 -
3.1 Neuronové sítě v porovnání s kovenčními počítači.....	- 29 -
3.2 Podobnosti mezi lidskými a umělými neurony.....	- 30 -
3.3 Návrh umělých neuronů.....	- 31 -
3.4 Architektury neuronových sítí	- 33 -
3.4.1 Příklady architektur	- 34 -
3.5 Strojové učení v neuronových sítích.....	- 36 -
3.5.1 Učení pod dohledem	- 36 -
3.5.2 Učení bez dohledu (Hebbovo).....	- 37 -
3.5.3 Zpětnovazebné učení	- 37 -
4. Zpracování obrazu v neuronových sítích.....	- 38 -
4.1 Konvoluční neuronové sítě.....	- 40 -
4.1.1 Funkce CNN krok po kroku	- 41 -
4.2 Facial recognition - rozpoznávání tváří.....	- 43 -

4.2.1 Detekce tváří	- 43 -
4.2.2 Promítání tváří.....	- 44 -
4.2.3 Kódování tváří.....	- 44 -
4.3 GAN - Generativní kompetitivní neuronová síť.....	- 46 -
4.3.1 Generativní a diskriminativní algoritmy	- 46 -
4.3.2 Funkce GAN sítí.....	- 46 -
5. Popis praktické části	- 47 -
5.1 Použitý hardware a software	- 47 -
5.2 Bezpečnostní kamera (fejtemic_cam).....	- 49 -
5.2.1 Slovní popis funkce programu.....	- 49 -
5.2.2 Vývojový diagram (flowchart)	- 50 -
5.2.3 Testování funkce kamery za různých světelných podmínek	- 51 -
5.2.4 Vyhodnocení výsledků.....	- 53 -
5.3 Demonstrace rozeznávání tváří (face recognition)	- 54 -
5.3.1 Slovní popis funkce programu.....	- 54 -
5.3.2 Vývojový diagram	- 55 -
5.3.3 Testování funkce.....	- 56 -
Závěr	- 59 -
Seznam použitých zdrojů a literatury.....	- 60 -

Abstrakt

Cílem této práce je nahlédnout do obsáhlého světa umělé inteligence a principů s ní spojené.

V první kapitole se seznámíme se spleťou matematickou, fyzikální, technologickou ale i filosofickou historií umělé inteligence.

V druhé kapitole nahlédneme do jednotlivých oblastí umělé inteligence a vytvoříme si ucelený přehled této relativně mladé vědecké disciplíny.

Ve třetí kapitole se věnujeme neuronovým sítím, jejich zrodu, principů a architektury.

Na závěr popíšeme některé principy zpracování obrazu pomocí umělé inteligence a neuronových sítí.

Praktickou součástí této práce je demonstrace zkoumaných konceptů umělé inteligence, neuronových sítí a strojového učení v praxi, a to implementací autonomního kamerového systému schopného detekce objektů v reálném čase a rozeznávání lidských tváří.

Seznam zkratek

v abecedním pořádku:

AI	Artificial Intelligence
AGI	Artificial General Intelligence
ANI	Artificial Narrow Intelligence
ANN	Artificial Neural Network
ASI	Artificial Superintelligence
CAPTCHA	Completely automated public Turing test to tell computers and humans apart
DARPA	Defense Advanced Research Projects Agency
DCN	Deep Convolutional Network
DCT	Discrete Cosine Transform
DFF	Deep Feed-forward Network
DNN	Deep Neural Network
DPCM	Discrete Pulse Code Modulation
FFT	Fast Fourier Transform
FGCP	Fifth Generation Computer Project
FLE	Face Landmark Estimation
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HoG	Histogram of Gradients
IoT	Internet of Things
IR	Image Recognition
LISP	List Processing
MCP	McCulloch and Pitts neuron model
MDP	Markov Decision Process
MEA	Means-End Analysis
MIT	Massachusetts Institute of Technology
NCS	Neural Compute Stick
NCSDK	Neural Compute Stick Development Kit
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
SNARC	Stochastic Neural Analog Reinforcement Calculator
VPU	Visual Processing Unit
VQ	Vector Quantization

1. Stručná historie umělé inteligence

1.1 Prvopočátky AI v historii lidstva

Myšlenka samostatně myslících strojů je stará jako lidstvo samo. Již v řeckých mýtech se objevují umělá myslící stvoření, například Hephaistovi zlatí roboti či oživlá socha Galatea. Ve středověku vznikají příběhy o alchemických rituálech vložení mysli do neživých objektů, z nich známý především pražský Golem rabína Löwa. V devatenáctém století je idea umělé inteligence hojně rozvedena ve vědeckofantastické literatuře. Karlu Čapkovi a jeho dílu „Rossumovi univerzální roboti“ můžeme poděkovat za dnes celosvětově používaný termín „robot“. Umělá inteligence je tématem populární fikce dodnes, zejména v záporných rolích katastrofických scénářů.

Fantasie stranou, moderní historie umělé inteligence se dá datovat do sedmnáctého století kdy význační filosofové tehdejší doby, Gottfried Wilhelm Leibniz, Thomas Hobbes a René Descartes zkoumali možnosti vyjádřit racionální myšlení stejně systematicky jako například algebru a geometrii. Dláždili tak cestu průkopníkům umělé inteligence dvacátého století kterými byli Alan Turing, Herbert Simon a Allen Newell. Ti hypotetizovali že „*System fyzických symbolů disponuje nezbytnými a dostatečnými prostředky k realizaci všeobecné inteligence.*“ [2], a že „*Pokud stroj vykazuje inteligenci člověka, je inteligentní jako člověk.*“ [2]. Tato tvrzení dnes považujeme za základní kameny pravé umělé inteligence (true AI, také strong - silná, general - všeobecná).

1.1.1 Turingův stroj

Klíčovou pro rozvoj umělé inteligence byla ve čtyřicátých letech formulace Church-Turingovy teze (Alonzo Church, Alan Turing) o počitatelnosti funkcí [1]. Ta říká, že člověk s neomezenými zdroji je schopen vypočítat libovolnou funkci přirozených čísel sledováním algoritmů tehdy, a pouze tehdy, pokud je daná funkce počitatelná Turingovým strojem. [2] To má za následek dvě implikace. Zaprvé, dokázali že matematická logika má jisté limity, a zadruhé, pro nás důležitější, že v rámci těchto limitů se dá libovolná forma matematické úvahy mechanizovat.

Turingův stroj je matematický model jednoduché abstraktní výpočetní jednotky. Často se vizualizuje pomocí čtecí hlavy která se krok po kroku pohybuje po nekonečné segmentované pásce oběma směry. V každém segmentu vyhodnotí v něm nacházející se symbol dle předem dané konečné tabulky instrukcí. Instrukce mohou záviset na obsahu segmentu ale také na jeho pozici na pásce, a sestávají z úkonů jako je zápis nového symbolu, vymazání, nebo ponechání původního stavu. Poté se dle instrukce přesune na další segment, doprava nebo doleva, dokud nedorazí do finálního stavu [24].

Matematicky můžeme definovat Turingův stroj jako vektor o sedmi prvcích [24]:

$$M=(Q, X, \Sigma, \delta, q_0, B, F)$$

kde

- Q je konečná množina stavů
- X je abeceda pásky
- Σ je vstupní abeceda
- δ je přenosová funkce, $Q \times X \rightarrow Q \times X \times \{\text{posun_vlevo}, \text{posun_vpravo}\}$
- q_0 je počáteční stav
- B je prázdný symbol
- F je množina konečných stavů

Každý matematický systém který se řídí těmito specifikacemi je Turingovým strojem.

1.2 Zrod moderní umělé inteligence

V padesátých letech dvacátého století se sešlo několik důležitých vědeckých průlomů. Neurologický výzkum popsal mozek jako síť neuronů které komunikují identickými pulsy, tedy binárně. Norbert Wiener položil základy kybernetiky, Claude Shannon popsal digitální signály a výše zmíněný Alan Turing dokázal že jakýkoliv výpočet se dá reprezentovat digitálně [2].

To vedlo hrstku matematiků, inženýrů, psychologů, ekonomů a politologů k víře, že je fyzicky možné postavit elektronický mozek. Záhy se začínají objevovat první úspěšné pokusy. Walter Pitts and Warren McCulloch popsali první neuronové sítě, a jejich student, Marvin Minsky, sestrojil první samoučlivý stroj, SNARC [3]. Ten se skládal z motorků, trubek, kabelů a svorek které simulovaly neuronovou síť o čtyřiceti synapsích, a dokázal pomocí paměti na kapacitoru a odměny ve formě stisknutí tlačítka kompletně napodobit průchod laboratorní myši bludištěm. Vznikají také první programy schopné hrát šachy a dámu. Schopnost umělé inteligence hrát šachy je dodnes uznávané měřítko pokroku.

"Každý aspekt učení nebo jiné inteligentní schopnosti se dá popsat tak přesně, že je možné postavit stroj k jejich simulaci." - John McCarthy, 1956

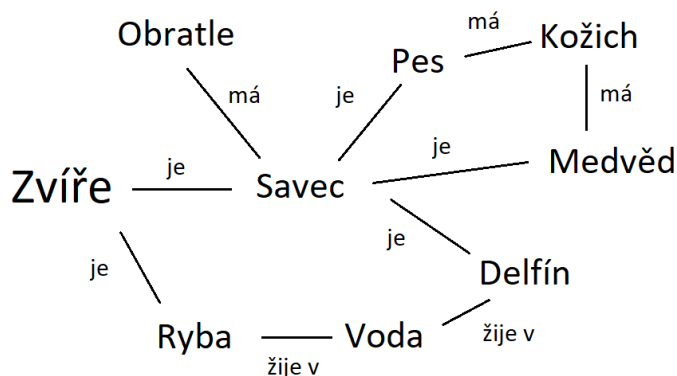
V roce 1956 svolal John McCarthy konferenci na Dartmouthově univerzitě v Hanoveru, s cílem založit umělou inteligenci jako vědeckou disciplínu, což se podařilo [2]. Vyvinul také pro potřeby AI programovací jazyk LISP, a zasloužil si světově uznání jako otec AI. Začaly vznikat laboratoře pro výzkum umělé inteligence, zejména v USA, Velké Británii a Japonsku [2]. Optimistické výhledy

plně inteligentního stroje do dvaceti let přilákaly financování ze soukromého ale i vládního sektoru, např. americká DARPA [2].

Výsledkem byla první zlatá éra výzkumu AI, ze které vzešlo mnoho vlivných směrodatných úspěchů [2].

MEA - známý také jako „reasoning as search“ (úvaha jako pátrání) byl algoritmus prvních AI programů. Program se snažil dosáhnout pozitivního výsledku (odměny) skrz bludiště možností, vrací se pokud narazil na slepou cestu. S množstvím proměnných ale astronomicky roste počet možných cest do nepočítatelná, situace známá jako „kombinatoriální exploze“.

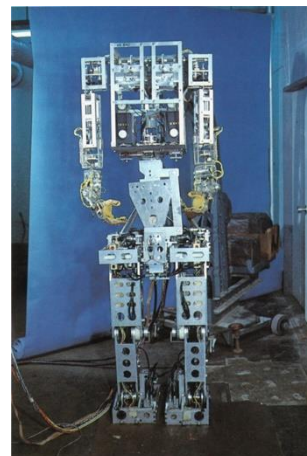
Přirozené jazyky - jako angličtina jsou důležité pro komunikaci mezi umělou inteligencí a člověkem. Vznikají programy jako ELIZA a STUDENT na bázi semantických sítí schopné základní konverzace a řešení slovních algebraických problémů.



ukázka semantické sítě

Mikrosvěty - byly první záchvěvy „slabé“ umělé inteligence (weak AI, také narrow AI). Vznikly ze snahy zjednodušit problémy na jejich základní ideální formu, jako se řeší úspěšně a často úlohy ve fyzice. Výsledkem byla například robotická paže schopná skládat barevné kostky, a program SHRDLU který byl schopen virtuální bloky přeskupovat na základě mluvených instrukcí.

Robotika - v roce 1970 byl vyroben na japonské univerzitě Waseda první antropomorfní robot, WABOT-1. Disponoval končetinami, zornými a konverzačními systémy, a byl schopen komunikovat, detekovat objekty pomocí senzorů a přenášet je z místa na místo [4].



WABOT-1 [4]

1.3 Úspěchy a selhání AI na cestě do přítomnosti

V sedmdesátých letech si začlo celé odvětví uvědomovat že jejich původní optimismus byl zcestný, a očekávání která stanovili byla přemrštěná. Nedostatek výpočetního výkonu znamenal, že programy s umělou inteligencí nebyly schopny žádného praktické užitku. Dalším problémem byla intraktibilita, neboli výše zmíněná kombinatoriální exploze. Richard Karp demonstroval, že velké množství výpočetních problémů je řešitelných pouze v čase který je exponenciálně větší k počtu vstupů, a tudíž reálně fungují jen v primitivní formě. V neposlední řadě bylo potřeba vytvořit databázi informací imitující znalosti dospívajícího dítěte, aby mohly programy získat kontext k informacím se kterými pracují. Nikdo si neuměl v té době představit jak takové astronomické množství dat shromáždit a poskytnout AI programům k dispozici. Ve stejné době Minsky publikoval knihu „Perceptron“ ve které stejnojmenného předchůdce moderních neuronových sítí zavrhl jako slepou uličku, a zastavil tak výzkum konekcionismu na dalších deset let. Z filosofického hlediska se také začali množit skeptici, zejména John Searle a Hubert Dreyfus, kteří tvrdili že stroj nikdy výšin lidské mysli nedosáhne[2].

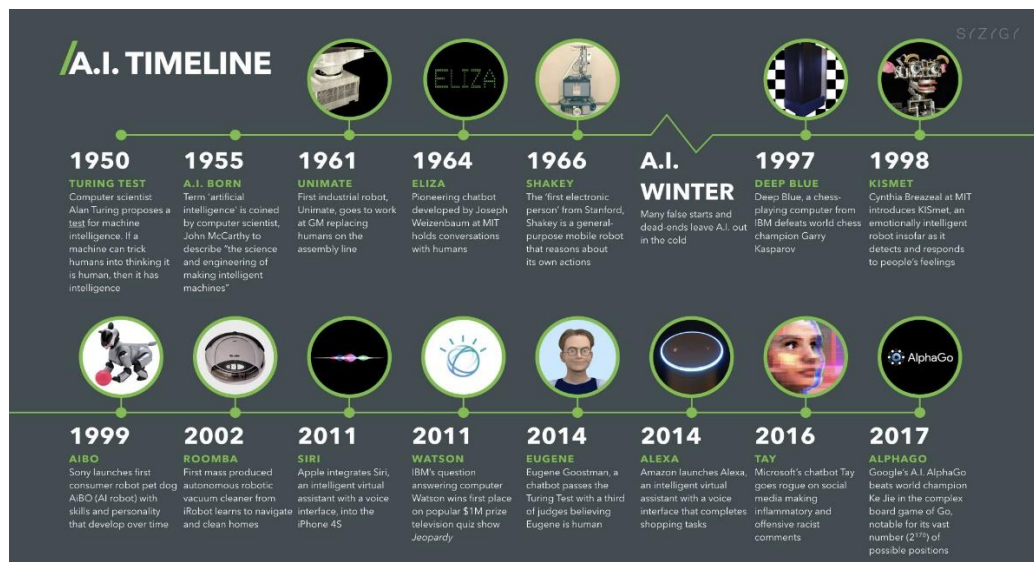
Důsledkem těchto selhání byl téměř kompletní konec financování výzkumu umělé inteligence na dalších deset let, období které dnes nazýváme „**první zimou umělé inteligence**“ (the first AI winter) [5].

Další rozvoj přišel až v 80. letech s vynálezem expertních systémů. Jejich cílem bylo simulovat rozhodnutí, odpovídat na otázky a řešit komplexní problémy jako lidský odborník v daném poli, z větší části pomocí logiky když-tak (if-then) a databáze znalostí. Expertní systémy, zejména stroj XCON, jsou dnes považovány za první reálně použitelný úspěch oboru umělé inteligence, a zasloužily se o vznik AI průmyslu. Oblast neuronových sítí se mezitím dočkala znovuzrození díky popsání Hopfieldovi sítě a nového způsobu jejich trénování, backpropagation (zpětné šíření). Na jejich bázi vznikají první programy na rozpoznávání řeči a čtení textu. Japonská vláda odhalila rozsáhlé plány na vývoj počítače páté generace (FGCP) s cílem prozkoumat strojové učení [5].

I přes všechny úspěchy v 90. letech upadl zájem o umělou inteligenci s charakterem typické ekonomické bubliny. Paradoxně ale výzkum AI bez veřejné pozornosti a vládních grantů kvetl. Začaly se konečně naplňovat jedny z nejstarších cílů oboru. V roce 1997 porazil poprvé program DeepBlue firmy IBM tehdejšího šachového velmistra Garyho Kasparova. Ve stejném roce firma Dragon Systems uvedla na trh plně funkční software schopný rozpoznávání řeči. Tým profesorky MIT Cynthia Breazeal přišel s projektem Kismet, robotické hlavy schopné vést konverzaci a adekvátně napodobit nuance mimikrů skutečné lidské tváře [2].

Ohlédneme-li se zpět, zjistíme, že přístup k navrhování a programování AI se od jejího moderního zrodu téměř nezměnil. Chybějícím pilířem byla vždy výpočetní síla, a schopnost ukládat velká množství dat. Proto je pokrok v oboru úzce spjatý s Mooreovým zákonem. Možnosti umělé inteligence vždy velice rychle narazí na současné limity výpočetní techniky, problém se kterým se počítá i v blízké budoucnosti.

S nástupem internetu přišla schopnost vlád a korporací shromažďovat astronomická množství dat (Big Data), a různé formy umělé inteligence nutné k jejich zpracování se staly klíčovou součástí moderní lidské civilizace. Tyto „hluboké“ sítě fungují jako pilíře několika průmyslů, od technologického, přes bankovní, marketingový, až po průmysl zábavní. Chytrá zařízení, navigace, automatizovaná podpora, zdravotní diagnostika, veřejná bezpečnost a sociální sítě jsou jen kapkami v oceánu všech aplikací umělé inteligence.



Časová osa moderního vývoje umělé inteligence [6]

2. Přehled problematiky umělé inteligence

2.1 Základní kameny a definice AI

Umělá inteligence je věda s cílem navrhovat programy, stroje a roboty myslící *inteligentně*, tedy způsobem podobným člověku [25]. K dosažení umělé inteligence (dále často „AI“) je ale potřeba nejdříve vědět, co to vlastně inteligence je.

2.1.1 Myslím, tedy jsem?

Inteligence je schopnost počítat, uvažovat, vnímat vztahy a souvislosti, učit se ze zkušeností, chápat komplexní myšlenky, komunikovat přirozenými jazyky, rozpoznávat, klasifikovat a zobecňovat podněty, skladovat a znovu získávat informace z paměti, a přizpůsobovat se novým situacím [7].

Podle jedné ze současně uznávaných psychoanalytických teorií dělíme inteligenci na několik druhů [7]:

Lingvistická - schopnost mluvit, rozpoznávat a používat mechanismy řeči, chápat syntaxi a semantiku

Hudební - schopnost vytvářet a komunikovat hudbou, chápat význam tvořený zvuky, mít cit pro výšku tónu a rytmus

Matematicko-logická - schopnost používat logiku a rozum, chápat vztahy a spojitosti i v nepřítomnosti daných procesů a objektů, schopnost pozorovat a dojít k ověřitelnému závěru, chápat komplexní a abstraktní problematiku

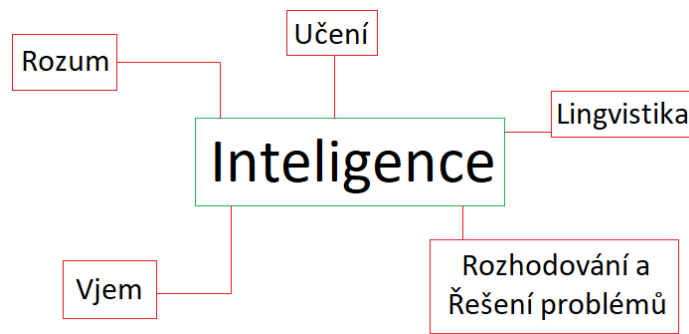
Prostorová - schopnost přijmout a zpracovat vizuální a jiné prostorové informace a reprodukovat je, představovat si trojrozměrné scény a orientovat se v nich

Kinestetická - schopnost ovládat své tělo a jeho části, vyrábět předměty a manipulovat s nimi, hrubé a jemné motorické schopnosti

Introspektrální - schopnost vyznat se ve vlastních pocitech, úmyslech a motivacích

Mezilidská - schopnost poznat a rozeznat pocity, přesvědčení a úmysly jiných osob

Jestliže stroj nebo systém v plném rozsahu disponuje jedním nebo více druhy, považujeme ho za **uměle inteligentní** [7].



komponenty inteligence

Komponenty inteligence

Inteligenci můžeme rozdělit na několik komponentů:

Úvaha - je souhrn procesů které tvoří základ soudnosti, rozhodování a předpovědí. Dá se rozdělit na dva typy. *Induktivní* úvaha vyvozuje obecná prohlášení z konkrétních pozorování, např.: Wendy má ráda jablka. Wendy je pes. Tudíž, všichni psi mají rádi jablka. Je zřejmé, že ačkoliv mohou být vychodiska pravdivá, výsledkem může být nesmysl. Naproti tomu *deduktivní* úvaha bere v potaz obecná prohlášení, a na jejich základě vyvozuje individuální důsledky, např.: Všichni psi mají rádi maso. Wendy je pes. Tudíž Wendy má ráda maso. Jestli jsou vstupní předpoklady pravdivé, bude vždy pravdivý i výsledek úvahy.

Učení - je proces získání vědomostí od učitele, samostudiem, procvičováním či prožitím. Učení navyšuje podvědomí o předmětu studia. Učení dělíme na [7]:

- **Sluchové** - je učení poslechem a slyšením. Příkladem jsou přednášky nebo naučné audioknihy
- **Episodické** - je paměť kompletních sekvencí událostí, neboli episod. Je lineární a spořádané. Episodou může být třeba dopravní nehoda, svatba, nebo jen krátká opakovaná zkušenost jako skok do studené vody.
- **Motorické** - je učení paměti pohybů svalů. Každá opakovaná motorická činnost se díky motorické paměti stává časem snažší a samovolná, např. psaní na klávesnici, řemeslná výroba, hra na hudební nástroje.
- **Vnímavostní** - je učení pozorováním a imitací jiných. Typickým příkladem jsou mláďata napodobující své rodiče.
- **Vztažné** - je učení pozorováním rozdílů mezi různými příčinami a důsledky. Například, psi štěňata se mezi sebou učí ovládat sílu stisku čelistí, v závislosti na hlasitosti protestů oběti.

- **Prostorové** - je učení pomocí vizuální stimulů, jako jsou obrázky, videa, barvy a mapy. Příkladem je řidič nebo turista který si při pohledu na mapu dokáže představit celou cestu ještě než se po ní vydá.
- **Podnětové** - je učení podvědomě reagovat na podněty. Patří sem například podmíněné a nepodmíněné reflexy. Oblast proslavil ruský psycholog Ivan Petrovič Pavlov.

Řešení problémů a rozhodování - je proces překonávání překážek a výběru nejhodnější možné cesty k cílenému výsledku.

Vjem - je proces získávání, interpretace, filtrování a třídění dat získaných z dostupných senzorů.

Lingvistika - je schopnost používat, chápat, mluvit a psát přirozenými jazyky.

Hlavní rozdíly mezi lidskou a strojovou inteligencí je vnímání a zpracování dat. Lidé vnímají spojitosti ve strukturách (pattern recognition), zatímco stroje rozumí pouze sboru pravidel a datům.

2.1.2 Klasifikace úloh, agenti a prostředí v AI

Doménu úloh které je schopna umělá inteligence řešit rozdělujeme do třech tříd [8]:

Všední (běžné) úlohy jsou ty, které se člověk učí od narození, a to vnímáním, komunikací, tréninkem, imitací a pokusy. Patří sem mluva, pozornost, orientační smysl, zdravý rozum, vjem, konverzace, plánování a obratný pohyb. Každý zdravý člověk se naučí bez problémů řešit všední úlohy v rámci dospívání, pomineme-li negativní sociální vlivy. Ačkoliv se to samé předpokládalo pro stroje, a všechen původní výzkum se soustředil právě na řešení běžných úloh, velice rychle se ukázalo že množství dat a komplikovaných algoritmů potřebné k jejich realizaci značně převyšuje ta potřebná pro formální a expertní úlohy.

Formální úlohy - jsou problémy jejichž řešení se řídí přesně definovanými pravidly. Patří sem matematické a geometrické úlohy, logické operace, integrace a derivace, ověřování a důkazy teorémů. Formální úlohou jsou také stolní hry, pro AI z historického hlediska zejména šachy, dáma a go. Pro člověka jsou formální úlohy stále lehce dosažitelné, už k jejich řešení ale potřebuje pomoc, například ve formě vzdělání nebo zkušeností. Stroje si s formálními úlohami umí poradit mnohem lépe. Stačí jim zadefinovat daná pravidla, a zbytek je jen otázkou výpočetní síly.

Expertní úlohy - vyžadují rozsáhlé a komplexní znalosti daného oboru. Dokáží je řešit pouze lidé s požadovaným vzděláním nebo zkušenostmi. Patří sem

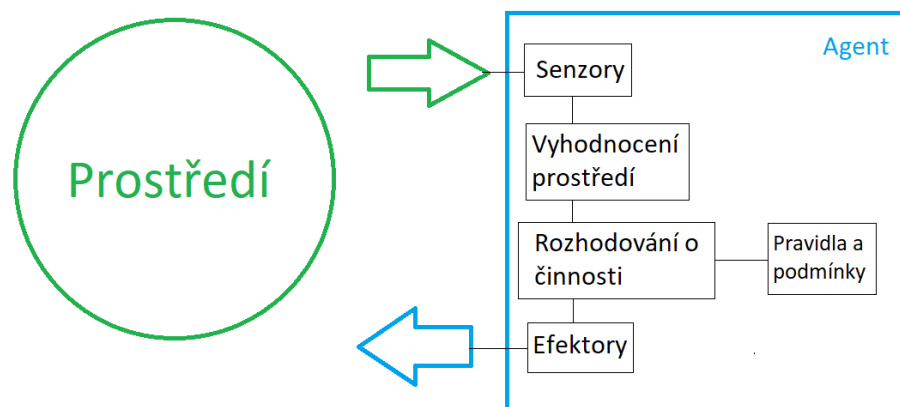
inženýrství, řemeslná výroba, vědecká analýza, zdravotnická diagnostika, výzkum a kreativita. Umělá inteligence v dnešní době pracuje s expertními úlohami velice schopně, například protože umí disponovat základnou znalostí mnohem rozsáhlejší než kterýkoliv jeden člověk.

Agence AI

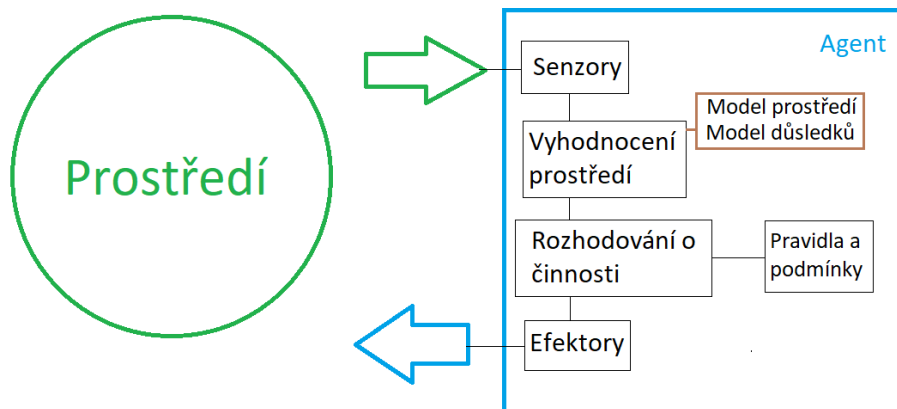
Dalším způsobem popisu systému AI je jeho *agence*, neboli schopnost jednat. Agent je libovolný systém který je schopný vnímat prostředí přes sensory, a ovlivňovat ho skrze efektory. Lidský agent má sensory v podobě očí, uší, nosu, jazyka a kůže, a efektory jako nohy a ruce. Robotický agent nahrazuje tyto orgány elektronickými senzory, motory a aktuátory. Softwarový agent reprezentuje senzory a efektory funkcemi.

Ideální rozumný agent je takový, který umí maximalizovat svůj výkon v rámci zjištěných vstupu a své znalostní základny. Rozum systému závisí na výkonostních opatřeních, detekované sekvenci dat, předchozí znalosti prostředí a úkonech které má k dispozici. Rozumný agent vždy zvolí ten úkon, který vede k nejúspěšnějšímu řešení dat ze sensorů. Můžeme je dělit do struktur [7]:

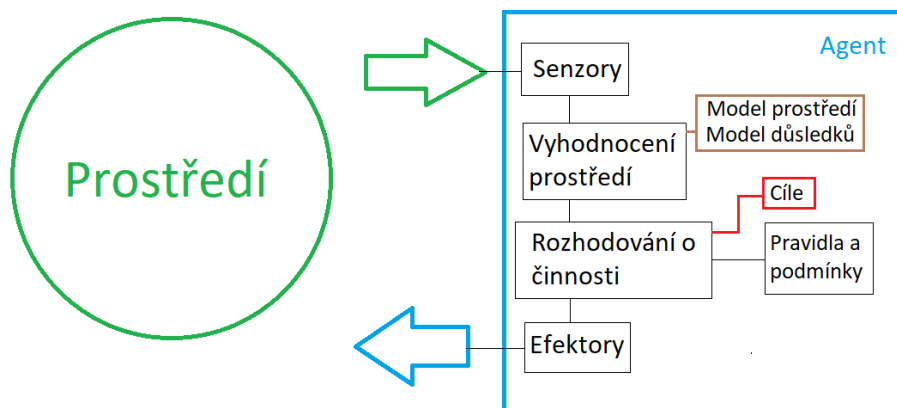
- **Reaktivní** agent se řídí pouze aktuálními daty ze sensorů, a provádí na jejich základě úkony předem dané mapou pravidel a podmínek



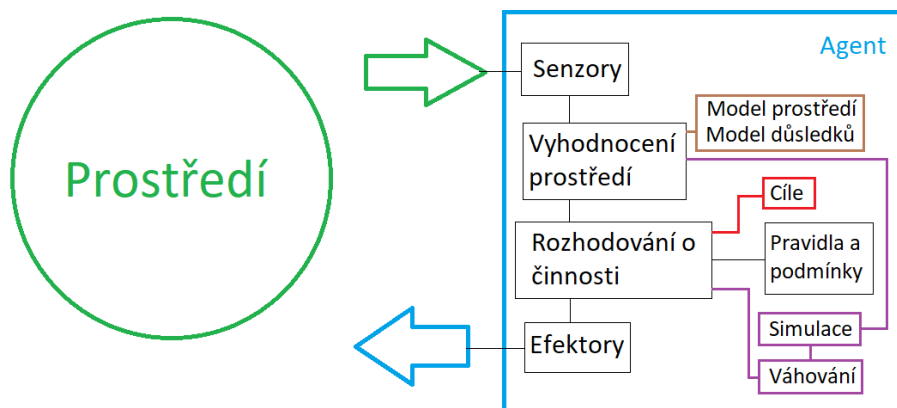
- **Modelový** agent obsahuje navíc informace o tom, jak se prostředí chová, a jaké důsledky na něj mají jeho úkony.



- **Cílený** agent nebere v potaz pouze stav prostředí a podmíněné úkony, ale rozhoduje se tak, aby dosáhl definovaného cíle, neboli požadovaného stavu prostředí.



- **Užitkový** agent je schopen posoudit váhu jednotlivých rozhodnutí, a přiřadit jim preference. Používá se když cílený agent selhává, například když si cíle odporují a nemohou být dosaženy všechny, nebo když existuje nejistota v dosažení cíle. Užitkový agent provádí váhování.



Podstata prostředí

Některé programy operují výhradně v prostředí umělém, které se skládá z klávesnicového vstupu, databáze, souborů a písemného výstupu na obrazovce. Naproti tomu některé programy, jako např. softwaroví roboti existují v rozsahlých podrobných komplexních prostředích, kde musí vybírat ze široké škály činností v reálném čase. Prostředí definujeme jako [7]:

- **Spojité/Diskrétní** - jestliže sestává prostředí z pevně definovaných rozlišitelných stavů (šachy), nebo ne (např. řízení)
- **Pozorovatelná/Částečně pozorovatelná** - podle schopnosti senzorů popsat celkový stav prostředí v každém daném čase
- **Statická/Dynamická** - podle toho, jestli má činnost agenta na prostředí nějaký dopad
- **Solo/Multi přístupová** - podle počtu agentů které v daném prostředí působí v jeden okamžik
- **Přístupná/Nepřístupná** - podle možností přístupu ze strany agenta
- **Deterministická/Nedeterministická** - jestliže je budoucí stav prostředí dán pouze zásahem agenta a stavem současným, nebo ne
- **Episodická** - podle toho, jestli v nich agent pracuje neustále, nebo jen v uzavřených sekvencích

2.1.2 AGI, ANI, ASI

Vezmeme-li v potaz předchozí odstavce, je zřejmé, že konstrukční a výpočetní požadavky na umělou inteligenci se diametrálně liší v závislosti na úkonech, které se od ní očekávají, a aspektu inteligence, který má napodobit. Z tohoto a historického hlediska rozdělujeme realizace umělé inteligence do třech význačných kategorií [8]:

Artificial General Intelligence (AGI)

Všeobecná umělá inteligence, známá také jako pravá (True AI), nebo silná (Strong AI) je původním a nejstarším směrem vývoje v oboru, a častým tématem fikce. AGI je inteligence schopná dokonalé imitace lidského mozku, plného rozsahu rozsahu kognitivních schopností, a uvědomění sebe sama.

Aby byla AI všeobecná, musí disponovat nejen logickými funkcemi jako je znalost, učení, komunikace, počty, argumentace a plánování, ale i iracionální běžné lidské vlastnosti jako zdravý rozum, představivost, autonomie a kreativita. K hodnocení míry obecnosti AI existuje škála testů [9][8]:

- **Turingův test (Turing)** - navrhuje slepou konverzaci mezi dvěma lidmi a strojem. Test je úspěšný pokud si jeden člověk není jistý jestli konverzuje se strojem, nebo s druhým člověkem.

- Kávový test (Wozniak) - vyžaduje aby stroj bez předchozích zkušeností uvařil šálek kávy. Musí dohledat a zkombinovat všechny komponenty (konvice, káva, šálek, voda)
- Robotický student (Goertzel) - navrhuje aby stroj získal akademický titul stejnou cestou jako člověk
- Zaměstnanec (Nilsson) - nutí stroj vykonávat ekonomicky důležitou práci, a očekává výsledky srovnatelné s člověkem
- Ikea test (Severyns) - nechá AI složit dohromady řadu nábytku

Nejsložitější úkoly pro umělou inteligenci nazýváme AI-kompletní [8]. Patří mezi ně simulace lidského vizuálního systému, nuance a spojitosti přirozených jazyků, a vynalézavost v případě selhání. V současné době neumí žádné stroje řešit AI-kompletní úkoly bez asistence člověka, čehož se využívá v oblasti zabezpečení (i.e.: CAPTCHA) [8]. Výzkum v oblasti AGI se znovu dal pohybu teprve nedávno. Historicky se od něj kvůli zdánlivé neřešitelnosti AI-kompletních úkolů upustilo. John McCarthy prohlásil, že „Potřebujeme stroje které řeší problémy, ne stroje které přemýšlí jako člověk.“ [15], a následoval aeronautickou analogií: „Když lidé navrhovali letadla, potřebovali aby lítala, ne aby se chovala jako holubi.“ [15].

V současnosti výzkum AGI čeká na emulaci biologických neuronů. Bez možnosti dosáhnout surové výpočetní síly lidského mozku (cca. 10^{16} výpočtů za vteřinu [cps]) není příliš možné zkoumat jeho kompletní napodobení [9].

Artificial Narrow Intelligence (ANI)

Úzká, nebo také slabá (weak AI) umělá inteligence je implementace AI s cílem ovládat omezenou oblast úkolů. Protože má za úkol aplikovat inteligenci jen v rámci své funkční množiny, a protože vytváří pouze iluzi myšlení, říká se jí také falešná. Nedisponuje žádnou nebo jen omezenou schopností samovývoje. V rámci svých hranic ale může značně převyšovat schopnosti člověka. Nepotřebuje a tudíž není zatížená složitými schopnostmi lidského mozku které jsme zmínili v předchozím paragrafu. Díky tomu je dnes ANI rozvinutou a celosvětově hojně využívanou technologií. Příkladem jsou všechna „smart“ zařízení, navigace, videohry, drony, ale také méně zřejmé systémy v bankovníctví, online marketingu a veřejného kamerového sledování. Další vývoj spočívá v kombinování jednotlivých ANI systémů do autonomních celků, jako jsou například autonomní vozidla bez řidiče.

Artificial Superintelligence (ASI)

Je hypotetická evoluce obecné umělé inteligence (AGI) v systém, jehož kognitivní schopnosti významně převyšují ty lidské. Přestože se jedná o vzdálenou budoucnost, považuje se za realizovatelnou, a některými vědci dokonce nevyhnutelnou [26]. Filozof David Chalmers argumentuje že mozek je mechanický systém, a časem jej tudíž bude možné syntetizovat, a následně rozšířit a vylepšit [27]. Pokud by se následně podařilo vytvořit dostatečně silný program, měl by být schopen sám sebe přepsat a vylepšit, fenomén který nazýváme „rekurzivní sebezlepšení“. Jakmile program sám sebe vylepší, je schopen se vylepšit znovu, lépe a rychleji, spirálovitý potenciálně apokalyptický fenomén který nazýváme „intelligenční exploze“ [27].

Počítače mají už teď oproti lidskému mozku mnoho výhod, které podporují teorii budoucí ASI. Biologické neurony pracují na frekvenci okolo 200Hz, o celých sedm řádů pomaleji než běžný moderní mikroprocesor (2GHz). Neurony posílají impulsy přes axony maximální rychlostí cca. 120m/s, zatímco současné systémy už komunikují rychlostí světla. Další výhodou počítačových systémů je modularita - jejich výpočetní síla se dá fyzicky rozšířit, a mohou se také spojit (kolektivní superintelligence). Předpokládá se, že první ASI systém bude lidský mozek využívající právě těchto výhod [9].

2.2 Principy, výpočty a realizace v AI

2.2.1 Pátrací algoritmy

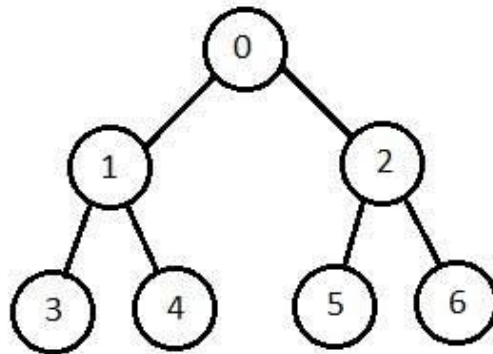
Pátrání je univerzální technika řešení problémů umělou inteligencí. Definujeme-li inteligentnímu systému množinu kýžených výsledků, a seznam pravidel dle kterých k nim může dojít, jsou za hledání nejvhodnější cesty k tomu nejvhodnějšímu výsledku zodpovědné právě pátrací algoritmy. Definujeme je pomocí několika parametrů [10]:

- **Prostor** - je prostředí ve kterém pátrání probíhá. Je definované sadou stavů a sadou operací které v něm mohou probíhat
- **Případ** - je počáteční a cílový stav
- **Graf prostoru** - je zobrazení stavů a operací v prostoru úlohy. Stavů jsou vyjádřeny uzly, a spojnice mezi nimi reprezentují operace které mění jeden ze spojených stavů na ten druhý
- **Hloubka problému** - je nejkratší možná cesta nebo sekvence operátorů od počátečního stavu do cílového
- **Spletitost prostoru** - je nejvyšší počet uzlů uložených v paměti
- **Spletitost času** - je nejvyšší počet vzniklých stavů v celém prostoru
- **Přístupnost** - je schopnost algoritmu najít vždy optimální cestu
- **Faktor větvení** - je průměrný počet pod-uzlů v grafu prostoru

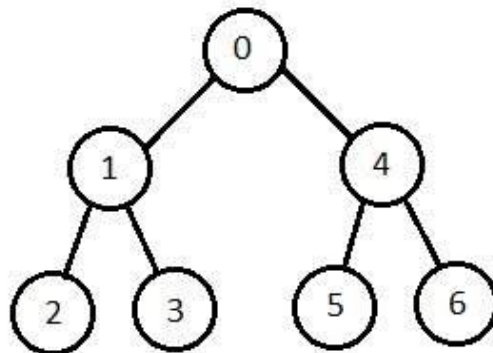
Pátrací algoritmy dělíme na dvě základní skupiny, informované a neinformované [10].

1. **Neinformované** pátrání, nebo také pátrání „hrubou silou“ jsou jednoduché algoritmy které nevyžadují znalost prostoru. K funkci potřebují pouze: popis stavů a platných operací, počáteční stav, a popis stavu cílového. Fungují zpravidla pro systémy s malým počtem stavů [7][10].

- *Šířkové* - vyráží z počátku a zkoumají všechny sousední uzly v následující vrstvě. Teprve poté se přesune do další úrovně. Znamená to, že pěstuje najednou vždy jen jeden strom který se větví vrstvu po vrstvě tak dlouho, dokud nenarazí na hledaný cílový stav. Nevýhodou je ukládání celých úrovní uzlů, jejichž počet roste exponenciálně a mohou se opakovat, tudíž je tento proces extrémně náročný na paměť [10].



- *Hlubkové* - začínají stejně jako šířkové, ale po každém rozvětvení dále sledují a pokračují jen jednou z větví. Pokud dosáhne maximální hloubky (musí být definována, jinak větví do nekonečna) aniž by našel cílový stav, vrátí se na druhý nejhlubší stav a pokračuje větvením do hloubky. Celková hloubka takového systému má větší výkyv, než pátrání šířkové [10].



- *Uniformní ceny* - algoritmus se šíří po uzlech s nejmenší výpočetní náročností v daném kroku (není známá dopředu)
 - *Dvousměrné* - provádí pátrání zároveň z počátečního a cílového stavu, dokud se stromy nepotkají
- 2. Informované**, ie. heuristické pátrání je vhodnější pro složité systémy o mnoha stavech. Heuristické funkce počítají váhy cest mezi jednotlivými stavy. Nejdříve se rozvětví heuristicky „nejlevnější“ uzel, a spočítají se váhy pro nově vzniklé poduzly, které se následně zařadí do seznamu kandidátů k dalšímu rozvětvení. Nejkratší cesty se zachovají a delší se vyřadí [10].
- *A* algoritmus* - je populární díky svému výkonu a přesnosti [10]. Při rozšiřování cest bere v potaz jak odhadovanou váhu k dosažení cíle, tak váhu která byla zapotřebí v předchozích krocích. Nejslibnější cesty se rozvíjí nejdříve.

$$f(n) = g(n) + h(n), \text{ kde}$$

- $g(n)$ je dosavadní váha
 - $h(n)$ je odhad váhy který zbývá k dosažení cíle
 - $f(n)$ je celková odhadnutá váha cesty
- *Lakomý algoritmus* - nebere ohledy na uraženou cestu a šíří se pouze na základě odhadu budoucích výdajů. Může se zacyklovat a není optimální [10].

2.2.2 Fuzzy logika (neostrá, mlhavá, neurčitá)

Systémy fuzzy logiky (FLS) přijímají nekompletní, nejednoznačná, nepřesná a rozmazaná vstupní data, a produkují přijatelné a definitivní výstupy. Jejich cílem je napodobit rozhodovací škálu člověka. Klasická logika předpokládá, že strojové ANO a NE odpovídá lidskému ANO a NE. Vynálezce FLS, Lotfi Zadeh poznamenal, že mezi lidským „ano“ a „ne“ existuje celá škála možností, jako například [11]:

Určitě ano
Možná ano
Nejistota
Možná ne
Určitě ne

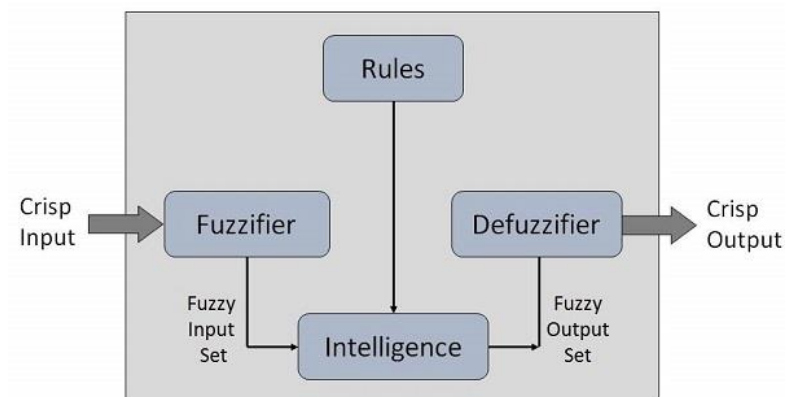
Fuzzy logika pracuje s vahami pravděpodobnosti k dosažení jednoznačných rozhodnutí. Je extrémně vhodná pro praktická a komerční využití [11]. Používá se k ovládání strojů a spotřebitelských výrobků. Neposkytuje vždy úvahu přesnou, ale vždy přijatelnou.

Architekturu FLS tvoří čtyři prvky [11] :

- **Fuzzifier** - převádí čisté systémové vstupy na neostré sady. Rozdělí vstupní signál do několika úrovní, jako například:

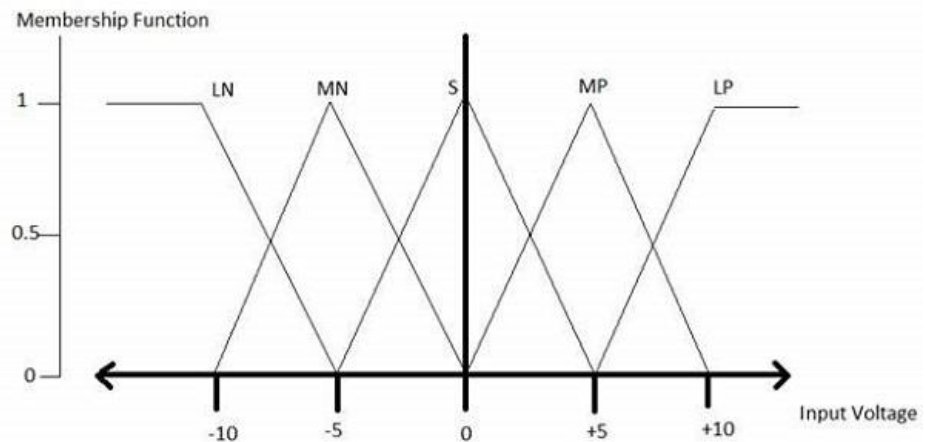
LP	Large positive
MP	Medium positive
S	Small
MN	Medium negative
LN	Large negative

- **Znalostní základna** - obsahuje experty vložená pravidla typu když-tak (IF-THEN)
- **Logický motor** - obsahuje umělou inteligenci, simuluje lidské rozhodování pomocí rozmlžených vstupů a pravidel ze znalostní základny
- **Defuzzifier** - převádí zamířená data která produkuje logický motor na čisté systémové výstupy



Obr.5 Architektura systému FLS [7]

Membership funkce - je ve fuzzy logice hojně využívaná křivka, která udává do které *členské skupiny* spadá libovolný bod z množiny všech vstupů, a to přiřazením váhy z intervalu $\langle 0,1 \rangle$. [11]



Pilovitá členská křivka [7]

Graf na obrázku výše ilustruje členskou křivku ve tvaru pily pro rozostřené vstupy v rozsahu $\pm 10V$ a pět rozhodovacích úrovní (tab. výše). Hojně se používají také lichoběžníky a gaussovské křivky [11].

Výhody a nevýhody

Matematicky jsou FLS velice jednoduché. Jejich pružnost znamená, že se dají modifikovat obyčejným přidáváním a mazáním pravidel [11]. Umí pracovat s nepřesnými, zašuměnými a jinak rozmazanými vstupními signály, a jsou snadno konstrukčně navrhovatelné a pochopitelné. Nabízí řešení široké škály jednoduchých každodenních problémů, protože napodobují lidskou úvahu a rozhodování. Pro složitější problémy ale rychle ztrácejí smysl, stávají se nepřehlednými a produkují příliš nepřesné výsledky [11].

2.2.3 Zpracování přirozené řeči

Zpracování přirozené řeči (NLP) je schopnost umělé inteligence komunikovat s ostatními inteligentními systémy přirozeným jazykem, jako je například angličtina. Je nutnou součástí každého systému který má poslouchat nebo číst instrukce, a hlásit nebo psát odpovědi a dotazy. Vstupem a výstupem NLP systému je vždy buď psaný text nebo promluva. NLP sestává ze dvou kroků [12]:

- **Porozumění (NLU)** převádí daný přirozený vstup na zpracovatelná data, a analyzuje různé aspekty jazyka. Z obou kroků je tím složitějším, protože se musí potýkat s bohatou strukturou, nejasností a rozporuplností přirozených jazyků. Jeden vstup může mít více významů, a více vstupů může mít význam stejný [12].
 - *Lexikální nejednoznačnost* je problém se slovy která mají různou funkci ve větě podle kontextu. Například slovo „místo“ může být podstatné jméno, předložka nebo příslovce.
 - *Syntaxová nejednoznačnost* nastává v případě, že můžeme stejnou větu vyložit několika způsoby. Například věta „Rád jím zmrzlinu s dětmi.“ implikuje rodinnou pohodu nebo kanibalismus.
 - *Odkazná nejednoznačnost* vzniká při odkazování ve větě na podstatná jména pomocí zájmen a podobných neurčitých slov. Například z vět: „Petra a Jana se potkaly na ulici. „Ahoj!“ pozdravila.“ není jasné, o kom je řeč.
- **Generace (NLG)** je proces vytváření frází a vět v přirozeném jazyce které dávají smysl, na základě interních dat systému. Provádí plánování samotného obsahu, složení vět a strukturu celé zprávy [12].

Postup zpracování přirozené řeči - sestává obecně z pěti kroků [12].

1. *Lexikální analýza* - identifikace a analýza struktury slov. Lexikon je sbírka slov a frází v daném jazyce. Lexikální analýza rozkládá text na paragrafy, věty a jednotlivá slova.
2. *Syntaktická analýza* - nebo také větný rozbor je proces analýzy pozice jednotlivých slov ve větě a jejich skládání tak, aby dávaly smysl a měly vztah k okolním slovům.
3. *Semantická analýza* - přiděluje slovům jejich slovníkový význam a kontroluje smysluplnost. To se realizuje mapováním syntaktických struktur a objektů v doméně úkonů. Vyřadí slovní spojení jako je „suchá voda“ nebo „horký led“.
4. *Debatní integrace* - bere v potaz to, že význam věty je závislý na větě předchozí. Připravuje také informace o významu současné věty pro tu budoucí.
5. *Pragmatická analýza* - je proces při kterém se interpretuje skutečný význam toho, co bylo řečeno, neboli kontext.

2.2.4 Expertní systémy

Expertní systémy jsou jednou z předních oblastí výzkumu AI. Jde o počítačové aplikace vyvinuté k řešení problémů v jedné vybrané oblasti, a to na úrovni rovné nebo převyšující kvalifikovaného zkušeného člověka. Vyznačují se vysokým výkonem, srozumitelností, spolehlivostí a vysokou citlivostí [13].

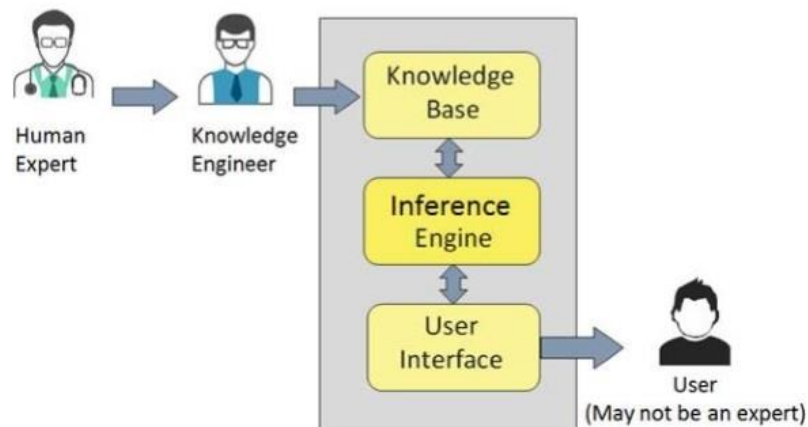
Expertní systémy **jsou** schopné:

- Radit, předávat instrukce, pomáhat uživateli v rozhodování
- Názorně předvádět data, vysvětlovat
- Interpretovat vstupní informace, diagnostikovat problémy
- Předvídat výsledky, odvozovat řešení
- Odůvodnit závěry, navrhnout alternativní řešení

Naopak **nejsou** schopné:

- Nahradit rozhodovací schopnost člověka
- Lidské způsobilosti
- Produkovat přesné výsledky z neadekvátní základny znalostí
- Samostatně zlepšovat vlastní znalosti

Obecně se expertní systémy skládají ze třech částí: znalostní základny, logického motoru a uživatelského rozhraní [13].



architektura expertních systémů [7]

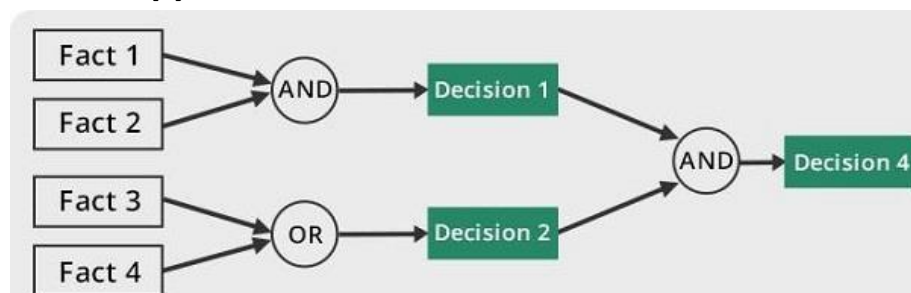
Znalostní základna musí obsahovat vysoce kvalitní znalosti v dané oblasti. Znalost je bezpodmínečně nutná k projevení inteligence, a úspěch každého expertního systému závisí na jeho sbírce vysoce přesných a kvalitních znalostí. Znalosti obsahují data, informace a předchozí zkušenosti. Základna obsahuje [13]:

- Faktuální znalosti - jsou považovány experty v oboru na univerzálně pravdivé
- Heuristické znalosti - vycházejí z praxe, zdravého rozumu, intuice a odhadu

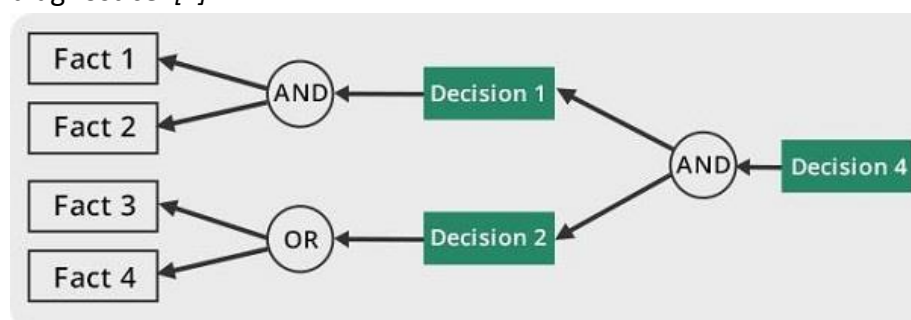
Data se reprezentují pravidly IF-THEN-ELSE (když-tak-jinak) [13]. Do znalostní základny je zadávají znalostní inženýři, osoby zvýšené empatie, učení a rychlého vyhodnocování [13]. Nahrávají, debatují a sledují odborníky v dané oblasti. Výsledná data kategorizují a organizují právě do pravidel která je pak expertní systém schopen využít [13].

Logický motor aplikuje znalosti ze znalostní základny na vstupní data aby došel k řešení. Používá dvě různé strategie [13]:

- *Dopředné řetězení (Forward chaining)* - sleduje od příčiny řetěz podmínek a kořenů dokud nedojde ke konečnému výsledku. Bere v potaz všechna fakta a pravidla systému. Používá se při hledání dopadů a k predikcím, jako například vliv změny úrokové sazby na trh s akciemi. [7]



- *Zpětné řetězení (Backward chaining)* - hledá příčiny ke známému důsledku. Ze známé situace sleduje zpětně sleduje řetěz podmínek, které k ní mohly vést. Používá se hojně například v lékařské diagnostice. [7]



Uživatelské rozhraní zprostředkovává komunikaci mezi uživatelem a expertním systémem. Využívá výše zmíněné přirozené jazyky (NLP) a umožňuje, aby uživatel nemusel být expert v poli AI. Má za úkol vysvětlit jakým způsobem došel systém k závěru, a snadno tak ověřit kredibilitu dedukcí.

3. Neuronové sítě

Umělá neuronová síť (dále ANN - Artificial Neural Network) je výpočetní model inspirovaný způsobem, kterým biologické nervové systémy, jako například mozek, zpracovávají informace. Klíčovým prvkem tohoto modelu je unikátní struktura systému zpracovávání informace. Skládá se z velkého množství hustě propojených procesních prvků (neuronů) které pracují jednotně k řešení konkrétních problémů. Neuronové sítě se, jako lidé, učí příkladem [14]. Neuronová síť se přizpůsobí konkrétním úkolům, jako je například rozpoznávání struktur nebo klasifikace dat, skrze vzdělávací proces. Učení v biologických systémech obnáší úpravy synaptických spojů, které se vyskytují mezi jednotlivými neurony. To platí i pro umělé neuronové sítě.

Neuronové sítě, se svou pozoruhodnou schopností najít smysl v komplikovaných, nepřesných a neúplných datech, se dají použít k hledání struktur, souvislostí a tendencí které jsou člověka nebo slabší výpočetní techniky příliš komplikované. Vycvičená neuronová síť se dá považovat za „experta“ v kategorii, jejíž data dostala k analýze [14]. Takový umělý odborník se dá využít k budoucím odhadům v nových situacích a k zodpovězení hypotézních otázek. K dalším výhodám patří[14]:

- **Adaptivní učení** - schopnost naučit se jak zvládat úkoly na základě daných dat nebo první zkušenosti
- **Samo-organizace** - neuronová síť je schopná sama organizovat a reprezentovat data která získává při učení
- **Provoz v reálném čase** - výpočty v neuronových sítích probíhají paralelně a vzniká speciální hardware který je schopný toho využít
- **Chybová tolerance skrze redundantní kódování** - částečná destrukce sítě vede k odpovídajícímu poklesu ve výkonosti, nicméně některé schopnosti systému zůstávají funkční i při závažném poškození sítě.

3.1 Neuronové sítě v porovnání s konvenčními počítači

Neuronové sítě přistupují k řešení problémů jinak než běžné počítače. Ty používají algoritmický postup - počítač sleduje sadu instrukcí k vyřešení problému. Pokud nemá k dispozici všechny potřebné instrukce, neumí problém vyřešit. To v praxi znamená že konvenční systémy řeší pouze problémy které už známe a řešit umíme.

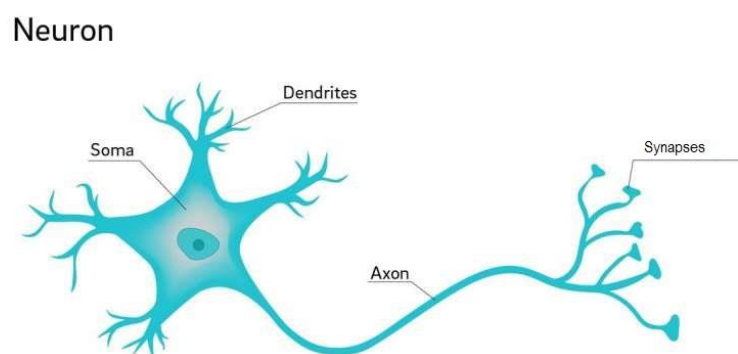
Neuronové sítě zpracovávají informace způsobem podobným lidskému mozku. Síť se skládá z velkého množství hustě propojených výpočetních prvků (neuronů) které pracují současně k řešení daného problému. Neuronové sítě se učí příkladem, a nedají se naprogramovat ke konkrétní činnosti [14].

Příklady ze kterých se učí musí být pečlivě vybrány, jinak systém jejich studiem ztrácí čas, nebo hůře, učí se dělat chyby [14]. Největší nevýhodou tohoto systému je, že síť se učí problémy řešit sama od sebe, a proto může být její funkce nepředvídatelná. To je kontrastem ke konvenčním počítačům které mají kognitivní přístup k řešení problémů - způsob řešení je popsán krátkými nezaměnitelnými instrukcemi [14]. Ty jsou následně přeloženy do vyšších programovacích jazyků, a konečně do strojového kódu kterému počítač rozumí. Každý krok cesty je předem daný a předvídatelný, a pokud dojde k nečekané chybě, víme že je o softwarová nebo hardwarová, a není to chyba uvažování stroje[14].

Neuronové sítě a tradiční počítače si vzájemně nekonkurují, ale doplňují se. Některé problémy, jako aritmetické operace jsou vhodné pro algoritmický přístup tradičních počítačů, a problémy vhodnější pro neuronové sítě. Zdaleka největší množství problémů vyžaduje systémy které využívají kombinaci obou technologií [15]. Zpravidla je řeší tradiční počítač spravující neuronovou síť kterou má dispozici [14].

3.2 Podobnosti mezi lidskými a umělými neurony

Stále není přesně známo jak se lidský mozek učí zpracovávat informace. Typický neuron přijímá signály ze struktur které nazýváme *dendrity*. Neuron vysílá záchvěvy elektrické aktivity dlouhým tenkým výběžkem jménem *axon* (také: *neurit*), který se na konci dělí do tisíců větví. Větvě jsou zakončeny *synapsami*, přísavkami které zajišťují spoj mezi jednotlivými neurony. Jestliže neuron přijme dostatečný impuls ze sousední synapse do svého dendritu, vyšle vlastní impuls po svém axonu do připojených synapsí a řetěz pokračuje. Spekuluje se, že k učení dochází když se zlepšuje účinnost přenosu těchto impulsů po neuronové síti[14].

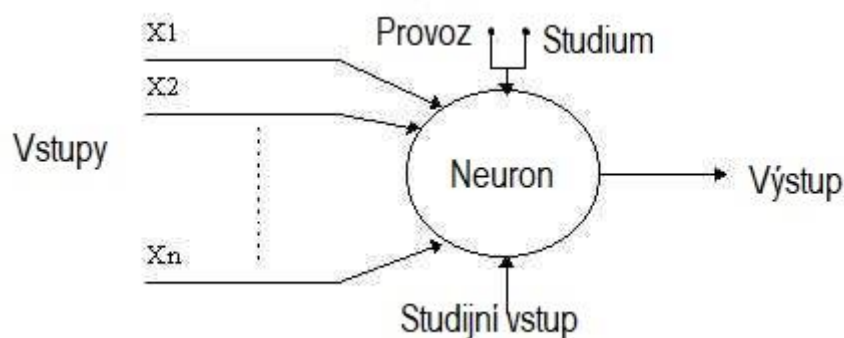


Neuron [16]

3.3 Návrh umělých neuronů

Sítě umělých neuronů se snažíme napodobit tak, že provádíme počítačové simulace vlastností skutečných neuronů a jejich spojení. Protože je ale naše současná znalost neuronů nekompletní, a naše výpočetní síla omezená, současné modely jsou pouze hrubé idealizace skutečných neuronových sítí [14].

Umělý neuron je zařízení s mnoha vstupy a jedním výstupem. Neuron má dva funkční stavy, studijní a provozní [14]. Ve studijní módu se neuron učí v reakci na různé tvary vstupů vysílat impulsy, a nebo naopak nevysílat nic. V provozním módu pak čeká, dokud na jeho vstup nedorazí naučená sekvence, a následně pošle dál naučenou odpověď. Pokud neuron vstupní sekvenci nepoznává, řídí se jeho chování zvláštními aktivačními pravidly, které mu definujeme pro tento případ[18].



Obr.9 náhradní schéma neuronu

MCP model neuronu

Výše popsaný model toho neumí víc, než konvenční počítač. K tomu potřebujeme sofistikovanější model MCP (McCulloch and Pitts) [18]. Ten přiřazuje jednotlivým vstupům (X_n) individuální váhy (W_n). Váhované vstupy se následně sčítají, a pokud součet překročí definovanou hranici (T), neuron vystřelí [18].

$$W_1X_1 + W_2X_2 + \dots + W_nX_n > T$$

Přidáním vstupních vah a aktivačního prahu významně roste pružnost a výkon neuronu. MCP neuron se přizpůsobuje situacím změnou svých vstupních vah a/nebo aktivačního prahu. Pro takové přizpůsobení existují algoritmy, jako například pravidlo Delta pro systémy s dopřednou vazbou, a backpropagation (zpětné šíření chyb) pro systémy se zpětnou vazbou[18].

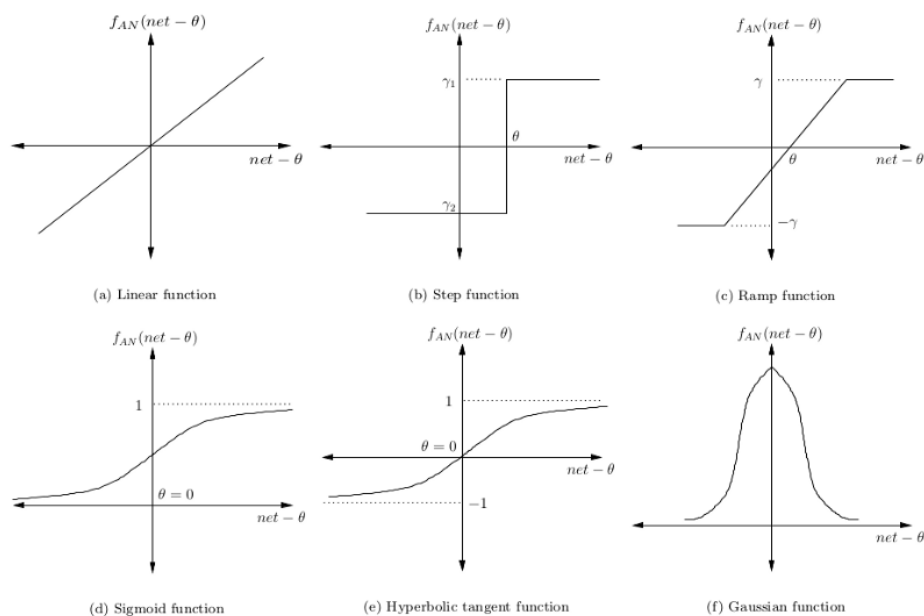
Přenosové funkce / Aktivační pravidla

Chování neuronové sítě závisí nejen na vahách, ale také na vstupně-výstupní (přenosové) funkci. Ty jsou typicky [18]:

- **lineární** - výstup je přímo úměrný celkovému váhovanému vstupu
- **prahové** - výstup má dvě možné úrovně, v závislosti na tom, jestli celkový váhovaný vstup přesahuje danou hranici, nebo ne.
- **sigmoidní** - má výstup který se mění, ale ne lineárně v závislosti na vstupu

Jsou důležitým prvkem neuronových sítí, a mají na svědomí jejich vysokou flexibilitu [18]. Aktivační pravidlo definuje výpočet toho, jestli se má neuron pro danou strukturu na vstupu aktivovat. Týká se všech možných struktur, nejen těch které se neuron naučil.

Jednoduchý příklad aktivačního pravidla funguje na bázi Hammingovi vzdálenosti. Předpokládejme že neuron zná sadu stavů které ho aktivují (1), a sadu stavů které ho naopak nechávají chladným (0). Jestliže obdrží stav který nezná, porovná jeho vstupy s nejbližšími stavy 0 a 1 které zná. Pakliže se podobá neznámý stav více stavu 1, tj. jeho Hammingova vzdálenost je kratší, chová se jako kdyby detekoval stav 1. Pokud je Hammingova vzdálenost k nejbližším stavům 0 a 1 stejná, zůstane stav jako nevyhodnocený.



Ukázky přenosových funkcí v ANN [18]

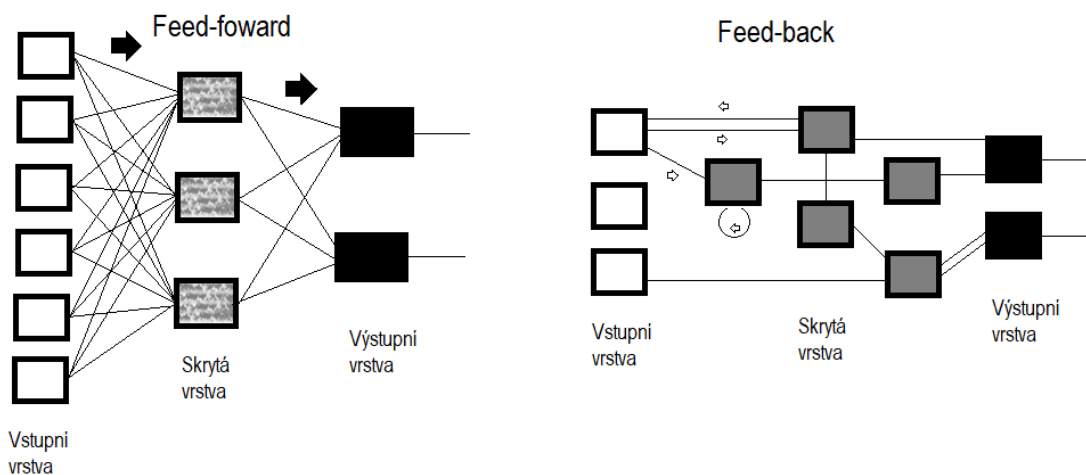
3.4 Architektury neuronových sítí

Nejběžnější model architektury neuronových sítí pracuje se třemi vrstvami neuronů [17]. **Vstupní** vrstva pracuje se surovými daty na vstupu sítě. **Skrytá a výstupní** vrstva berou v potaz data ze všech spojů, a *váhy* kterými jsou spoje zatíženy. Skryté vrstvy se mohou skládat z libovolného množství podvrstev. Takové sítě nazýváme Deep Neural Networks (DNN), neboli hluboké [7].

Struktury sítí můžeme rozdělit rámcově do dvou skupin [17]:

Feed-forward, neboli sítě s dopřednou vazbou, umožňují směr šíření signálu pouze jedním směrem. Neobsahuje žádné smyčky, a výstup dané vrstvy neovlivňuje sebe sama, ani žádné vrstvy předchozí. Obecně jde o rychlý model pro jednoduché mapování vstupů k výstupům. Používá se hojně k rozeznávání struktur (pattern recognition).

Feed-back, sítě se zpětnou vazbou, umožňují pohyb signálů oběma směry, zavedením smyček. Po přivedení vstupu se jejich vnitřní stavy dynamicky mění tak dlouho, dokud nedosáhnou rovnováhy. Říká se jim také interaktivní, nebo rekurentní (vratné). Jejich výpočetní potenciál je omezený skoro jen tím, jak komplikovaně jsou navrženy.



3.4.1 Příklady architektur

1. **Perceptron (P)** - nejjednodušší a nejstarší model neuronu. Sečte dvě vstupní hodnoty, aplikuje na ně přenosovou funkci a pošle na výstup [17].

Perceptron (P)



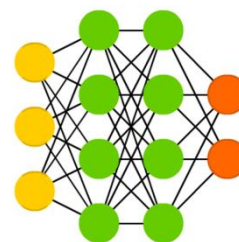
2. **Radial Basis Network (RBF)** - jsou shodné s klasickým modelem dopředné vazby, ale používají radiální přenosovou funkci, např. sigmoid. To jim umožňuje širší interpretaci dat než logické funkce. Mají jednu plně propojenou skrytou vrstvu [17].

Radial Basis Network (RBF)



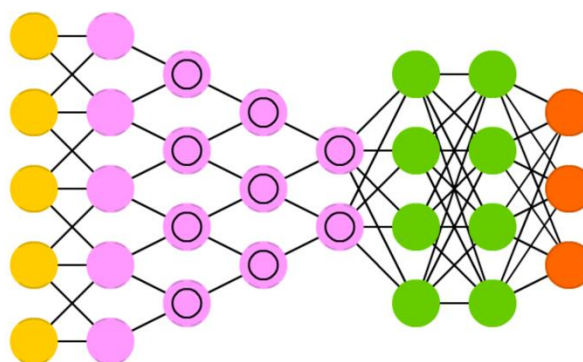
3. **Deep Feed Forward (DFF)** - jsou první generace sítí s „hlubokým“ neboli vícevrstevným učením. Obsahují dvě a více plně propojených skrytých vrstev. K jejich vývoji došlo teprve na zlomku milénia, kdy se vyřešil problém s kumulací chyb mezi vrstvami. Dnes tvoří základ pro většinu v praxi využívaných neuronových sítí [17].

Deep Feed Forward (DFF)



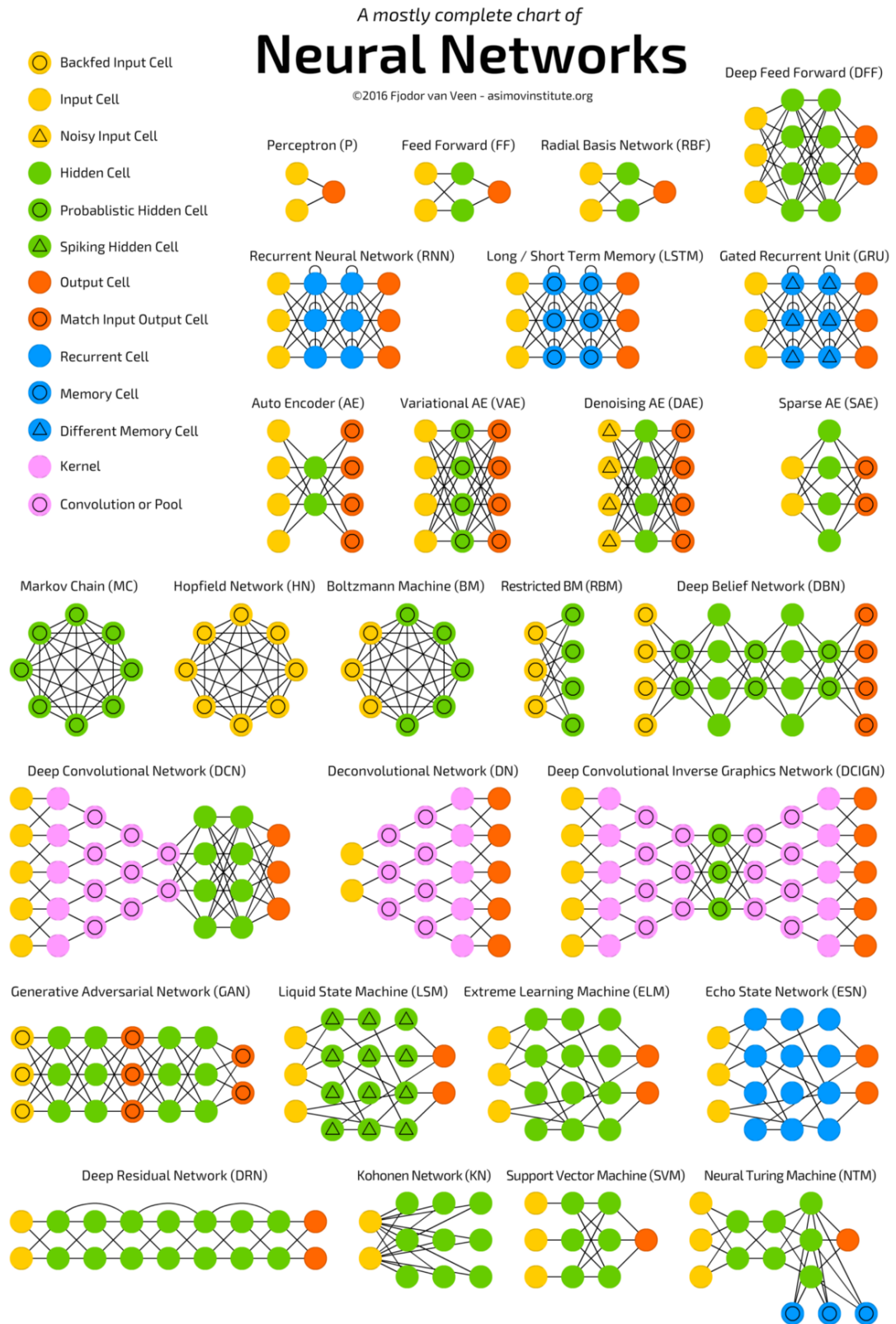
4. **Deep Convolutional Network** - hluboké konvoluční sítě dnes patří k těm nejvyužívanějším. Obsahují konvoluční jádra a buňky. Jádra zpracovávají vstupní informace a buňky je zjednodušují, zpravidla nelineárními funkcemi. Využívají se často ke zpracování obrazu. Operují s malým okénkem (např. 20x20 pixelů) obrazu, které se pousouvá, pixel po pixelu. Načtená data se předávají do konvolučních vrstev. První vrstva detekuje gradienty, druhá hrany, třetí tvary atd. Konec DCN sítě tvoří často struktura DFF, kde dochází k dalšímu zpracování dat [17].

Deep Convolutional Network (DCN)



[17]

Příklady mnoha dalších známých typů sítí jsou ilustrovány v diagramu na další stránce.



Ilustrace většiny známých neuronových sítí [17]

3.5 Strojové učení v neuronových sítích

Strojové učení je obor zkoumání algoritmů a statistických modelů které počítačové systémy využívají k tomu, aby samostatně plnily úkoly bez výslovných instrukcí, pomocí dedukce a rozpoznávání struktur [28]. Je podmnožinou umělé inteligence [28]. Algoritmy strojového učení budují matematické modely založené na vstupních datech, které nazýváme „trénovací“, na jejichž základě pak provádí odhady a rozhodnutí, aniž by je k tomu někdo vysloveně naprogramoval. Strojové učení je úzce spjato s poli optimalizace a statistiky [20].

Druhy algoritmů strojového učení se liší podle jejich přístupu, typu dat která přijímají a produkují, a charakteru úkolu nebo problému který mají řešit [28].

3.5.1 Učení pod dohledem

Algoritmy učení s dohledem vytváří matematický model ze sady dat, která obsahuje jak vstupy, tak požadované výstupy. Těmto datům říkáme „trénovací“, a skládají se z ukázek. Každá ukázka má jeden nebo více vstupů, a jeden požadovaný výstup, který nazýváme dozorčí signál. V matematickém modelu je každá ukázka jedním vektorem, a trénovací data jsou maticí. Opakovanou optimalizací této úlohy algoritmy učí síť neuronů matematickou funkci, která je schopná předpovídat výstup pro novou sadu vstupů [20]. Úspěšně optimalizovaná funkce je schopná odhadnou správně výstup pro vstupní data, která nebyla součástí trénovací sady. Algoritmus který časem zvyšuje přesnost odhadů svých výstupů považujeme za naučený. Systémy s dohledem mají zpravidla dva módy, učení a provoz, říkáme tedy, že se učí offline [28].

Podle možného tvaru výstupních dat se používají různé statistické algoritmy[28]:

- **Klasifikace** je proces „škatulkování“ výsledků do definovaných množin, např. při detekci objektů v obraze na „auto“ nebo „člověk“.
- **Regresní analýza** je sada statistických procesů pro zjišťování vztahů mezi závislými a nezávislými daty. Používá se pokud je výstupem číselná škála
- **Podobnostní analýza** využívá klasifikace i regrese. Jejím cílem je vyjádřit podobnost dvou objektů. Využívá se hojně pro vytváření žebříčků (ranking), ověření mluvčího, detekci objektů, rozpoznávání tváří a tvorby doporučení (marketing).

3.5.2 Učení bez dohledu (Hebbovo)

Algoritmy pro učení bez dohledu pracují s daty která obsahují pouze vstupy, a hledají v nich struktury, jako například seskupení a shluky datových bodů. Učí se tím pádem z dat která nejsou nijak popsána, klasifikována ani kategorizována. Místo toho aby Hebbovské systémy reagovaly na zpětnou vazbu, pozorují v trénovacích datech výše zmíněné anomálie, a poté nových neznámých datech reagují na jejich přítomnost nebo absenci. Používá se hlavně při odhadu hustoty pravděpodobností ve statistice. Hebbovské systémy se zpravidla učí za chodu, neboli online [28].

Ve strojovém učení mluvíme o **shlukové analýze**(cluster analysis). Algoritmus organizuje pozorování dat do clusterů tak, že si jsou v rámci jednoho clusteru podle daných kritérií podobná, a od ostatních clusterů se liší.

3.5.3 Zpětnovazebné učení

Učení posilováním se zabývá tím, jak se se softwaroví činitelé (agenti) chovají v daném prostředí, aby dosáhli největšího úspěchu. Jde o širokou oblast do které patří koncepty jako např. teorie her, teorie inteligence roje, teorie informace, a další. Ve strojovém učení mluvíme hlavně o Markovově rozhodovacím procesu (MDP) [28].

Markovův rozhodovací proces je stochastický model diskrétní v čase, vhodný pro popis takového rozhodování, kde výsledek operace je částečně náhodný, a částečně závislý na volbě agenta který ji provedl. Můžeme ho definovat vektorem [29]

$$(S, A, P_a, R_a), \text{ kde}$$

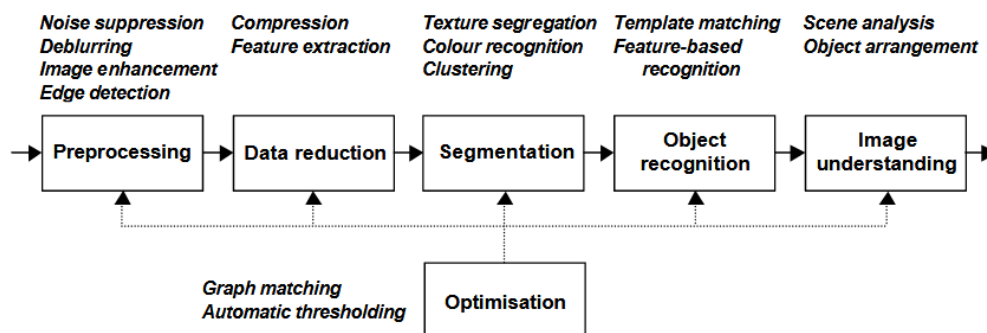
- S je konečná množina stavů
- A je konečná množina možných akcí
- P_a je pravděpodobnost, že v daném stavu \underline{s} a čase \underline{t} povede daná akce \underline{a} do stavu \underline{s}' v čase $\underline{t+1}$
- R_a je očekávaná odměna za přesun ze stavu \underline{s} do stavu \underline{s}' po provedení akce \underline{a}

Cílem algoritmu je naučit stroj správným rozhodnutím.

4. Zpracování obrazu v neuronových sítích

Jednou z nejdůležitějších aplikací neuronových sítí v dnešním světě je rozeznávání objektů v obraze. Image recognition (dále IR) se využívá k úkolům jako je popis obsahu obrázků, vyhledávání objektů, navádění robotů, autonomní vozidla, kamerový dozor a mnoho dalších. Rozeznávání v obraze je pro lidi a zvířata přirozenou schopností, ale extrémně náročným úkolem pro počítače. Stal se ale v posledních letech dosažitelným díky vývoji nových nástrojů a technologií [19]. Nejúčinnějšími nástroji k úspěšnému rozeznávání v obraze jsou hluboké neuronové sítě (DNN), konkrétně pak jejich konvoluční varianty (CNN) [19]. Konvoluční neuronové sítě jsou architektury navržené k efektivnímu zpracování, korelování a chápání velkého množství dat v obrazech s vysokým rozlišením.

Oči člověka vidí obraz jako sadu signálů které přijímá a interpretuje zraková kůra mozku. Výsledkem je „pocit“ ze scény, závislý na objektech které vidíme a našich přechozích zkušenostech s nimi. IR tento proces imituje. Systém vidí obraz jako sadu vektorů, nebo jako rastr diskretních numerických hodnot pro jednotlivé pixely [19]. Během rozeznávání převádí systém dané konfigurace hodnot na sady, které reprezentují fyzické objekty nebo rysy. Typický IR systém funguje podle následujícího schématu [19].



Blokové schéma rozeznávání v obraze [19]

1. **Předzpracování** - je sada úprav které je nutné provést na cestě mezi senzorem a dalším zpracováním. Můžeme je dělit na:
 - a. *Rekonstrukce obrazu* - je komplikovaný matematický proces vytváření 2-D obrazů z jednorozměrných dat, např. tomografie z rentgenu
 - b. *Restaurace obrazu* - je nejobsáhlejší způsob předzpracování. Obecně má za úkol odstranit chyby zavedené fyzickým snímacím systémem, jako například rozostření, rozmazání (motion blurr), zkreslení a další. Patří sem také filtrace šumu.

- c. *Vylepšení obrazu* - má za úkol zvýraznit prvky obrazu, například zvýšením kontrastu a manipulací spektrálního rozsahu (noční vidění).
2. **Redukce dat** - je proces, během kterého se obraz rozloží do jednoduchých vektorů nebo matic které obsahují stěžejní data.
 - a. *Komprese* - je zredukování obrazového signálu matematickou transformací, např. do spektrální oblasti. Nejčastěji se používá diskrétní kosinová a rychlá Furierova transformace (DCT, FFT), vektorová kvantizace (VQ) nebo DPC modulace [19]. Může být bezztrátová (lossless), nebo ztrátová - například po kvantování
 - b. *extrahování rysů (feature extraction)* - je reprezentace dat jejich nejvýznamnějšími rysy. Jestliže data obsahují redundantní informace (např. stejné pixely), můžeme jejich redukcí získat vektor rysů. S tím pak provádíme výpočty, místo celého obrazu. Patří sem detekce hran, hřbetů, rohů a pohybu [19].
 3. **Segmentace** - je rozdělení dat do shluků, ve kterých jsou si vzájemně podobná podle daných kritérií (např. barva, textura, intenzita). Segmentace je klasifikační úloha - jejím úkolem je pojmenovat jednotlivé pixely a rozdělit je do skupin [19].
 - a. *K-means* - je příkladem iterativní algoritmu používaného k segmentaci obrazu do K shluků [20]. Obraz rozdělíme, náhodně nebo nějakou heuristickou metodou, na K různých segmentů. Následně přiřazujeme jeden po druhém všechny pixely ke zvoleným segmentům, podle minimální vzdálenosti ke středu daného clusteru. Vzdáleností může být myšlen rozdíl barev, intenzity nebo textur. Následně se vypočítá průměr hodnot v každém clusteru a přesune se jejich střed. Opakuje se přiřazování pixelů ke klusterům. Celý proces se opakuje dokud nedojde ke konvergenci - stavu kdy přestávají pixely přeskakovat mezi clustery. Výsledkem je obraz segmentovaný do K úrovní.



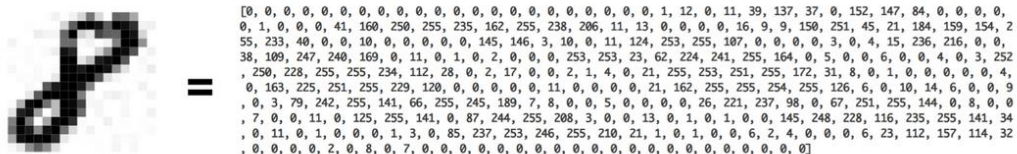
Obraz po provedení k-means segmentace ($k=16$) [20]

4. **Rozeznávání objektů** - je proces při kterém se data z přechozích kroků porovnájí s těmi, které má neuronová síť natrénované, a dochází k jejich klasifikaci.
5. **Chápání obrazu** - je samostatná složitá úloha umělé inteligence z obrazu interpretovat mínění.

4.1 Konvoluční neuronové síť

Pokud bychom chtěli detekovat objekty v obraze základní neuronovou sítí, narazili bychom na několik problémů.

Každý obraz si můžeme představit jako matici čísel která popisují jednotlivé pixely:



[21]

Pro detekci objektů, jako je číslovka 8, bychom potřebovali neuronovou síť s tolika vstupy, kolik má obrázek pixelů, tedy 324. Z předchozích kapitol víme, že by bylo možné takové síti dát k trénování velké množství obrázků daného objektu, a byla by pak schopná jeho rozpoznání. To bude ale fungovat jen a pouze pro obrázky podobné těm, se kterými jsme trénovali. Například, matice obrázku s číslovkou 8 veprostřed má úplně jiné hodnoty než matice obrázku kde je číslovka v rohu, a tudíž ji takový systém nezná, a nedetekuje. Stále v rámci tradičních neuronových sítí se tento problém dá částečně vyřešit buď tak, že síť natrénujeme na obrázky stejného objektu v jiných pozicích a zvýšíme počet vrstev neuronů (DNN), nebo budeme obraz analyzovat posuvným oknem a čekat, než narazíme na správnou pozici (brute force). Oba tyto přístupy jsou pomalé, nepraktické, neúplné a výpočetně náročné [21].

Řešením je **konvoluce**. Narozdíl od plně propojených neuronových sítí, v konvolučních sítích (CNN) jsou neurony uspořádané do trojrozměrných struktur které se zabývají každá jiným rysem objektu [21]. Pro příklad, při detekci člověka může jedna skupina neuronů poznat hlavu, jiná skupina nohy, další ruce. Protože se celá síť pohybuje po obraze malým konvolučním okénkem, analyzují se pixely pouze ve vztahu k pixelům okolním. To je narozdíl od běžných plně propojených sítí výpočetně dosažitelné.

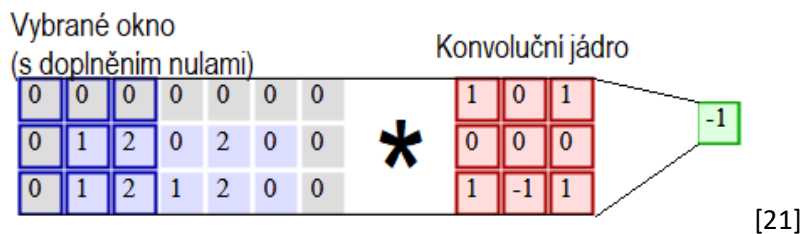
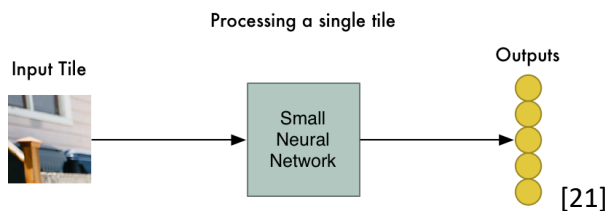
4.1.1 Funkce CNN krok po kroku

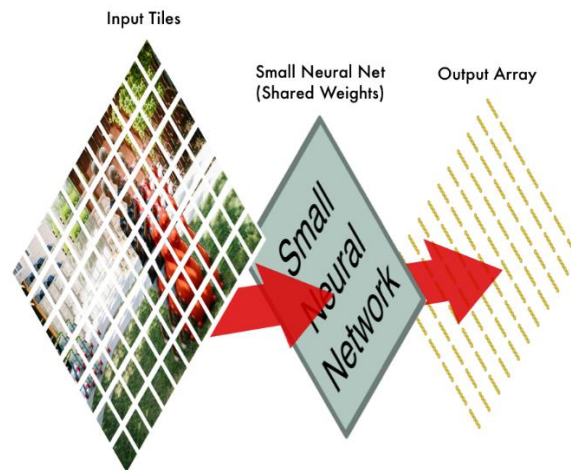
1. Obrázek je rozdělen do oken. Velikost oken je daná velikostí CNN která bude obraz analyzovat. Okna se překrývají, čímž roste spolehlivost výpočtů, ale také redundance



obrázek rozdělený do oken [21]

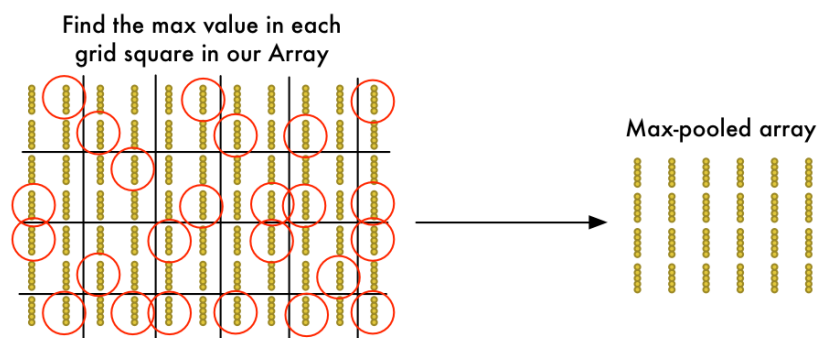
2. Okna prochází konvoluční neuronovou sítí. Ta se skládá z jednotlivých clusterů neuronů. Každý cluster je natrénován s vahami které odpovídají tomu, co má detekovat. Váhy takového clusteru tvoří konvoluční jádro. Neuronová síť provede **konvoluci** okénka s jádrem, jejíž výstupem je vektor pravděpodobností, že se v okénku nachází hledaný objekt.





[21]

3. Výsledná matice pravděpodobností stále obsahuje velké množství redundantních dat. Těch se částečně zbavíme downsamplingem. Nejčastější formou downsamplingu v CNN je tzv. „max pooling“. Max pooling jednoduše zachová nejvyšší hodnotu z daného výběru, a ostatní zahodí. Toho můžeme při detekci objektů využít, protože výsledky ukazující na detekci jsou ty největší, a tak zůstanou zachovány [21].

*max pooling* [21]

4. Kroky konvoluce a poolingů opakujeme dle požadavků a možností sítě. Například, při prvním cyklu může síť rozeznávat osoby, v druhém jejich končetiny, hlavu a torzo, a ve třetím detaily tváře. Konečným výstupem konvolučních clusterů je matice pravděpodobností (vah). Tu zpravidla použijeme jako vstupní data pro standardní plně propojenou hlubokou neuronovou síť, která rozhodne jestli se jedná o detekovaný objekt, nebo ne [21].

4.2 Facial recognition - rozpoznávání tváří

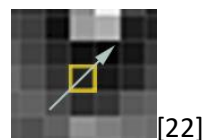
Rozpoznávání tváří je jednou z nejvyšších aplikací neuronových sítí v oblasti zpracování obrazu. Sestává z několika problémů:

- najít v obraze všechny tváře
- chápat, že pootočená tvář, nebo tvář ve stínu patří stále tomu samému člověku
- rozeznat v tváři unikátní rysy, jako velikost očí, délka nosu, tvar brady apod.
- porovnat detekovanou tvář s databází a správně jí přiřadit jméno

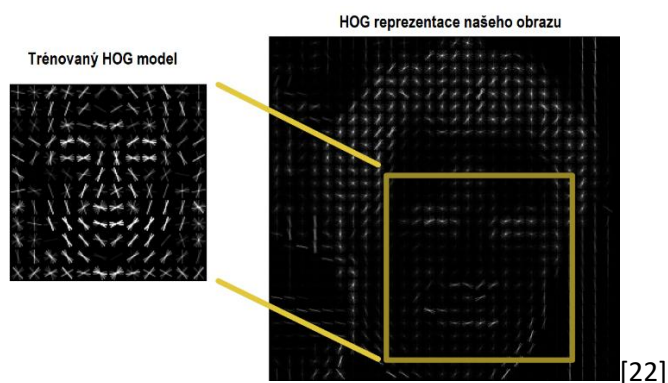
Lidský mozek řeší tyto problémy automaticky. Počítače musí problém řešit krok po kroku, a využívají k tomu řadu různých algoritmů strojového učení.

4.2.1 Detekce tváří

Nejpoužívanější metodou detekce tváří v obraze je tzv. „histogram orientovaných gradientů“ (dále HOG) [22]. V první řadě obraz převedeme na černobílý, protože barvu k rozeznání tváře nepotřebujeme. Následně každý jednotlivý pixel nahradíme šipkou která ukazuje směr, kterým se nachází nejtmaší sousední pixel. Těmto šipkám říkáme gradienty, a ukazují tok světla v obraze [22]. Mají oproti pixelům jednu zásadní výhodu - jsou identické pro světlé i tmavé obrázky stejné tváře.

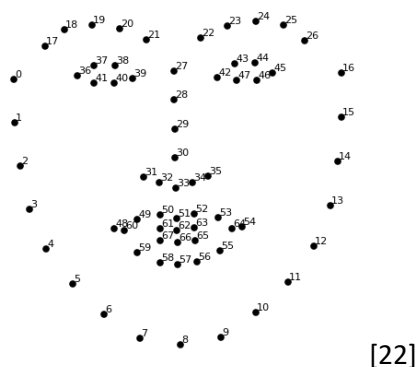


Protože k detekci nepotřebujeme zdaleka všechny gradienty, rozdělíme obraz do menších oken ve kterých vliv obsažených gradientů sečteme a zobrazíme. Získáme tím reprezentaci tváře která nese její nejdůležitější rysy. Takovou reprezentaci můžeme porovnat s HOG modelem který byl natrénován z velkého množství tváří, a detekovat tak tvář [22].



4.2.2 Promítání tváří

Po nalezení tváře musíme vyřešit problém s natočením. Tváře snímané z různých úhlů představují pro systém úplně novou sadu dat. Abychom to obešli, upravujeme obraz tak, aby byly oči a rty vždy na stejném místě v obraze. K tomu se využívá algoritmus **odhadu mezníků tváře (face landmark estimation - FLE)**. Ten funguje na principu hledání 68 specifických bodů (landmarks) které najdeme na každé tváři - špička brady, vnější okraj očí, koutky úst, apod [22]. K nalezení a přiřazení těchto bodů v libovolné tváři se používají natrénované neuronové sítě.



Když je tvář zmapovaná algoritmem FLE, víme kde jednotlivé rysy jsou, a můžeme obraz otáčet, zvětšovat, zmenšovat a ořezávat tak, aby byly oči a ústa co nejvíce ve středu. Snažíme se vyhnout metodám jako je prohýbání které by narušily paralelní linie.



Princip rouztahování tváře mapou mezníků [22]

S pomocí těchto postupů umíme libovolně natočenou tvář vycentrovat, a použít k dalšímu zpracování.

4.2.3 Kódování tváří

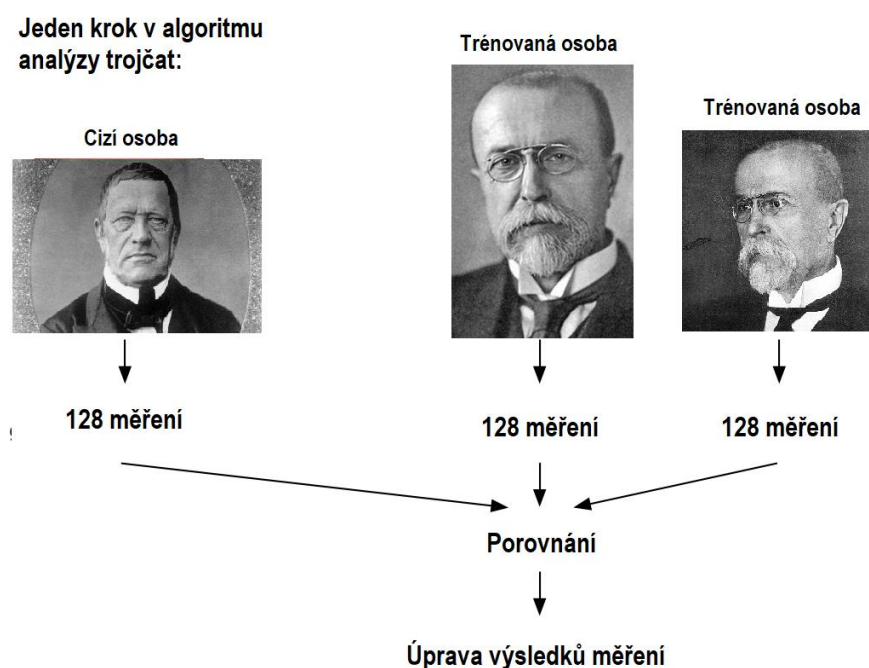
Teoreticky je rozeznávání tváří stejnou úlohou jako rozeznávání jakéhokoliv jiného objektu. Problém ale nastává, když chceme tvář jedince reprezentovat daty. Měření která nám dávají smysl, jako barva a velikost očí, tvar úst, nebo špičatost uší počítačovému systému nic neříkají, a jsou nepoužitelné. Výzkumem se zjistilo, že je nejlepší nechat síť samu rozhodnout, která data si

zapamatuje. Ke kódování tváří se používají výše popsané hluboké konvoluční sítě (DCN). Jedním z algoritmů pro trénování tváří je **metoda trojčat** [22].

V rámci metody trojčat poskytneme DCN síti tři rozdílné obrázky. Dva obsahují tvář kterou se snažíme naučit, třetí obsahuje tvář cizího člověka. Následně řekneme síti, aby vytvořila přesně 128 měření každé tváře. Algoritmus pak zkontroluje všechna měření a upraví je tak, aby měly obrázky stejné osoby výsledky které se blíží, a aby se zároveň vzdalovaly obrázku cizí osoby [22].

DCN algoritmus opakuje například tisíckrát s tisíci různými fotkami cizinců. Při každém průchodu se sada měření správné tváře stává odlišnější od cizích.

Výsledkem je sada 128 měření jednoho člověka. Těmto sadám se říká obtisky (embedding) a nikdo paradoxně neví, jaké měření obsahují [22]. Vybrala je sama síť a jsou tak unikátní, že je můžeme použít jako trénovací data pro jednoduché neuronové sítě, a tím pádem k rozeznání tváře klasifikačním algoritmem.



4.3 GAN - Generativní kompetitivní neuronová síť

GAN (Generative Adversarial Network) je zvláštní architektura hlubokých neuronových sítí která dokáže napodobit jakékoliv rozložení dat, tedy například obrázky, hudbu, mluvu, literaturu ale i umělecké malby. Skládají se ze dvou navzájem soupeřících sítí (proto kompetitivní) [23].

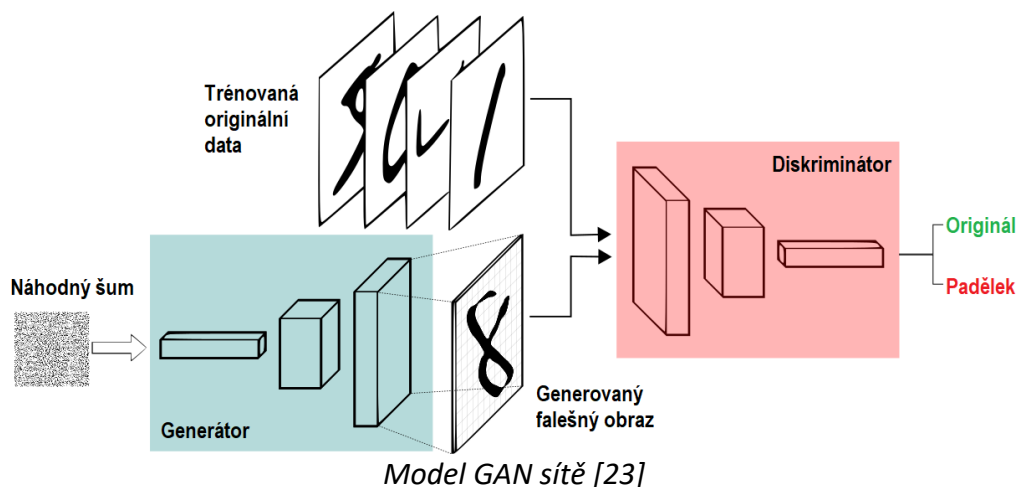
4.3.1 Generativní a diskriminativní algoritmy

S diskriminativními algoritmy jsme se setkali ve všech předchozích kapitolách. Jejich úkolem je na základě vstupních dat rozhodnout, do které kategorie patří. Příkladem takové systému je detekce spamu v emailové schránce. Jestliže pošta obsahuje určitá slova, roste pravděpodobnost že se jedná o spam. Matematicky to můžeme zapsat jako $p(y|x)$, neboli „pravděpodobnost (p) že pošta je spam (y) závisí na jeho obsahu (x)“. Diskriminativní síť tedy mapují obsah dat do kategorií a učí se hranice mezi nimi.

Generativní algoritmy fungují opačně. Pro dané kategorie hledají obsah. Použijeme-li předchozí příklad, pak matematicky je generativní síť definovaná jako $p(x|y)$, a slovně „Jestliže byla pošta spam (y), jaká je pravděpodobnost (p) že se v ní nacházel obsah (x)?“. Generativní algoritmy tedy předpovídají data na základě kategorií, a mapují jejich rozložení [23].

4.3.2 Funkce GAN sítí

Jak už bylo řečeno, GAN síť se skládají ze dvou soupeřících sítí, generátoru a diskriminátoru. Generátor vytváří (nejdřív náhodně) nová data, zatímco diskriminátor je hodnotí podle natrénovaného modelu. Cílem generátoru je vytvořit taková data, která diskriminátor vyhodnotí jako autentická (př. kopie obrazu). Diskriminátor má za úkol odhalit všechna data, která se odlišují od natrénovaného vzoru. Vzniká tedy dvojitá zpětná vazba - mezi generátorem a diskriminátorem, a mezi diskriminátorem a původním obrázkem [23].



5. Popis praktické části

Cílem praktické součásti této práce bylo demonstrovat některé koncepty umělé inteligence a neuronových sítí v oblasti zpracování obrazu. Sestavil jsem model autonomního kamerového modulu složeného z malého jednodeskového počítače Raspberry Pi, USB webkamery a USB zařízení k simulaci hlubokých neuronových sítí Intel® Movidius™ NCS. V prostředí Python jsem implementoval aplikaci pro bezpečnostní kameru na bázi detekce objektů v obraze, a experimentálně prověřil její schopnost fungovat za různých světelných podmínek.



5.1 Použitý hardware a software

Hardware:

Raspberry Pi 3 Model B 64-bit 1GB RAM - počítač

Trust SpotLight Webcam Pro - webkamera

Intel® Movidius™ Neural Computer Stick - výpočetní modul

Monitor HP 2309m

USB klávesnice Genius

USB myš Genius

Software:

Raspbian Stretch 9.9 - operační systém

NCSDK2 - vývojové prostředí k zařízení Movidius

- NC App Zoo - kolekce ukázkových aplikací a modelů pro Movidius

Python3 - programovací jazyk a prostředí

- knihovny: OpenCV3.3, TensorFlow, NumPy

5.1.1 Intel® Movidius™ Neural Computer Stick

Movidius je výpočetní USB zařízení od firmy Intel. Jeho účelem je poskytnout energeticky účinnou a cenově přijatelnou výpočetní podporu malých, často baterií napájených zařízení jako jsou drony a spotřebiče s IoT konektivitou.

Obsahuje vývojový kit (NCSDK) který je schopný rychlé tvorby prototypů, validace a nasazení hlubokých neuronových sítí (DNN). Profilování, ladění a kompilace DNN se provádí v SDK na vývojovém počítači, který by měl disponovat vlastní grafickou kartou. Natrénovanou a zkompilovanou hlubokou neuronovou sítí pak zvládá Movidius provozovat na všech kompatibilních koncových zařízeních.

Výpočetní sílu zařízení Movidius zajišťuje visuální procesor *Myriad 2 VPU*. Vlastně si můžeme Movidius představit jako grafickou kartu (GPU) na USB flash disku. Jeho primární funkcí je dodat slabým procesorům výpočetní sílu potřebnou k funkci náročných neuronových modelů jako GoogLeNet k plnění úkolů jako je rozeznávání objektů v obraze.

Movidius je navržen pro funkci v operačních systémech Ubuntu 16.04 a Raspbian Stretch. Podporuje dvě struktury DNN - TensorFlow a Caffé (ve které umí pracovat i s konvolučními sítěmi). Umožňuje realizaci složitých neuronových modelů jako je SqueezeNet, GoogLeNet a AlexNet na zařízeních s nízkou výpočetní silou.



Vývojový diagram Intel® Movidius™ NCS (zdroj: <https://software.intel.com/>)

5.2 Bezpečnostní kamera (fejtemic_cam)

Modul má za úkol nepřetržité monitorování prostoru. Sleduje osoby v zorném poli a v případě, že se jejich počet změní, uloží aktuální snímek do předem zvolené složky v síti, a vyšle do terminálu varovnou zprávu. K online detekci osob využívá zařízení Movidius, které obsahuje hlubokou neuronovou síť. Tu jsem „naučil“ detekci objektů veřejně dostupným předtrénovaným grafem MobileNet SSD* z nabídky Caffe Model Zoo. Program vznikl adaptací ukázkového programu k detekci objektů z veřejné nabídky NC App Zoo, a operuje na bázi struktury Caffe.

Modul může být připojen na wi-fi síť a napájen baterií. Lze ho tedy umístit kamkoliv, odkud bude posílat snímky a varovné zprávy do centrální jednotky, např. stolního počítače.

**Pozn.: MobileNet SSD (single shot detector) - je model detekce a rozeznávání objektů z jednoho snímku*

5.2.1 Slovní popis funkce programu

Modul je napájený, zapnutý, připojený k síti a obsahuje všechny podpůrné knihovny. Má stažený graf natrénovaných hodnot (MobileNet SSD *graph* file) nebo přístup k internetu aby si ho mohl stáhnout. Program je spuštěn v terminálu příkazem `make run_cam` v patřičném umístění.

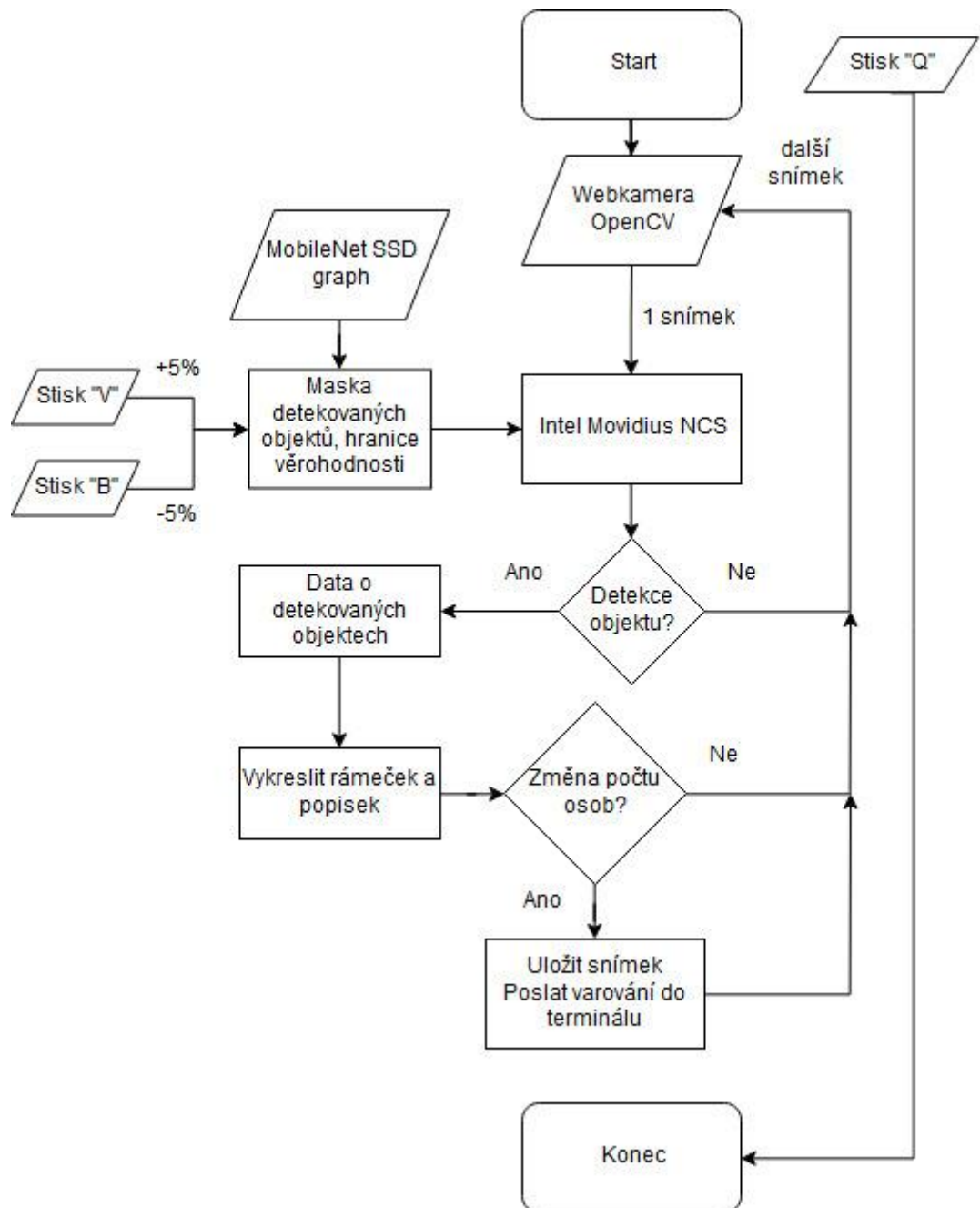
1. Program načte potřebné knihovny. Naváže kontakt se zařízením NCS a webkamerou.
2. Do NCS načte natrénovaný graf hodnot. Pomocí OpenCV otevírá okno s živým přenosem.
3. Spouští inferenční (vyvozovací) funkci. Jednotlivé snímky procházejí zařízením NCS. Pokud v nich detekuje objekt, vrací informace o věrohodnosti a pozici detekovaného objektu v obraze. Na základě těchto informací zobrazí kolem detekovaného objektu barevný rám. Pokud se ve snímku změnil počet detekovaných osob, pošle o tom do terminálu zprávu a dotýčný snímek uloží do složky. Inferenční funkce je nekonečná smyčka a běží, dokud ji uživatel neukončí.
4. Smyčka se dá ukončit klávesou „Q“. Po jejím stisku se vyčistí a zavře zařízení NCS, vypne se webkamera a zavře okno s přenosem.

Vedlejší funkce / features:

- Graf MobileNet SSD je natrénován k detekci dvaceti objektů, nejen osob, uspořádaných do matice. Které objekty bude modul detekovat se dá nastavit maskou.
- Výchozí věrohodnostní úroveň vyžadovaná k detekci je nastavena v programu na 60%. Za chodu je možné ji měnit stisknutím kláves ‚V‘(+5%) a ‚B‘(-5%).

```
labels = ('background',
         'aeroplane', 'bicycle', 'bird', 'boat',
         'bottle', 'bus', 'car', 'cat', 'chair',
         'cow', 'diningtable', 'dog', 'horse',
         'motorbike', 'person', 'pottedplant',
         'sheep', 'sofa', 'trash', 'tvmonitor')
object_classifications_mask = [0, 0, 0, 0, 0, 1, 1,
                              1, 1, 1, 0, 0, 1, 0,
                              0, 1, 0, 0, 0, 0, 0]
```

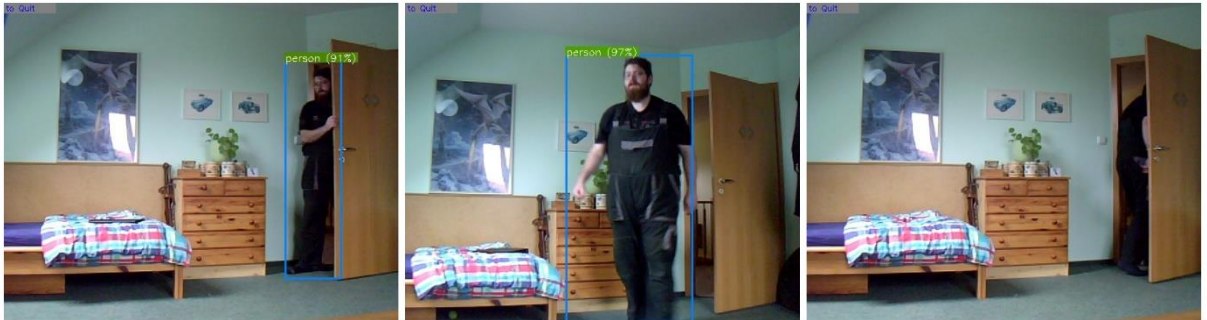
5.2.2 Vývojový diagram (flowchart)



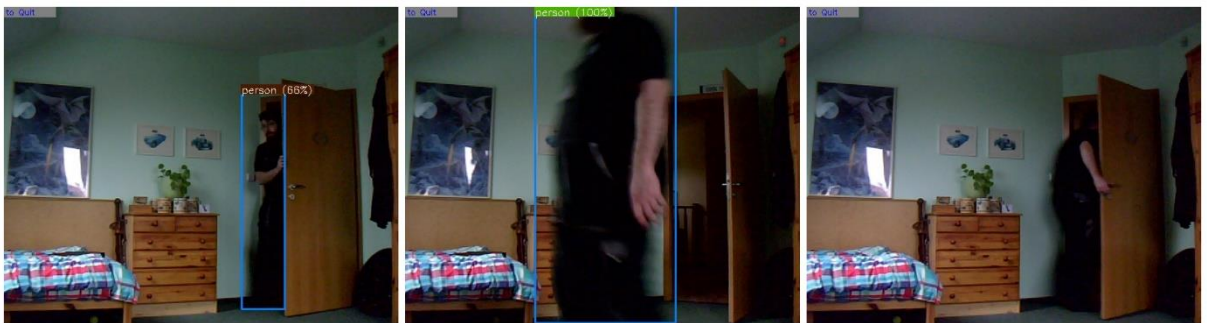
5.2.3 Testování funkce kamery za různých světelných podmínek

Cílem experimentu bylo zjistit, jaký má vliv osvětlení v místnosti na schopnost kamerového modulu vykonávat svou funkci. Pokusy jsem provedl pro různé úrovně denního a umělého světla. Sledoval jsem kdy sejme kamera první a poslední snímek, a kolik jich během pokusu vznikne. Experiment simuloval narušitele hlídané místnosti.

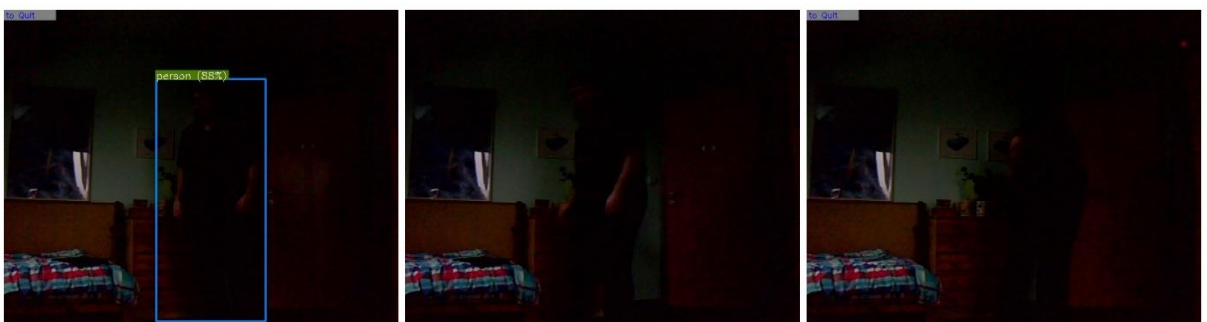
Denní světlo (14:00, zataženo) - 3 snímky



Odpolední šero (17:00, zataženo) - 3 snímky



Podvečer (18:30) - 13 snímků



Tma před západem slunce (20:30) - 0 snímků (úprava trasy)



Umělé světlo (max) - 4 snímky



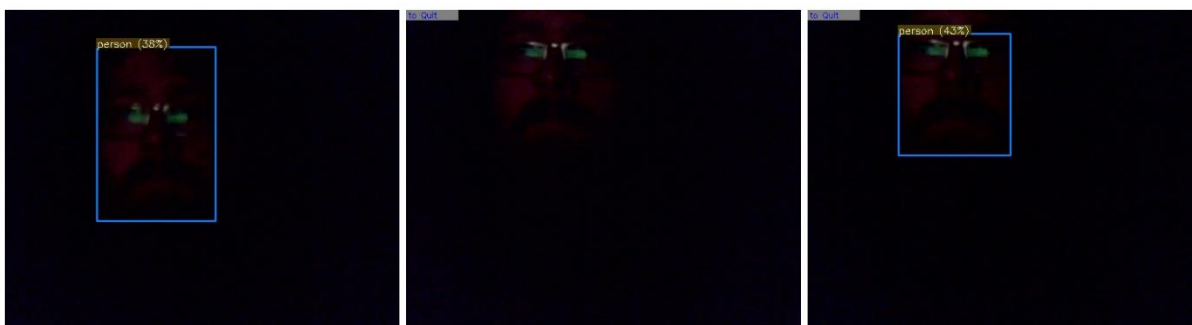
Umělé světlo (běžné osvětlení) - 5 snímků



Umělé světlo (šero) - 6 snímků



Téměř úplná tma (svit monitoru) - 0 snímků (úprava trasy+snížení hranice)



5.2.4 Vyhodnocení výsledků

Z výsledků experimentu jsou zřejmé dva nedostatky systému.

1. S horším osvětlením klesá schopnost kamery detekovat správně objekty. Toho si můžeme všimnout na poklesu věrohodností detekcí mezi jednotlivými pokusy, také prodlevy mezi vstupem do místnosti a prvním snímkem. Zároveň do jisté míry roste pravděpodobnost detekce objektu, který v zorném poli ve skutečnosti není (viz. obraz vyhodnocen jako kočka). V obou případech blížící se tmě už mne nebyl schopen systém při pokusu zaznamenat. Aby si mne všiml, musel jsem v případě západu slunce jít ke kameře blíže, a v případě osvětlení monitorem ještě navíc snížit vyžadovanou věrohodnost na velmi nízké hodnoty.

Problém je způsoben především nedostatečnou kvalitou kamery a omezených možnostech grafického zpracování počítače Raspberry Pi. Částečně by se dal problém vyřešit instalací infračerveného podsvícení, ale k dosažení zásadně lepších výsledků by bylo nutné vyměnit stěžejní komponenty.

2. Můžeme pozorovat, že s lepším osvětlením klesá počet uložených snímků. To je způsobeno implementací pravidla pro jejich ukládání. Připomínám, že snímky se ukládají pokud dojde ke změně počtu osob v zorném poli. V ideálních podmínkách (tzn. při dostatečném osvětlení) je člověk sledován od chvíle, kdy vstoupí do místnosti, do chvíle, kdy ji opustí. Výsledkem jsou pouze dva snímky. Naproti tomu ve zhoršených podmínkách dochází během cesty místností k výpadkům detekce, což má za následek mnohem víc snímků narušitele a vede paradoxně ke zlepšení funkce.

Tento problém se dá snadno odstranit zavedením proměnné, která bude hlídat počet po sobě jdoucích snímků, které obsahovaly stabilní počet lidí. Po dosažení předem definované hodnoty na takovém čítači (např. 5 snímků při 5fps = 1 vteřina) by se následující snímek uložil.

5.3 Demonstrace rozeznávání tváří (face recognition)

Nezávisle na aplikaci s bezpečnostní kamerou jsem se pokusil demonstrovat princip vytváření grafů pro neuronové sítě z trénovacích dat. Využil jsem k tomu program „*video_face_matcher_multipleFace*“ z nabídky vzorových aplikací NC App Zoo. Ten využívá modelu struktury TensorFlow k učení neuronové sítě FaceNet z poskytnutých trénovacích dat (fotek obličejů). Přístup FaceNet sítě k rozeznání obličejů stojí na bázi mezníků tváře (face landmarks). Program sleduje přenos z webkamery v reálném čase a rozhoduje, jestli je osoba v obraze podobá některé z trénovacích fotek na základě parametru „total_difference“. Výsledek zobrazí barevným rámečkem a výpisem jednotlivých věrohodnostních kritérií, kterými jsou rozdíly mezi mezníky.

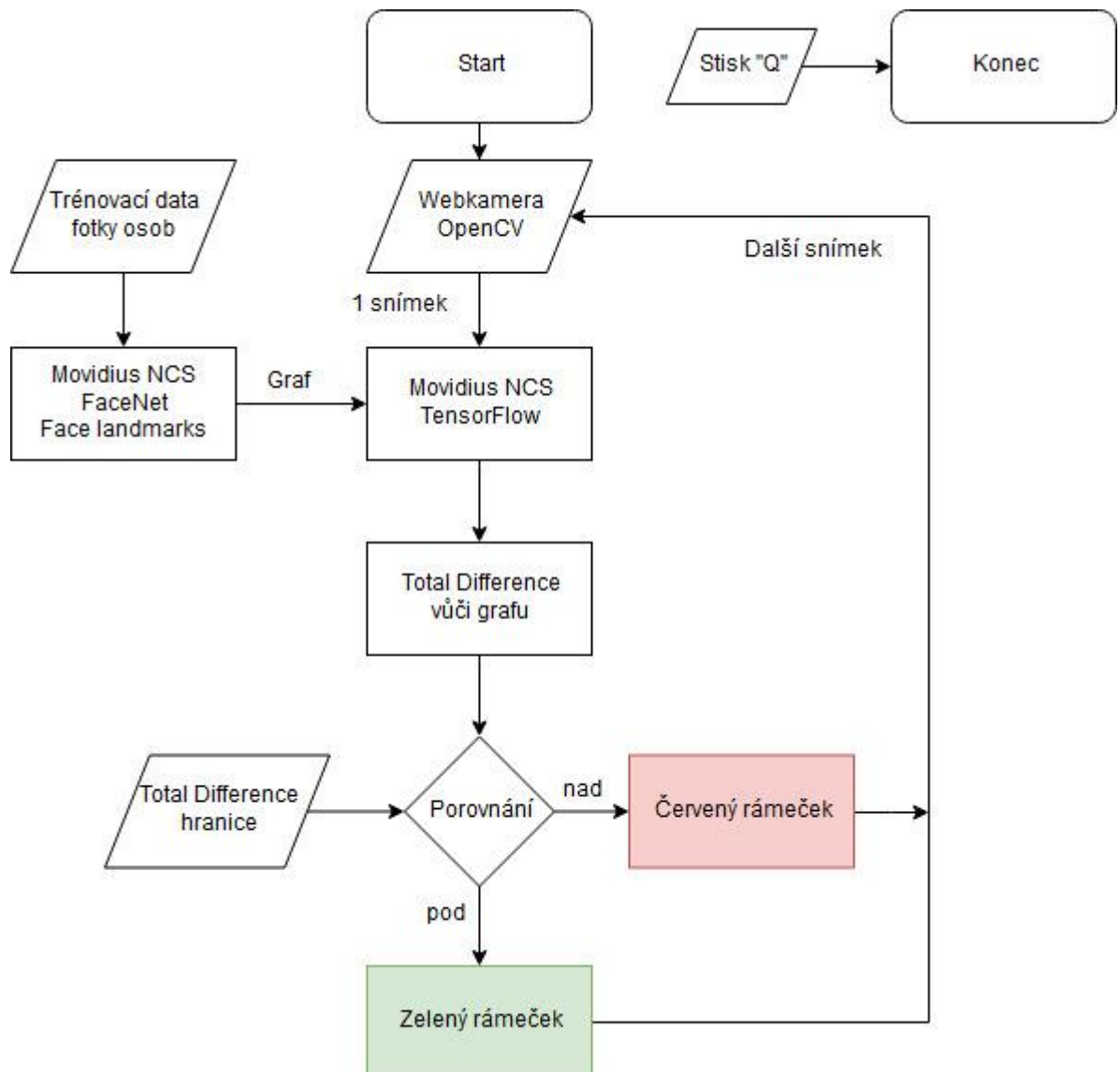
5.3.1 Slovní popis funkce programu

1. Zařízení Movidius simuluje neuronovou síť TensorFlow/FaceNet. Do sítě se postupně vkládají všechny trénovací fotky, ve kterých se hledají a vyčíslují mezníky tváře. Výsledek učení je uložen jako tzv. graf.
2. Je spuštěn hlavní program k rozeznávání tváře. Navazuje kontakt se zařízením Movidius NCS a webkamerou. Do NCS je nahrán graf získaný z předchozího kroku, a OpenCV otevírá okno s přímým přenosem webkamery.
3. NCS porovnává inferenční funkci každý snímek zorného pole kamery s nahraným grafem, a vypisuje do terminálu jak moc se snímek liší od natrénovaných dat. Rozdíl jednotlivých mezníků tváře je sečten do parametru „total_difference“, a nastavením prahu pro tuto hodnotu rozhodujeme přesnost rozeznávání.
4. Pokud je v zorném poli jedna z osob, pro které je natrénován graf, klesne parametr „total_difference“ pod hranici pro jeden nebo více trénovacích obrázků. To indikuje program zeleným rámečkem a výpisem jména fotky, které se člověk v zorném poli nejvíc podobá. V opačném případě zobrazuje rámeček červený. Program funguje ve smyčce.
5. Smyčka se dá ukončit klávesou „Q“. Po jejím stisku se vyčistí a zavře zařízení NCS, vypne se webkamera a zavře okno s přenosem.

```
Total Difference is: 1.23909956217
Total Difference is: 0.608490228653
Total Difference is: 0.626376509666
Total Difference is: 0.972404301167
Total Difference is: 0.834983885288
Total Difference is: 0.589920222759
PASS! File camera frame 460 matches 60870629_2352893331656542_2065702544
88_n.jpg
```

Ukázka výpisu hodnot Total Difference

5.3.2 Vývojový diagram



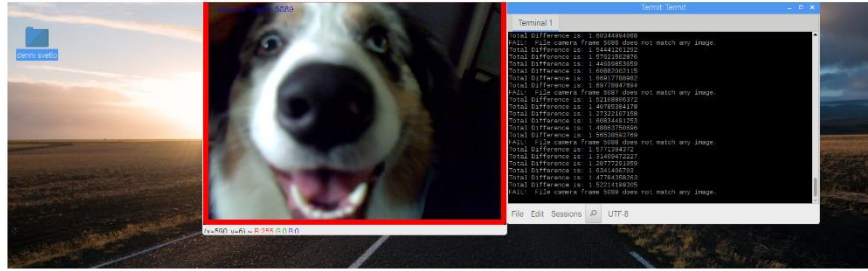
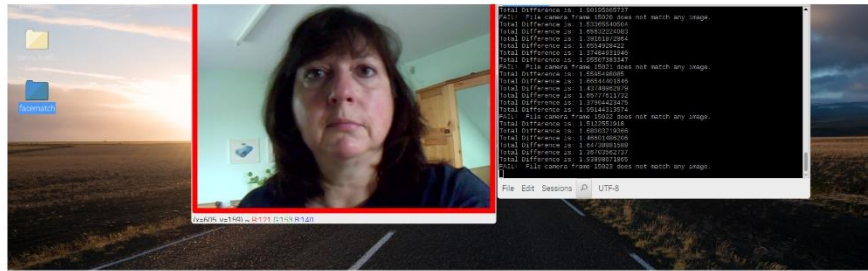
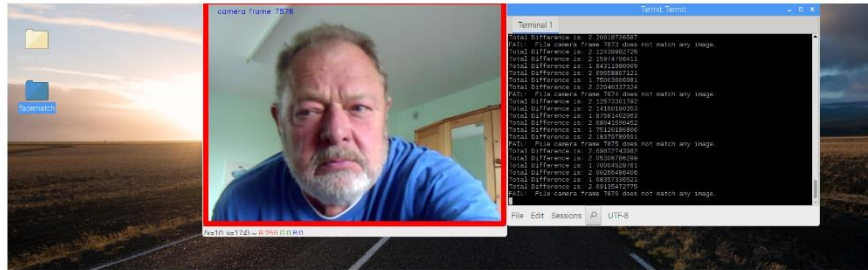
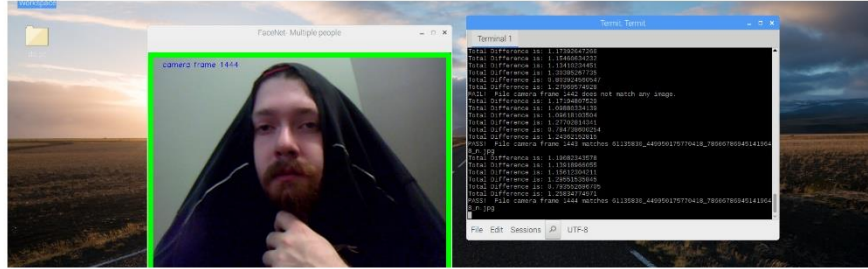
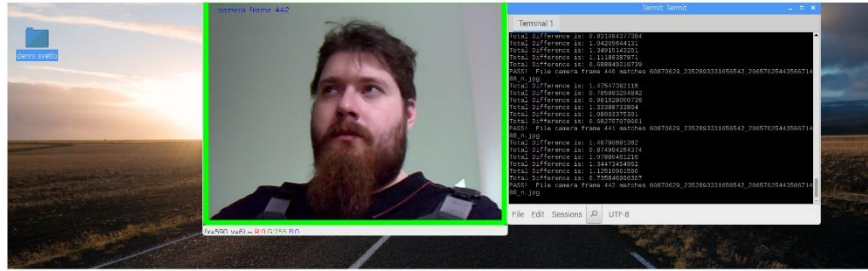
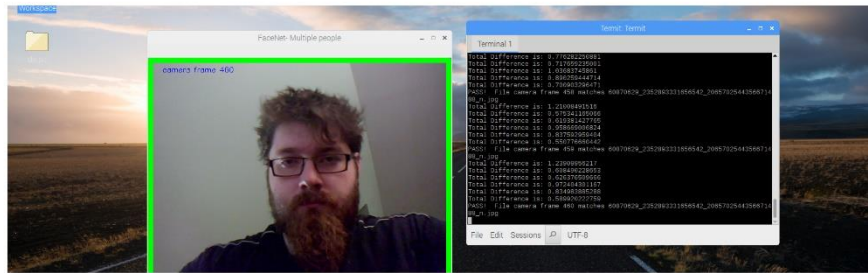
5.3.3 Testování funkce

Pro testování funkce a demonstrace principu rozeznávání jsem natrénoval FaceNet model svými fotkami, a vytvořil tak graf pro rozeznání pouze mé vlastní tváře. Při testování jsem před kameru postavil sebe a své bližní a ověřoval, zda model funguje podle předpokladů.

Použitá trénovací data:



Výsledky:



Vyhodnocení:

Rozeznávání tváří funguje podle očekávání. Program je schopen bez problémů rozeznat osobu, pro kterou byl v modelu FaceNet vytvořen graf trénovaných hodnot mezníků tváře. Na druhé a třetí fotce jsem záměrně algoritmu ztížil práci, ale i tak mne bez problémů poznal. Naproti tomu ostatní testované tváře spolehlivě vyřadil. Při experimentu byla mezní hranice parametru Total Difference nastavena na 0.8. Vyšší hodnoty způsobí, že může program mylně rozeznat špatné osoby. Nižší hodnoty naopak snižují toleranci, a může se stát, že program nerozezná ani osobu trénovanou.

Závěr

V rámci této diplomové práce jsem mě za úkol seznámit se s metodami zpracování obrazu využitelných v bezpečnostních kamerových systémech. Protože všechny tyto metody využívají technologie neuronových sítí, umělé inteligence a strojového učení, pojal jsem tuto práci jako úvod do tematiky. Ačkoliv by pochopení každého odvětví umělé inteligence vyžadovalo desítky tisíc stran publikací samo pro sebe, myslím, že se mi v této práci podařilo vytvořit stručný přehled oboru.

V praktické části jsem měl za úkol implementovat a otestovat autonomní kamerový modul využívající výše zmíněných konceptů na bázi počítače Raspberry Pi. Tento úkol jsem splnil implementací kamerového modulu schopného monitorování prostoru, detekce osob které se v něm pohybují, a ukládání jejich fotek do místa v síti. Zároveň jsem provedl experiment k zhodnocení detekčních schopností modulu v závislosti na osvětlení prostoru. Nakonec jsem použil modul k trénování vlastních fotek a tvorbě grafu, který byl dále součástí demonstrace principu rozeznání tváře (facial recognition).

Seznam použitých zdrojů a literatury

- [1] M. Tim Jones. (2017, June 1). *A beginner's guide to artificial intelligence, machine learning, and cognitive computing*. <https://developer.ibm.com> [online]
- [2] Wikipedia contributors. (2019, April 14). *History of artificial intelligence*. Wikipedia, The Free Encyclopedia. [online]
- [3] Reuben Hoggett. (2009, November 17). *1951 – SNARC Maze Solver – Minsky / Edmonds (American)*. <http://cyberneticzoo.com/> [online]
- [4] Waseda University Humanoid Robotics Institute. *History of Humanoid Robot in Waseda University*. <http://www.humanoid.waseda.ac.jp> [online]
- [5] Rockwell Anyoha. (2017, August 27). *The History of Artificial Intelligence*. Harvard University. <http://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/> [online]
- [6] Paul Marsden (2017, August 21). *Artificial Intelligence Timeline Infographic – From Eliza to Tay and beyond*. Digital Wellbeing. <https://digitalwellbeing.org/> [online]
- [7] Artificial Intelligence Tutorial. (2018). <https://www.tutorialspoint.com/> [online]
- [8] Chethan Kumar GN. (2018, August 31). *Artificial Intelligence: Definition, Types, Examples, Technologies*. Medium. <https://medium.com/> [online]
- [9] Tannya D. Jajal. (2018, May 21). *Distinguishing between Narrow AI, General AI and Super AI*. Medium. <https://medium.com/> [online]
- [10] Prashant Gupta. (2018, January 10). *Search Algorithms in Artificial Intelligence*. <https://hackernoon.com/> [online]
- [11] Surya Priy, Abhishek rajput (2018). *Fuzzy Logic | Introduction*. Geeks for geeks. <https://www.geeksforgeeks.org/> [online]
- [12] Dr. Michael J. Garbade. (2018, October 15). *A Simple Introduction to Natural Language Processing*. *Becoming Human: AI Magazine*. <https://becominghuman.ai/> [online]
- [13] Rinu Gour. (2018, November 15). *What is Expert System in Artificial Intelligence - How it Solves Problems*. Medium. <https://medium.com/> [online]
- [14] Christos Stergiou. Dimitrios Siganos. (1996). *Neural Networks*. Imperial College of London. <https://www.doc.ic.ac.uk> [online]
- [15] Michael Peragine. (2013, March 13). *The Universal Mind: The Evolution of Machine Intelligence and Human Psychology*. Xiphias Press. San Diego, CA.
- [16] <https://medicalxpress.com/>, obrázek neuronu [obrázek][online]

[17] Fjodor van Veen. (2017, March 31). A mostly complete chart of neural networks. The Asimov Institute. <http://www.asimovinstitute.org> [obrázek][online]

[18] Mukul Malik. (2018, April 3). Basics of Neural Network. Becoming Human. <https://becominghuman.ai/> [obrázek][online]

[19] M. Egmont-Petersen. D. de Ridder. H. Handels. (2001, May 12). Image processing with neural networks - a review. Pattern Recognition: The Journal of Pattern Recognition Society. Utrecht University, Netherlands. [obrázek]

[20] Wikipedia contributors. (2019, May 6). Machine Learning. Wikipedia, The Free Encyclopedia. [online][obrázek]

[21] Adam Geitgey. (2016, June 13). Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Medium online magazine. <https://medium.com/> [online][obrázek]

[22] Adam Geitgey. (2016, June 24). Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning. Medium online magazine. <https://medium.com/> [online][obrázek]

[23] Unspecified author. (2019). A Beginner's Guide to Generative Adversarial Networks (GANs). Skymind Inc.. <https://skymind.ai/> [online][obrázek]

[24] Robert Mullins. (2012). Introduction - What is a Turing machine?. University of Cambridge. <https://www.cl.cam.ac.uk>

[25] John McCarthy. (2007). What is artificial intelligence?. Article. Stanford University.

[26] John Loeffler. (2019, February 3). Should we fear artificial superintelligence?. Interesting engineering online magazine article. <https://interestingengineering.com> [online]

[27] Wikipedia contributors. (2019). Superintelligence. Wikipedia, The Free Encyclopedia. [online]

[28] Shagufta Tahsildar. (2018). Machine Learning Basics. QuantInsti company. <https://www.quantinsti.com/> [online]

[29] Pascal Wallisch, Nicholas G. Hatsopoulos, in [MATLAB for Neuroscientists \(Second Edition\)](#), chapter on Hidden Markov Models 2014