

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

KATEGORIZÁCIA OBRAZU V KLOUD ROBOTIKE
Diplomová práca

2014

Jaroslav Ondo, Bc.

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

KATEGORIZÁCIA OBRAZU V KLOUD ROBOTIKE
Diplomová práca

Študijný program: Umelá inteligencia
Študijný odbor: 9.2.8. Umelá inteligencia
Školiace pracovisko: Katedra kybernetiky a umelej inteligencie (KKUI)
Školiteľ: Prof. Ing. Peter Sinčák, CSc.
Konzultant:

2014 Košice

Jaroslav Ondo, Bc.

Abstrakt

Predkladaná diplomová práca sa zaoberá teóriou cloud robotiky a hľadáním riešenia problému rozpoznávania a klasifikácie objektov zo statických obrazov. Celé spracovanie sa vykonáva v prostredí cloudu. Na dosiahnutie uvedeného cieľa sú využívané neurónové siete typu MF ARTMAP tréňované pomocou obrazových deskriptorov a pomocou farebného spektra obrazov. Deskriptory obrazu extrahujeme zo vstupných statických obrazov pomocou algoritmov počítačového videnia, konkrétne pomocou algoritmov SIFT a SURF. Presnosť uvedených metód tréňovania neurónovej siete navzájom porovnávame. V práci venujeme značnú pozornosť cloudu, cloudovým službám a prepojeniu cloudu s robotikou. Pomocou robota Nao popisujeme spôsob pripojenia robota priamo ku cloudovej službe a tiež spôsob komunikácie medzi robotom a cloudovou službou. Demonstrujeme možnosť použitia robota Nao ako zdroj vizuálneho vstupu pre potreby klasifikácie objektov z obrazu. Hlavnou platformou, ktorú využívame je cloudová platforma Microsoft Windows Azure.

Kľúčové slova

cloud, robotika, neurónové siete, MF Artmap, počítačové videnie, rozpoznávanie objektov, SIFT, SURF

Abstract

This diploma thesis deals with theory of cloud robotics and finding the solution to the problem of objects recognition and classification from the statical images. Entire processing is executed in the cloud environment. MF ARTMAP neural networks trained by image's descriptors and colour spectrum are used to reach this aim. The precision of the classification reached by using these training methods are compared. The image's descriptors are extracted from input statical images using computer vision algorithms, namely SIFT and SURF algorithms. In this work we pay attention to cloud, cloud services and linking cloud and robotics. By using the Nao robot, we describe the way for connection robot into cloud service and also the way for communication between robot and cloud service. We demonstrate the possibility of usage Nao robot like a source of visual input for needs of objects classification. Main platform which we use, is Microsoft Windows Azure platform.

Key words

cloud, robotics, neural networks, MF Artmap, computer vision, object recognition, SIFT, SURF

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **9.2.8 Umelá inteligencia**

Študijný program: **Umelá inteligencia**

Názov práce:

Kategorizácia obrazu v kloud robotike

Image Categorization in Cloud Robotics

Študent:

Bc. Jaroslav Ondo

Školiteľ:

prof. Ing. Peter Sinčák, CSc.

Školiace pracovisko:

Katedra kybernetiky a umelej inteligencie

Konzultant práce:

Pracovisko konzultanta:

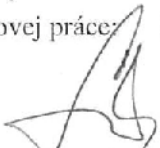
Pokyny na vypracovanie diplomovej práce:

1. Vypracovať prehľad v oblasti Kloudovej robotiky a jeho aplikačnom potenciály
2. Implementovať na AZURE platforme neuronovú sieť MF ARTMAP pre spracovanie obrazu ako službu
3. Vytvoriť prostredie interakcie človek počítač pre platformu Windows 8 a system NAO
4. Realizovať experimenty na naprogramovanej neurónvej sieti na testovacích obrazoch
5. Vytvoriť metodickú užívateľskú , systémovú príručku a ďalšiu dokumentáciu podľa pokynov vedúceho práce

Jazyk, v ktorom sa práca vypracuje: slovenský


Termín pre odovzdanie práce: 02.05.2014

Dátum zadania diplomovej práce: 31.10.2013


prof. Ing. Peter Sinčák, CSc.

vedúci garantujúceho pracoviska




prof. Ing. Liberios Vokorokoš, PhD.

dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 2. mája 2014

.....

podpis

PodĎakovanie

V úvode tejto práce by som rád vyjadril svoje podĎakovanie vedúcemu diplomovej práce prof. Ing. Petrovi Sinčákovi, CSc. za jeho odborné rady, postrehy, usmernenia, trpezlivosť a vedenie počas celého obdobia prípravy práce. Špeciálne chcem podĎakovať Ing. Danielovi Lorenčikovi a Ing. Tomášovi Cádrikovi za konzultácie v oblasti cloud technológií a usmernenia pri programovaní služieb pre platformu Microsoft Windows Azure.

Obsah

Zoznam obrázkov	9
Zoznam tabuliek	10
Zoznam symbolov a skratiek	11
Slovník termínov	12
Úvod	13
1. Formulácia úlohy a cieľ práce	14
2. Cloud computing a cloud robotika	15
2.1. Definícia cloudu a cloudových služieb	15
2.1.1. Servisné modely cloudu	16
2.1.2. Výhody cloudu	18
2.1.3. Poskytovatelia cloud služieb.....	19
2.2. Spojenie robotiky a cloud služieb	20
2.2.1. Definícia cloud robotiky	21
2.2.2. Porovnanie siete robotov a cloud robotiky	21
2.2.3. Výhody a aplikačný potenciál cloud robotiky	23
2.2.4. Existujúce projekty cloud robotiky.....	23
2.2.5. Zhrnutie cloud technológií	26
3. Rozpoznávanie objektov vo vizuálnom obraze	27
3.1. Spracovanie vizuálnych obrazov.....	27
3.2. Proces rozpoznávania objektov.....	28
3.3. Práca s príznakmi.....	30
3.3.1. Algoritmus SIFT	31
3.3.2. Algoritmus SURF	35
3.4. Klasifikácia a kategorizácia.....	39
3.4.1. Klasifikácia verus kategorizácia	39
3.4.2. Metódy klasifikácie objektov.....	39
3.4.3. Neurónové siete	40

3.4.4.	Neurónové siete typu ART.....	41
3.4.5.	MF ARTMAP.....	42
4.	Návrh systému pre rozpoznávanie objektov z obrazu v cloud robotike	47
4.1.	Požiadavky na systém rozpoznávania a jeho teoretický návrh	47
4.1.1.	Výber klasifikačnej metódy	47
4.1.2.	Výber cloud platformy na vývoj a implementáciu.....	47
4.1.3.	Výber frameworku a programovacieho jazyka.....	48
4.1.4.	Rozhrania cloud služby.....	48
4.2.	Implementácia navrhnutého systému.....	50
4.2.1.	Príprava reprezentatívnej množiny obrazov	50
4.2.2.	Predspracovanie a segmentácia obrazov reprezentatívnych množín	51
4.2.3.	Detekcia príznakov a konštrukcia deskriptorov.....	52
4.2.4.	Implementácia a tréning neurónovej siete typu MF ARTMAP	52
5.	Experimenty a analýza získaných výsledkov.....	55
5.1.	Výsledky získané použitím reprezentatívnej množiny s jednoduchými obrazmi	55
5.1.1.	Presnosť klasifikácie pri použití SIFT deskriptorov a 1. RM	56
5.1.2.	Presnosť klasifikácie pri použití SURF deskriptorov a 1. RM	56
5.1.3.	Presnosť klasifikácie pri použití RGB farebných zložiek obrazových bodov a 1. RM	57
5.2.	Výsledky získané použitím reprezentatívnej množiny so zložitejšími obrazmi	58
5.2.1.	Presnosť klasifikácie pri použití SIFT deskriptorov a 2. RM	59
5.2.2.	Presnosť klasifikácie pri použití SURF deskriptorov a 2. RM	59
5.2.3.	Presnosť klasifikácie pri použití RGB farebných zložiek obrazových bodov a 2. RM	60
5.3.	Analýza výsledkov.....	61
	Záver	64
	Zoznam použitej literatúry	65
	Prílohy	67

Zoznam obrázkov

Obr. 1 Základná schéma cloud computingu [5]	16
Obr. 2 Porovnanie nákladov pri spojení cloud robotiky a konvenčného modelu [5]	18
Obr. 3 Web portál Windows Azure	20
Obr. 4 Komunikácia robot – robot medzi dvoma susednými robotmi v sieti robotov [11]	22
Obr. 5 Architektúra cloud robotiky [11]	22
Obr. 6 Základná schéma systému RoboEarth [8]	25
Obr. 7 Segmentácia viacerých objektov na obraze	28
Obr. 8 Blokový diagram procesu klasifikácie a kategorizácie	29
Obr. 9 Detekcia rohu pomocou rovnako veľkého lokálneho okolia bodu pri rôznych mierkach (zváženiach) obrazu [17]	31
Obr. 10 Schéma zobrazuje rozostrené obrazy v rôznych mierkach a nasledovný výpočet rozdielov Gaussových rozostrení obrazu	32
Obr. 11 Hľadanie lokálneho extrému v rozdieloch Gaussových rozostrení obrazu	33
Obr. 12 Postup generovania SIFT deskriptoru	34
Obr. 13 Ilustrácia výpočtu sumy intenzít v integrálnom obraze	35
Obr. 14 Aproximácia Gaussových jadier pomocou 2D filtrov [19]	36
Obr. 15 Zmenšovanie mierky pôvodného obrazu a zväčšovanie 2D filtru [19]	37
Obr. 16 Princíp priradovania dominantného smeru orientácie bodu záujmu [19]	38
Obr. 17 Ilustrácia konštrukcie deskriptoru SURF	38
Obr. 18 Vplyv parametra F na tvar funkcie príslušnosti fuzzy zhľuku v jednorozmernom priestore [21]	43
Obr. 19 Vplyv parametra E na tvar funkcie príslušnosti fuzzy zhľuku v jednorozmernom priestore [21]	43
Obr. 20 Topológie MF ARTMAP siete	45
Obr. 21 Zobrazenie grafického rozhrania na zariadeniach s rôznym rozlíšením displeja.	49
Obr. 22 Základná schéma komunikácie robota Nao s cloud službou	50
Obr. 23 Ilustrácia problému tréningu neurónovej siete	53
Obr. 24 Návrh zmeny topológie MF ARTMAP siete	54

Zoznam tabuliek

Tab. 1 Parametre a presnosť neurónovej siete pri použití SIFT deskriptorov a 1. RM	56
Tab. 2 Kontingenčná a konfúzna tab. pre testovaciu množinu pri použití SIFT deskriptorov a 1. RM	56
Tab. 3 Parametre a presnosť neurónovej siete pri použití SURF deskriptorov a 1. RM.....	57
Tab. 4 Kontingenčná a konfúzna tab. pre testovaciu množinu pri použití SURF deskriptorov a 1. RM	57
Tab. 5 Parametre a presnosť neurónovej siete pri použití RGB farebných zložiek pixelov vstupného obrazu a 1. RM.....	57
Tab. 6 Kontingenčná a konfúzna tabuľka pre testovaciu množinu pri použití RGB farebných zložiek pixelov vstupného obrazu	58
Tab. 7 Kontingenčná a konfúzna tabuľka pre trénovaciu množinu pri použití RGB farebných zložiek pixelov vstupného obrazu	58
Tab. 8 Parametre a presnosť neurónovej siete pri použití SIFT deskriptorov a 2. RM	59
Tab. 9 Kontingenčná a konfúzna tab. pre testovaciu množinu pri použití SIFT deskriptorov a 2. RM	59
Tab. 10 Parametre a presnosť neurónovej siete pri použití SURF deskriptorov a 2. RM.....	59
Tab. 11 Kontingenčná a konfúzna tab. pre testovaciu množinu pri použití SURF deskriptorov a 2. RM.....	60
Tab. 12 Parametre a presnosť neurónovej siete pri použití RGB farebných zložiek pixelov vstupného obrazu a 2. RM	60
Tab. 13 Kontingenčná a konfúzna tabuľka pre testovaciu množinu pri použití RGB farebných zložiek pixelov vstupného obrazu a 2. RM	60
Tab. 14 Kontingenčná a konfúzna tabuľka pre trénovaciu množinu pri použití RGB farebných zložiek pixelov vstupného obrazu a druhej reprezentatívnej množiny.....	61
Tab. 15 Sumárny prehľad dosiahnutých výsledkov	61

Zoznam symbolov a skratiek

ART	Adaptívna Rezonančná Teória
MF	z anglického Membership Function, funkcia príslušností
MF ARTMAP	typ neurónovej siete, kombinujúci teóriu fuzzy množín a adaptívnej rezonančnej teórie
RM	reprezentatívna množina
SIFT	z anglického Scale Invariant Feature Transform, príznaková transformácia invariantná voči zmene mierky
SURF	z anglického Speeded-Up Robust Transform, rýchly róbusny detektor a deskriptor

Slovník termínov

Obraz je dvojrozmerná reprezentácia scény. Počítačové videnie zvyčajne pracuje s digitálnym obrazom vyjadreným pomocou dátovej štruktúry. [1]

Objekt je entita, ktorá bude rozpoznávaná, klasifikovaná alebo kategorizovaná [2]

Príznak je význačná veličina popisujúca jednu vlastnosť skúmaného objektu. Objekt je zvyčajne charakterizovaný viacerými príznakmi. Môže byť vyjadrený numerický alebo symbolický. [2] [3]

Príznakový priestor je n - dimenzionálny priestor, kde n je počet dimenzií príznakového priestoru. Všetky možné objekty sú situované v tomto priestore. Osi príznakového priestoru reprezentujú jednotlivé príznaky objektu. [2]

Porovnávanie je proces hľadania korešpondencií medzi dvoma alebo viacerými objektmi. [1]

Cluster reprezentuje množinu blízkych bodov v príznakovom priestore. V tejto práci cluster predstavuje jeden alebo viac objektov rovnakej kategórie. [2]

Fuzzy cluster predstavuje fuzzy množinu blízkych bodov v príznakovom priestore. [2]

Fuzzy relácia je relácia, ktorá určuje funkciu príslušností fuzzy clustra. [2]

Trieda predstavuje dobre pomenovanú množinu podobných objektov, ktoré boli do triedy klasifikované. Triedy sú zvyčajne vopred definované, nemenné a neprekrývajúce sa. [4]

Kategória je trieda, do ktorej je objekt kategorizovaný. Kategórie sa môžu prekrývať a nemusia byť vopred pomenované. [4]

Klasifikácia priraduje všetkým objektom jednu triedu v rámci systému vzájomne sa neprekrývajúcich tried. Modeluje a predpovedá nominálne atribúty (triedy). [4]

Kategorizácia je proces rozdelenia sveta (univerza) do skupín objektov, pričom objekty v rámci jednej kategórie sú si navzájom podobné. [4]

Reprezentatívna množina je usporiadané množina dvojíc $s = ((y_1, d_1), (y_2, d_2), \dots, (y_m, d_m))$, kde m je počet vzoriek v množine. Y predstavuje vstup a d predstavuje očakávaný výstup (triedu al. kategóriu). [3]

Trénovacia množina je podmnožinou reprezentatívnej množiny, používaná na tréning neurónových sietí

Testovacia množina je podmnožinou reprezentatívnej množiny, používaná na testovanie neurónových sietí

Úvod

V posledných rokoch si cloudové technológie prerazili cestu k obrovským masám používateľov. Tieto technológie už dávno nevyužívajú len odborníci. S príchodom chytrých mobilných telefónov, tabletov a s rozširovaním verejných WiFi sietí sa cloudové technológie čoraz viac približujú laickým používateľom. Tí ich využívajú denno-denne v podobe rôznych aplikácií, zdieľaných kalendárov, e-mailových kont ako napríklad Gmail, sociálnych sietí, dátových úložísk a pod.

Prakticky neobmedzený výpočtový výkon a neobmedzená veľkosť pamäťového priestoru predurčujú cloudové technológie na použitie aj v špecializovaných aplikáciách. Medzi takéto určite patria robotické aplikácie, ako je navigácia robota v priestore, uchopovanie predmetov, rozpoznávanie objektov a podobne.

Dnes už cloud technológie majú svoje pevné miesto v robotike, kde umožňujú riešiť donedávna neprekonateľné problémy. Spojeniu cloud technológií a robotiky sa hovorí cloud robotika. Práve toto spojenie sa stalo hlavnou témou tejto práce. V nej sa budeme venovať problému rozpoznávania obrazu, klasifikácií a kategorizácií objektov zo statických vizuálnych obrazov v cloud robotike.

Úloha rozpoznávania je multidisciplinárna úloha často sa vyskytujúca aj v robotike. Prakticky každý súčasný robot získava vizuálny obraz zo svojho prostredia. Či už ide o robota, ktorý vizuálne kontroluje kvalitu výrobkov na konci výrobnéj linky, alebo ide o humanoidného robota interagujúceho s deťmi, či o autonómneho prieskumného robota vo vesmíre, každý z nich spracováva obraz zo svojich kamier. Na základe výsledkov spracovania sa rozhoduje akú akciu vykoná.

Tento široký aplikačný potenciál nás viedol k spracovaniu témy rozpoznávania objektov zo statických vizuálnych obrazov pomocou metód umelej inteligencie. Konkrétne pomocou neurónových sietí typu MF ARTMAP. V práci navrhujeme systém na rozpoznávanie objektov zo statického obrazu implementovanom na cloud platforme Microsoft Windows Azure. Porovnáваме presnosť klasifikácie pri trénovaní neurónovej siete pomocou deskriptorov obrazu nájdených algoritmi SIFT, SURF. Porovnáваме aj presnosť klasifikácie pri trénovaní neurónových sietí pomocou RGB farebného spektra každého pixelu obrazu.

Pri spracovaní práce vychádzame predovšetkým z odborných článkov, monografií a internetových zdrojov zameraných na problematiku cloudových technológií, spracovania obrazu, rozpoznávania, popisu obrazu pomocou deskriptorov a v neposlednom rade zameraných aj na problematiku neurónových sietí.

1. Formulácia úlohy a cieľ práce

V predkladanej práci sa budeme venovať problematike klasifikácie a kategorizácie obrazu v cloud robotike. V úvodnej časti práce čitateľa uvedieme do problematiky cloud technológií a cloud robotiky. Predstavíme cloud, rozoberieme jeho výhody a nevýhody a stručne predstavíme cloud servisné modely. Postupne prejdeme od všeobecného cloudu ku špecifickej cloud robotike. Definujeme ju, popíšeme jej vlastnosti, výhody, nevýhody a aplikačný potenciál. V závere tejto kapitoly uvedieme niekoľko existujúcich projektov cloud robotiky.

V nasledujúcich kapitolách sústredíme našu pozornosť na teóriu kategorizácie obrazu. Spomenieme základné kroky spracovania vizuálneho obrazu v počítačovom videní a popíšeme proces rozpoznávania objektov zo statických obrazov. Podrobne vysvetlíme prácu s príznakmi a samotnú klasifikáciu, keďže tieto predstavujú podstatu procesu rozpoznávania a my ich využívame v praktickej časti práce. V rámci práce s príznakmi rozoberieme algoritmy počítačového videnia pre extrakciu príznakov z obrazov. Konkrétne algoritmy SIFT a SURF. Pri klasifikácii presne vysvetlíme rozdiel medzi klasifikáciou a kategorizáciou, poskytneme prehľad klasifikačných metód a klasifikátorov. Priblížime neurónové siete, konkrétne neurónovú sieť typu MF ARTMAP, ako hlavný klasifikátor využívaný v našej implementácii.

Postupne sa dostaneme z teoretickej roviny k praktickému návrhu systému na klasifikáciu obrazov na cloude ako službu. V tejto časti analyzujeme problém a vypracujeme teoretický návrh riešenia. V návrhu systému na klasifikáciu obrazov použijeme neurónové siete typu MF ARTMAP trénované pomocou trénovacích množín vytvorených pomocou SIFT a SURF deskriptorov a tiež pomocou farieb jednotlivých pixelov obrazu. Tu popíšeme postup tvorby systému, technologické a vedecké problémy spojené s implementáciou a spôsoby ich riešenia. Realizácia navrhnutého systému zahŕňa implementáciu neurónovej siete typu MF ARTMAP na cloud platforme Windows Azure. V tejto časti opíšeme spôsob pripojenia a spôsob komunikácie robota NAO s implementovanou cloud službou.

V závere sa budeme venovať popisu experimentov, ktoré sme vykonali na implementovanom teoretickom modeli s cieľom porovnania zvolených metód popisu obrazov. Budeme porovnávať presnosť klasifikácie neurónových sietí typu MF Artmap trénovaných pomocou deskriptorov získaných algoritmom SIFT, algoritmom SURF a pomocou RGB farebných zložiek každého pixelu v trénovacích obrazoch. Tiež budeme porovnávať presnosť klasifikácie týchto neurónových sietí pri dvoch rôznych trénovacích množinách. Jedna trénovacia množina bude obsahovať obrazy jednoduchších, ľahko rozlíšiteľných objektov a druhá bude obsahovať zložitejšie a detailnejšie obrazy objektov. Posledná časť bude venovaná zhrnutiu dosiahnutých výsledkov.

2. Cloud computing a cloud robotika

Cloud computing¹ si v posledných rokoch získava dôveru a popularitu čoraz väčšieho počtu ľudí, či už odborníkov v IT oblasti alebo laikov. Postupne sa jeho využívanie rozširuje a preniká do množstva oblastí, a to aj do špecifických a špecializovaných oblastí akou je robotika.

Aby sme mohli využívať cloud computing v robotike, musíme vedieť čo to je, aké sú jeho vlastností, výhody, nevýhody a rozdiel oproti klasickému modelu server klient. Týmito otázkami sa budeme zaoberať v prvej časti tejto kapitoly. V druhej časti sa budeme podrobnejšie venovať cloud robotike.

2.1. Definícia cloudu a cloudových služieb

IT infraštruktúra, budovaná klasickým spôsobom sa v posledných rokoch rozrástla natolko, že jej prevádzka a správa sa stala finančne náročná, a to najmä kvôli neefektívnemu využitiu výkonu, vysokej energetickej náročnosti a zložitej správe. [5] Pod klasickým budovaním IT infraštruktúry rozumieme budovanie, nasadzovanie, udržiavanie a spravovanie vlastných informačno-komunikačných systémov, serverov a počítačových sietí v rámci každej organizácie osobitne. Klasické IT infraštruktúry sú vhodnými kandidátmi na optimalizáciu vedúcu k zníženiu nákladov spoločností. Jedným z možných nástrojov optimalizácie existujúcich IT infraštruktúr je využitie cloud computingu.

Podľa Oxfordského výkladového slovníka je cloud computing definovaný ako prístup využívajúci sieť vzdialených serverov hostovaných na internete, slúžiacich na ukladanie, správu a spracovanie dát, pričom využívanie týchto vzdialených serverov je uprednostňované pred využívaním lokálnych serverov alebo osobných počítačov [6]

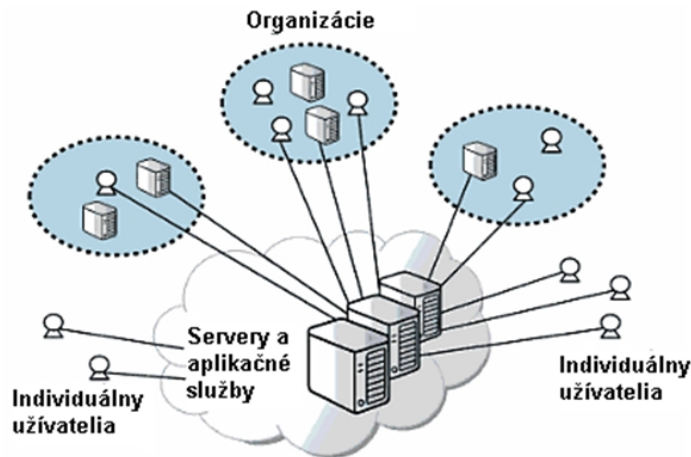
Rozborom uvedenej definície môžeme o cloud computingu konštatovať nasledujúce:

- cloud computing je nový model vývoja a používania firemných aplikácií, softvérových platforiem a hardvérovej infraštruktúry [5]
- aplikácie a dáta sú uložené na vzdialených cloudových serveroch a používateľovi sú prístupné pomocou webového prehliadača. Tieto aplikácie používateľ využíva ako službu. Výhodou je možnosť pristupovať k uloženým dátam a aplikáciám kedykoľvek, z ľubovoľného miesta na zemi a pomocou akéhokoľvek elektronického zariadenia pripojiteľného do siete internet.

¹ Slovné spojenia ako je cloud, cloud robotika a cloud computing sú prebraté z angličtiny a nezvyknú sa prekladať. Preto v tejto práci budeme používať anglické verzie pojmov.

- Fyzická infraštruktúra je pre používateľa skrytá a používateľ využíva len služby cloudu. Teda využíva len funkcionality cloudu, pričom nemusí vlastniť hardvérovú infraštruktúru a softvérové licencie pre používaný softvér. Tým odpadá potreba nákladného budovania, nasadzovania a správy vlastných IT infraštruktúr.

Na obr. 1 je znázornená základná schéma cloud computingu.



Obr. 1 Základná schéma cloud computingu [5]

2.1.1.1. Servisné modely cloudu

Existujú tri základné modely, v ktorých je cloud poskytovaný, a to:

- IaaS – Infrastructure as a Service – infraštruktúra ako servis
- PaaS – Platform as a Service, platforma ako servis
- SaaS – Software as a Service, softvér ako servis

V literatúre sa niekedy objavujú aj iné modely cloudu, ako napríklad HaaS – Human as a Service, ktorý zahŕňa ľudské služby, ktoré sa nedajú rovnako kvalitne nahradiť informačnými technológiami. Ide napríklad o dizajnérske služby. [5] Ďalšie typy modelov môžu byť DaaS – Desktop as a Service alebo BPaaS – Business Process as a Service a pod. [7] V nasledujúcom texte charakterizujeme základné servisné modely cloudu.

2.1.1.1.1. IaaS – infraštruktúra ako servis

Infraštruktúra ako servis je model, ktorý ponúka používateľom hardvérové prostriedky pripravené a nakonfigurované na použitie. Zväčša sú tieto hardvérové prostriedky ponúkané vrátane základného softvéru, ako je operačný systém s nastavenou IP adresou, sieťovou konfiguráciou a diskovým priestorom. Tento model umožňuje používateľovi vytvárať a manipulovať s virtuálnymi servermi, pričom správa týchto virtuálnych strojov je ponechaná na používateľovi. Poskytovateľ cloudovej služby zabezpečuje len nepretržitú dostupnosť vytvorených virtuálnych strojov. [7] [8]

Výhodou IaaS modelu je plná kontrola používateľa nad vytvorenými virtuálnymi strojmi a ich škálovateľnosť. To znamená, že používateľ môže ľubovoľne meniť konfigurácie využívaných virtuálnych serverov a dátových úložísk a tak prispôbovať výpočtový výkon a pamäťový priestor svojim aktuálnym potrebám. Zaujímavým príkladom je prevádzka oficiálnej web stránky veľkej športovej udalosti ako napríklad Olympijských hier. Oficiálna stránka je pred a dlho po uskutočnení športovej udalosti navštevovaná minimálne. Naopak s blížiacim sa dátumom začiatku udalosti a počas samotnej udalosti sú oficiálne web stránky masovo navštevované, čím sa zvyšujú nároky na serveri, ktoré stránky hostujú. Škálovateľnosť nám umožní jednoducho zväčšiť výpočtové kapacity serverov počas doby, kedy je zvýšená návštevnosť. Dokonca môžeme dočasne vytvoriť nové virtuálne serveri a nasadiť na nich celý web portál, čím sa záťaž na jednotlivé serveri rozdelí.

2.1.1.2. PaaS – platforma ako servis

Platforma ako servis je cloud model, ktorý poskytuje používateľovi platformu pre vývoj vlastných aplikácií ako cloud služieb, ktoré budú nasadené v infraštruktúre poskytovateľa cloudového riešenia. Poskytovateľ tiež zabezpečuje dostupnosť a správu infraštruktúry. Používateľ pri tomto modeli nemá právo pracovať s nastaveniami virtuálnych strojov, na ktorých sú nasadené vytvorené aplikácie.

Typickými príkladmi platformy ako servis je Windows Azure od spoločnosti Microsoft. Platforma Windows Azure je operačný systém pre služby typu cloud computing. Slúži ako prostredie pre vývoj, hostovanie a správu cloudových služieb, vyvinutých a nasadených na tejto platforme. Windows Azure využívame v praktickej časti tejto diplomovej práce. Podrobnejšie bude predstavená v kapitole 2.1.3.1. Inými príkladmi PaaS sú platformy Facebook Platform od spoločnosťou Facebook alebo AppEngine od Googlu. Zatiaľ čo Facebook Platform poskytuje prostredie pre tvorbu aplikácií pre sociálnu sieť Facebook, AppEngine predstavuje škálovateľné a výkonné prostredie pre tvorbu web aplikácií. [5]

2.1.1.3. SaaS – softvér ako servis

Softvér ako servis je cloud model, ktorý ponúka zákazníkovi finálny softvérový produkt. Poskytovateľ zabezpečuje neustálu dostupnosť, aktualizáciu softvéru a zabezpečenie infraštruktúry. Používateľ iba využíva softvér ako službu, pričom má právo meniť len osobné nastavenia a nastavenia svojho účtu, pomocou ktorého sa prihlasuje do služby.

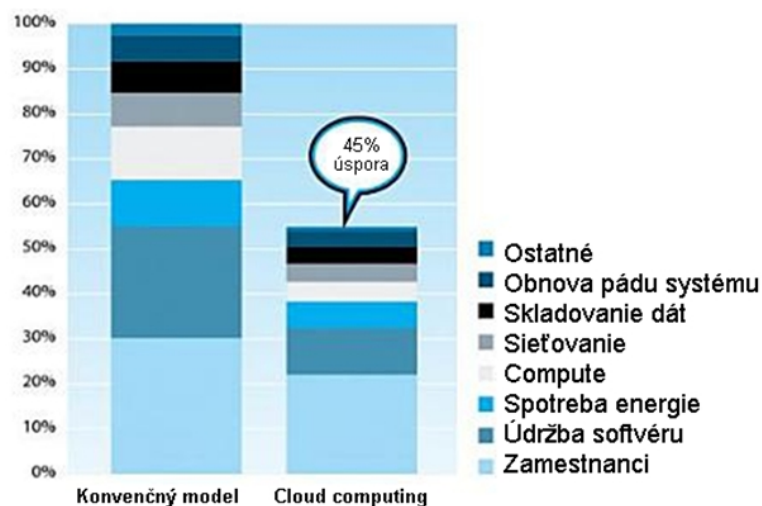
Výhodou tohto modelu je jednoduchá správa aplikácií. Softvérový vývojár, ktorý používa tento model na distribúciu svojho softvéru, môže jednoducho aktualizovať, modifikovať a

upravovať svoj softvér a všetky zmeny sa okamžite prejaví u všetkých používateľov. Odpadá tak potreba vydávania aktualizácií.

Cloudový model SaaS má podobu klasickej web aplikácie, často poskytovanej zadarmo. Tento cloudový model používa asi každý používateľ internetu, pričom mnohokrát netuší, že využíva cloudovú službu. Ako príklady môžeme spomenúť známe sociálne siete Facebook, Twitter, Google+, e-mailovú službu Gmail, Google drive, OneDrive, Office 365 a pod.

2.1.2. Výhody cloudu

V posledných rokoch sa stal cloud computing moderným a atraktívnym nástrojom. Snáď najdôležitejšou výhodou cloud computingu je jeho cena. Napriek tomu, že používateli sa môže zdať na prvý pohľad vysoká, treba si uvedomiť, že náklady spojené s budovaním vlastnej infraštruktúry, spravovaním a udržiavaním sú omnoho vyššie. Na obr. 3 je graf zobrazujúci úsporu nákladov pri použití cloud computingu v porovnaní s konvenčným modelom.



Obr. 2 Porovnanie nákladov pri spojení cloud robotiky a konvenčného modelu [5]

Ďalšou výhodou je samoobslužnosť. Používateľ si objednáva služby, nastavuje ich, vytvára virtuálne stroje kedykoľvek to potrebuje a zmeny sa prejavujú okamžite. So samoobslužnosťou súvisí aj flexibilita a využiteľnosť prostriedkov. Flexibilita cloudu umožňuje pružne reagovať na meniace sa potreby. Napríklad zvýšenie nárokov na pamäťový priestor je možné vyriešiť okamžite rozšírením objednaného pamäťového priestoru o dodatočnú kapacitu. Ak hovoríme o využiteľnosti prostriedkov, bolo štatisticky dokázané, že využiteľnosť zdrojov v cloude je 80%, zatiaľ čo v klasickej infraštruktúre len 30%. V prípade, že už nepotrebujeme objednaný rozsah zdrojov, môžeme časť z nich vrátiť poskytovateľovi a on ich môže ponúknuť inému používateľovi.

Nesmieme zabúdať ani na redundanciu dát v cloude, ktorá zabezpečuje ich spoľahlivosť, vysokú dostupnosť a bezpečnosť. Jednotlivé cloud služby je možné spolu prepojiť. Pre vývojárov aplikácií

typu SaaS plynie ešte jedna dôležitá výhoda cloudu, a to tá, že pri vývoji softvéru nemusia brať ohľad na manažment a správu hardvéru, ale môžu sa venovať len vývoju softvéru.

2.1.3. Poskytovatelia cloud služieb

V súčasnosti existuje mnoho poskytovateľov cloudových služieb, cloudových riešení a platforiem, avšak naozaj veľkých nadnárodných poskytovateľov ponúkajúci komplexné balíky služieb je len niekoľko. Najznámejší z nich sú spoločnosti Google, Microsoft a Amazon. Vzhľadom na ciele tejto práce a jej obmedzený rozsah, budeme sa venovať iba Microsoft Windows Azure platfore, pre ktorú vyvíjame službu na rozpoznávanie obrazu v praktickej časti tejto práce.

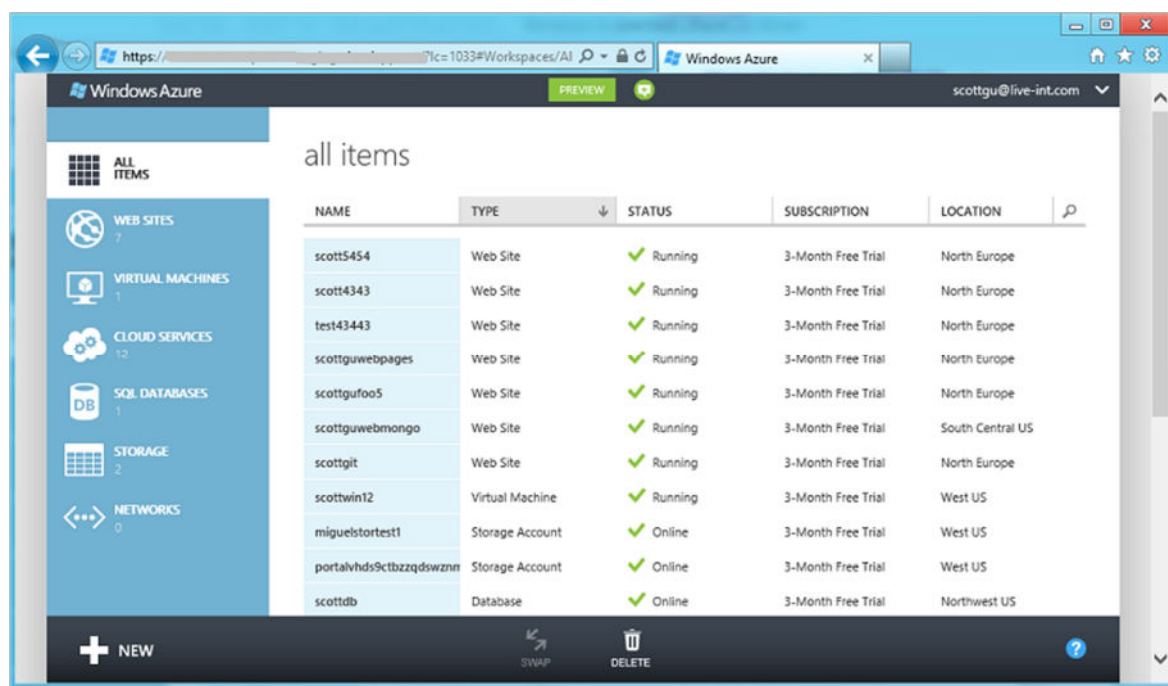
2.1.3.1. Microsoft Windows Azure

Ako už bolo spomenuté v kapitole 2.1.1.2, Windows Azure je operačný systém pre služby typu cloud computing. Poskytuje celú škálu funkcií pre tvorbu rôznych aplikácií od jednoduchých web stránok až po podnikové informačné systémy. Okrem samotného operačného systému ponúka ďalšie nástroje a služby pre vývojárov, z ktorých môžeme spomenúť Microsoft SQL Azure alebo Windows Azure AppFabric. Microsoft SQL Azure je relačná databázová služba na cloude, ktorá zjednodušuje prevádzku veľkého počtu databáz. Windows Azure AppFabric je služba, ktorá vývojárom pomáha prepojiť cloudové aplikácie navzájom resp. umožňuje prepojiť cloudové aplikácie s interne nasadenými aplikáciami vo firme. [9]

Pri delení podľa modelov cloudu môžeme Windows Azure zaradiť do modelu PaaS a tiež do modelu IaaS. Vo Windows Azure sa tieto dva modely vzájomne prelínajú.

Azure je flexibilná platforma, ktorá umožňuje škálovanie výpočtových zdrojov presne podľa aktuálnych potrieb používateľa. Používateľ môže využívať službu automatického škálovania, ktorá používateľovi šetrí čas potrebný na zmenu požadovaných výpočtových kapacít. Tiež šetrí náklady spojené s neprimerane veľkou objednávkou výpočtových kapacít, ktoré práve nie sú využívané.

Pre potreby správy výpočtových kapacít, nastavení, tvorby a správy databáz, správy cloud dátového úložiska (z angl. cloud storage) a virtuálnych strojov existuje web portál s intuitívnym grafickým rozhraním, pomocou ktorého je možné vykonávať všetky nastavenia. Náhľad tohto web portálu je znázornený na obr. 3.



Obr. 3 Web portál Windows Azure

Aplikácie nasadené na platforme Windows Azure je možné vyvíjať pomocou vývojového prostredia Microsoft Visual Studio. Tohto vývojové prostredie umožňuje aj nasadzovanie vytvorenej aplikácie na cloud. Windows Azure podporuje širokú škálu programovacích jazykov od C#, .NET, VB cez Javu, Python, PHP až po Node.js a Ruby. Tiež sú podporované všetky najrozšírenejšie štandardy a protokoly ako SOAP, REST, XML alebo PHP. [9]

Windows Azure je spoplatnená platforma. Cena za jej využívanie závisí od požiadaviek klienta, od počtu virtuálnych serverov, počtu výpočtových jadier, počtu databáz, veľkosti požadovaného pamäťového miesta a pod. Pre výpočet presnej ceny je možné použiť interaktívnu kalkulačku na stránke <http://www.azure.microsoft.com/en-us/pricing/calculator/?scenario=full>. Spoločnosť Microsoft ponúka neobmedzenú trial verziu funkčnú 30 dní úplne zadarmo.

2.2. Spojenie robotiky a cloud služieb

V prvej časti trilógie Matrix je scéna, kde hlavný hrdina Neo stojí na streche mrakodrapu pri vrtuľníku a pýta sa Trinity, či vie pilotovať tento vrtuľník. Jej odpoveď je: „Zatiaľ nie.“ Následne sa však pripojí ku centrálnemu úložisku znalostí a do svojho mozgu si stiahne program na pilotovanie helikoptéry. Potom im už nič nebráni v tom, aby mohli spolu odletieť. V popísanej scéne je naznačená základná myšlienka cloud robotiky, ktorá hovorí o tom, že robot nemusí byť obmedzený výpočtovým výkonom determinovaným hardvérovou konfiguráciou robota, kapacitou jeho pamäte alebo jeho softvérovým vybavením. [10]

2.2.1. Definícia cloud robotiky

Pojem cloud robotika bola prvý krát použitá v roku 2010 Jamesom Kuffnerom, pracujúcim v laboratóriách Google. Tento pojem použil na opis nového rozvíjajúceho sa prístupu v robotike, ktorý je založený na cloud výpočtoch. Masívne pritom využíva cloudové služby, cloudové dátové [11]úložiská a všetky ostatné internetové technológie, umožňujúce masívne paralelné výpočty a zdieľanie obrovského množstva dát [12] [13]. Jej hlavným cieľom je vytvorenie infraštruktúry pre robotov, ktorá by umožnila komunikáciu medzi robotmi, nahrávanie a sťahovanie znalostí o riešenom probléme alebo sťahovanie kompletných programov pre riešenie konkrétnych problémov. Táto infraštruktúra predstavuje akýsi internet pre robotov. Využitie cloudu v robotike umožňuje odbremenenie robotov od výpočtovo náročných úloh, presunutím zložitých výpočtov z robotov do cloudového prostredia. Tam prebehne výpočet a robotovi sa vráti len výsledok. Týmto sa robot stáva nezávislý na vlastnej hardvérovej konfigurácii, pamäťových kapacitách a softvérovom vybavení.

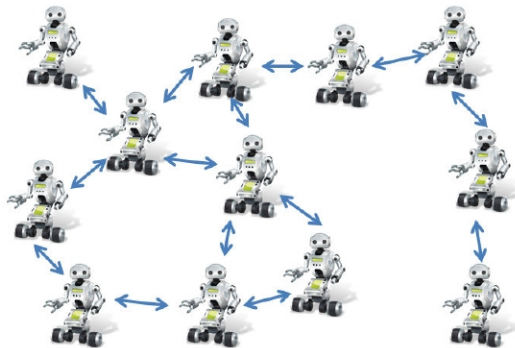
Avšak nie všetky úlohy sú riešiteľné popísaným spôsobom. Príkladom sú úlohy súvisiace s pohybom robota – stabilita, vyhýbanie sa prekážkam, samotný pohyb a pod. Hovoríme o tzv. kritických úlohách, ktoré musia byť vykonávané v reálnom čase. Nie je pri nich prípustné zdržanie vznikajúce pri komunikácii po sieti, preto sa musia vykonávať priamo na robotovi. Naopak úlohy, ktorých vykonávanie je možné presunúť na cloud a prípadné zdržanie neznižuje kvalitu ich vykonávania nazývame nekritické úlohy. Je dôležité vhodne rozdeliť množinu úloh na kritické a nekritické. [8]

2.2.2. Porovnanie siete robotov a cloud robotiky

Sieť robotov a cloud robotika sú si podobné. Dá sa povedať, že sieť robotov je generačný predchodca cloud robotiky. Pomocou siete robotov boli vyvinuté základné myšlienky a princípy dnes používané v cloud robotike. Nástupom cloud computingu sa vytvorili podmienky pre odstránenie niektorých podstatných problémov siete robotov. Teda cloud robotiku môžeme vnímať ako nástupcu siete robotov, resp. ako sieť robotov využívajúcu cloud computing na odstránenie niektorých svojich obmedzení.

Pod sieťou robotov rozumieme skupinu robotov, ktorí sú pripojení do jednej siete a vzájomne spolupracujú. Rozlišujeme dve základné aplikačné oblasti sieťovej robotiky, a to je teleoperácia a multi-robotické systémy. Pri teleoperácii človek – operátor riadi robota na diaľku cez sieť. Napríklad riadenie prieskumného robota na mesiaci, alebo operácia pomocou robota DaVinci. Aplikácia multi-robotického systému predstavuje kolaboráciu viacerých robotov pripojených do jednej siete pri riešení konkrétnej úlohy. Príklad je skupina prieskumných a záchranárskych

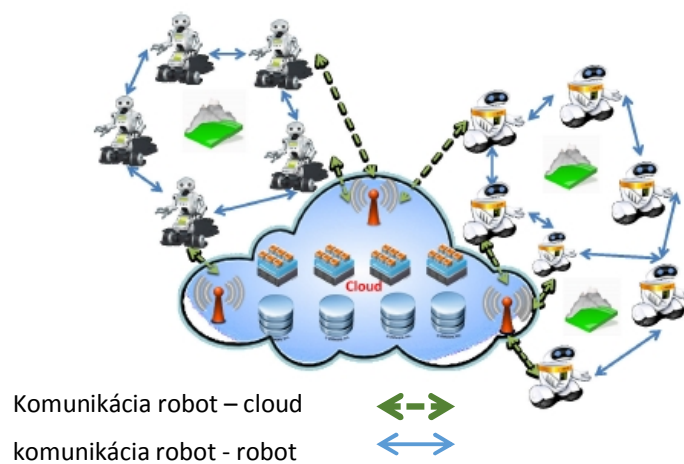
robotov, ktorý hľadajú nezvestných v troskách budov [11]. Na obr. 4 je znázornená sieť robotov, v ktorej každý robot komunikuje so svojimi najbližšími susedmi.



Obr. 4 Komunikácia robot – robot medzi dvoma susednými robotmi v sieti robotov [11]

Veľkým problémom siete robotov je, že všetky výpočty sa vykonávajú v tejto sieti. Teda prebiehajú priamo na hardvéry robotov. Napriek tomu, že môžeme celkový výpočet rozdeliť medzi viacerých robotov, stále sme obmedzení ich výpočtovými výkonmi a ich pamäťovými kapacitami. Iným veľkým obmedzením je fakt, že v rámci siete robotov nemôžeme získať ľubovoľnú informáciu. Prístupné sú len informácie, ktoré roboti dokážu získať pomocou svojich senzorov a výpočtov a ktoré navzájom zdieľajú. V neposlednom rade v sieti robotov existujú aj komunikačné obmedzenia vyplývajúce z vlastností komunikačných protokolov. [11]

Použitím cloud technológií s možnosťou škálovateľnosti výpočtových zdrojov a použitím cloud infraštruktúry je možné niektoré z týchto obmedzení odstrániť. Na obr. 5 je architektúra cloud robotiky s robotmi, ktorí komunikujú medzi sebou (robot – robot komunikácia) a zároveň komunikujú aj cloudovými službami (robot – cloud komunikácia).



Obr. 5 Architektúra cloud robotiky [11]

2.2.3. Výhody a aplikačný potenciál cloud robotiky

Zdá sa, že jediná veľká nevýhoda cloud robotiky je nutnosť neustáleho pripojenia robota k internetu. To je však vyvážené nasledujúcimi výhodami:

- **Možnosť presunutia náročných výpočtov z robota na cloud.** V takom prípade na jednoduchšie výpočty, ktoré musia byť vykonávané priamo v robotovi stačí aj slabšia hardvérová konfigurácia. Tým je možné predĺžiť životnosť batérie robota. Samotná robotická platforma sa takto zjednodušuje, stáva sa lacnejšou a jednoduchšou na údržbu. [11]
- **Prístup k obrovskému množstvu dát.** Robot má prístup ku cloud databázam a veľkému množstvu dát a informácií o riešenej úlohe alebo probléme. Nepotrebuje tak vytvárať a spravovať vlastnú databázu. [11]
- **Prístup ku zdieľaným znalostiam a skúsenostiam.** Pomocou cloudu je zabezpečený prístup robotov ku zdieľaným znalostiam a skúsenostiam iných robotov uložených v cloud pamäťových úložiskách. Takto majú možnosť učiť sa od iných robotov. Touto problematikou sa zaoberá projekt RoboEarth. [11]
- **Roboti môžu ukladať svoje dáta a znalosti** do cloud databázy a zdieľať ich s inými robotmi. Teda robot nemusí byť len v pozícii používateľa znalostí z cloudu, ale môže zastávať aj pozíciu aktívneho tvorca znalostí. [8]

Spomínané výhody naznačujú veľký aplikačný potenciál cloud robotiky predovšetkým v úlohách, ktoré spracúvajú veľké množstvá dát a vo výpočtovo náročných úlohách. Vo všeobecnosti môžeme vyčleniť štyri oblasti úloh, ktorých výkonnosť môže byť značne zvýšená pomocou cloud robotiky. Sú to 1) indexácia obrovských databáz obrázkov a modelov objektov, 2) masívne paralelné na príkladoch založené plánovanie pohybu a plánovanie neistoty, 3) zdieľanie výsledkov, trajektórií a dynamických stratégií kontroly v bežne používaných robotických systémov, 4) získanie ľudského dozoru na vyžiadanie. [14]

Z konkrétnych aplikácií cloud robotiky môžeme spomenúť rozpoznávanie objektov, uchopovanie predmetov, navigáciu robota v priestore, tvorba mapy prostredia, v ktorom robot existuje a pod.

2.2.4. Existujúce projekty cloud robotiky

2.2.4.1. Projekt „vzdialeného mozgu“

Hlavná myšlienka cloud robotiky, ktorá hovorí o pripojení robota na externý počítač pomocou internetu je pomerne stará. Už v 90-tých rokoch 20. Storočia na Univerzite v Tokiu prebiehal projekt, v ktorom bol vynájdený koncept „vzdialeného mozgu“. Bol to pokus fyzicky oddeliť senzory a motory robota od vysoko-úrovňového mysliaceho softvéru. Projekt prebiehal v čase

ranného internetu s veľmi obmedzeným výpočtovým výkonom a možnosťami. Realizáciu konceptu „vzdialeného mozgu“ umožnil až rozvoj cloudu s obrovským výpočtovým výkonom. Napriek tomu, že v tomto projekte ešte nemôžeme hovoriť o cloud robotike je významný rozpracovaním základných myšlienok.

2.2.4.2. Projekt DAVinCi

Podstatou projektu DAVinCi je tvorba mapy prostredia, ktorom robot existuje. Tvorba máp prostredia je klasickou robotickou úlohou. Zvyčajne sú mapy robotom alebo autonómnym vozidlom a to bez akejkoľvek znalostí o mapovanom prostredí. Úloha je splnená ak robot vytvorí mapu a správne sa v nej lokalizuje. Projekt DAVinCi, však použil zdieľané cloud dátové úložisko, čo umožnilo vytvárať mapu kolaboráciou viacerých robotov. Tí postupne zbierajú informácie z prostredia, odosielajú ich do cloud služby, ktorá ich spracováva a integruje do kompletnej mapy prostredia. Kompletná mapa sa ukladá v zdieľanom dátovom úložisku, odkiaľ roboti môžu čerpať informácie o prostredí. [15]

2.2.4.3. Navigácia robota v priestore

S úlohou mapovania prostredia súvisí aj úloha navigácie robota. Ide o aktivitu robota, pri ktorej sa presunie z aktuálnej pozície na zadanú pozíciu. Vo všeobecnosti existujú dva prístupy. Prvý prístup je navigácia bez mapy. V takom prípade sa robot pohybuje len na základe informácií zo svojich senzorov. Tento prístup je častokrát nespoľahlivý. Druhá skupina metód je založená na tvorbe máp, čo je úloha tvorby máp. Pri navigácií, však vytvorené mapy potrebujeme analyzovať a napláňovať trasu. Proces analýzy mapy vyžaduje prehľadávanie veľkého množstva dát. Pri plánovaní zase potrebujeme veľký výpočtový výkon. Obe požiadavky môžu byť splnené použitím cloud technológií. [11]

2.2.4.4. Uchopovanie objektov

Úloha uchopenia predmetu robotom je netriviálna úloha. Vyžaduje sa buď poznanie presného 3D modelu objektu alebo, ak takýto 3D model nie je známy, resp. je známy len čiastočne, vyžaduje sa získanie a predspracovanie veľkého množstva dát a vytvorenie modelu. To môže byť výpočtovo náročná operácia. Po získaní 3D modelu je podľa vybraného algoritmu uchopenia generovaný pohyb a samotné uchopenie predmetu. Cloud technológie sa tu využívajú na ukladanie 3D modelov a ich zdieľanie. V prípade potreby tvorby 3D modelu sa využíva veľký výpočtový výkon cloudu. [11] Tejto problematike sa venuje viacero výskumov. Spomenúť môžeme projekt uchopovania objektu pomocou Google object recognition object engine, ktorý prebiehal na Kalifornskej univerzite v Berkeley.

2.2.4.5. Projekt RoboEarth

RoboEarth je projekt, ktorý si dáva za cieľ vytvoriť internet pre robotov, t.j. gigantickú sieť a databázy, kde rôzne druhy robotov budú môcť zdieľať znalosti a učiť sa jeden od druhého. [12] Na budovanie tejto siete sa využíva sa cloud infraštruktúra.

Cieľom je dosiahnuť stav, kedy ľubovoľný robot bude schopný rozpoznať príkaz a dekomponovať ho na postupnosť akcií. Následne zisti, ktorú akciu nevie vykonať a nájde pre ňu vhodnú znalosť alebo akčný plán v databáze RoboEarth. Nakoniec zo všetkých znalostí a čiastočných akčných plánov pre jednotlivé akcie vyskladá kompletný akčný plán. Vykonaním kompletného akčného plánu robot splní zadaný príkaz. [8]

Za celý opísaný proces je zodpovedný RoboEarth Engine, ktorý je na cloude. Jeho výhodou je možnosť výberu akcie. Pre jednu úlohu zvyčajne existuje viacero prípustných možností riešenia (viacero možných akcií). Robot si môže vybrať takú akciu, ktorá je pri aktuálnych podmienkach najvhodnejšia. Celkový akčný plán sa vyskladá z rôznych akcií. Tento prístup sa nazýva AI brick. [8]

Znalosti v RoboEarth databáze, môžu byť trojakého typu, a to mapy prostredia, akčné plány alebo popis objektov, s ktorými robot interaguje. (napr. 3D model objektu, ktorý má uchopiť). Sú získavané z teleoperačného riadenia robotov. Človek – operátor, riadi robota pri vykonávaní určitej úlohy. V tomto procese sa zbierajú dáta zo všetkých dostupných senzorov a motorov a analyzujú sa. Zo zozbieraných dát sa vytvorí akcia pre riešenie úlohy (tzv. akčný plán) a uloží sa do zdieľanej databázy. Tento proces sa nazýva učenie sa z teleoperácie. [8]



Obr. 6 Základná schéma systému RoboEarth [8]

Projekt RoboEarth je ambiciózny a zároveň veľmi zaujímavý, ktorého realizácia by bez cloud technológií nebola možná.

2.2.5. Zhrnutie cloud technológií

Cloud technológie si získali svoje pevné miesto pri budovaní IT infraštruktúry a ponúkli nám tak silné výpočtové a pamäťové možnosti, ktoré ešte pred niekoľkými neboli mysliteľné. Preto cloud sa stal vhodným nástrojom na ukladanie obrovského množstva dát a na vykonávanie extrémne náročných výpočtov, aké môžu vznikáť pri realizácii algoritmov umelej inteligencie a v robotike. Z tohto uhla pohľadu sa cloud javy ako ideálny nástroj na použitie v robotike. To dokazuje viacero úspešných projektov ako napríklad RoboEarth.

3. Rozpoznávanie objektov vo vizuálnom obraze

Pre vyššie biologické organizmy a ľudí je videnie prirodzené a rozpoznávanie objektov vo videnom obraze je triviálna záležitosť. Samozrejme za predpokladu, že jedinec netrpí zrakovým postihom. Ľudia bez najmenších ťažkostí dokážu detailne rozpoznávať farby, tvary, materiály, veľkosti a mnohé iné atribúty pozorovaných objektov. Pozorované objekty vedia kategorizovať do viac ako 30 000 kategórií v čase kratšom ako 150ms. [2]

V prípade, že požadujeme aby videné objekty rozoznával počítač, robot alebo iný umelý stroj, triviálna úloha sa rázom stáva náročnou. Náročnosť stúpa z niekoľkých dôvodov ako sú napríklad strata informácie pri prechode z 3-dimezióneho priestoru do 2-dimezióneho, obrovský tok dát generovaných snímačmi obrazu, šumy reálnych obrazov a ďalšie. [16] Problematikou videnia umelých systémov sa zaoberá disciplína nazývaná počítačové videnie. Počítačové videnie je silne multidisciplinárna vedná disciplína, ktorá zahŕňa umelú inteligenciu, strojové učenie, matematiku, spracovanie obrazu a signálov, fyziku, robotiku a iné. [16]

Aplikačný potenciál počítačového videnia je široko spektrálny. Zhŕňa aplikácie využívané v rôznych oblastiach. Príkladmi sú aplikácie ako dekodér QR kódov v mobilných telefónoch, rozpoznávanie ŠPZ automobilov pri mýtnych bránach nad diaľnicami, navigácia autonómnych vozidiel vo fabrikách pomocou sledovania čiar na zemi, kontrola kvality výrobkov na koncoch výrobných liniek alebo aplikácie rozpoznávania objektov v robotike.

V praktickej časti práce implementujeme program slúžiaci na klasifikáciu objektov v dvojrozmernom vizuálnom obraze, čo je úloha spadajúca do počítačového videnia. Preto sa v nasledujúcom texte budeme zaoberať procesom klasifikácie a kategorizácie objektov z vizuálnych obrazov.

3.1. Spracovanie vizuálnych obrazov

V počítačovom videní existuje základný postup spracovania vizuálneho obrazu vedúci ku rozpoznávaniu objektu na obraze. Tento postup pozostáva z nasledujúcich krokov:

1. **Získanie digitálneho obrazu.** Obraz už môže byť v digitálnej podobe uložený v počítači. V inom prípade ho môžeme získať fotografovaním alebo skenovaním. Vtedy dochádza k prevodu spojitej obrazovej funkcie analógového obrazu na diskretnú obrazovú funkciu vhodnú pre digitálne obrazy.
2. **Predspracovanie obrazu** slúži na úpravu obrazu a jeho zlepšenie pre potreby ďalšieho spracovania. Pri predspracovaní sa využíva nadbytočnosť údajov v obraze.

Predspracovaním môžeme odstrániť šum, upraviť jas, vykonať geometrické transformácie, detekovať hrany a pod.

3. **Segmentácia** je proces, ktorým dochádza k rozdeleniu obrazu na disjunktné časti, zodpovedajúce objektom reálneho sveta. Opätovným zjednotením segmentovaných oblastí dostaneme pôvodný obraz. Príkladom je obraz, na ktorom je viacero objektov. My máme záujem klasifikovať len jeden z nich. Segmentáciou oddelíme jednotlivé objekty a potom môžeme pracovať už len so zvoleným objektom. Táto situácia je zobrazená na obr. 7.

4. Proces rozpoznávania objektov



Obr. 7 Segmentácia viacerých objektov na obraze

Z dôvodu obmedzeného rozsahu tejto práce sa v ďalšom budeme venovať len samotnému procesu klasifikácie a kategorizácie objektov.

3.2. Proces rozpoznávania objektov

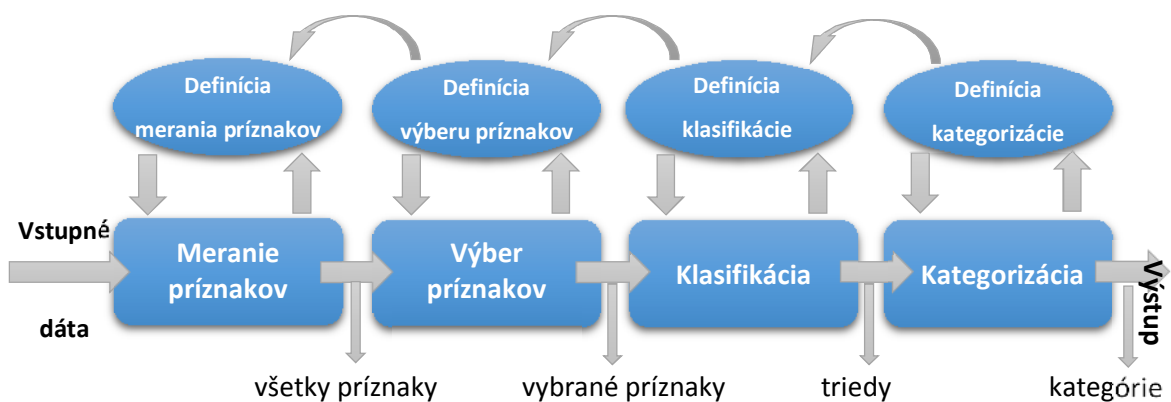
V prípade, že množinu rozpoznávaných objektov je možné rozdeliť do tried, tak úloha rozpoznávania objektov sa stáva klasifikačnou úlohou. Preto v nasledujúcom texte budeme hovoriť skôr o klasifikácií a kategorizácií ako o rozpoznávaní. Klasifikácia a kategorizácia sú si podobné. Presný rozdiel medzi nimi bude popísaný v podkapitole 3.4.1. Samotný proces rozpoznávania objektov je možné rozdeliť do nasledujúcich krokov:

1. **Definícia merania príznakov.** Vyberáme metódu merania príznakov vo vizuálnom obraze. Existuje viacero metód merania príznakov. Každá z nich je vhodnejšia na riešenie inej skupiny úloh. Preto je potrebné otestovať viacero metód. Často nevieme, ktoré príznaky budú dávať najlepšie výsledky. Preto výber metódy by mal byť iteratívny a mal by byť založený na podrobnej analýze vstupných dát. Pozn.: meranie príznakov znamená získať atribúty (vlastnosti) obrazu alebo objektu na obraze.
2. **Meranie príznakov** predstavuje samotnú extrakciu príznakov pomocou zvolenej metódy z predchádzajúceho kroku. V prípade potreby je možné sa vrátiť ku kroku 1, vybrať inú metódu a extrakciu opakovať.

3. **Definícia výberu príznačov.** Extrakciou príznačov získame všetky príznačky, ktoré je možné vyextrahovať zo vstupného obrazu pomocou zvolenej metódy. Nie všetky príznačky sú však potrebné pre ďalšie spracovanie. Tento krok slúži na výber metódy, ktorá určí nepotrebné príznačky a zredukuje množinu vyextrahovaných príznačov ich vylúčením.
4. **Výber príznačov** predstavuje vykonanie vybraného algoritmu redukcie príznačov. Redukcia príznačov je totožná s redukciami dimenzií príznačového priestoru. V prípade, že po redukcii nám ostane málo príznačov, znamená to, že príznačový priestor je príliš redukovaný a následná klasifikácia a kategorizácia nemusí byť presná. Preto je vhodné vrátiť sa k predchádzajúcemu kroku a zvoliť iný algoritmus extrakcie.
5. **Definícia klasifikácie** slúži na výber klasifikačnej metódy, ktorú budeme používať na klasifikáciu. Výber metódy závisí od charakteru vstupných dát.
6. **Klasifikácia** predstavuje implementáciu vybranej klasifikačnej metódy a samotnú klasifikáciu tréningových príkladov.
7. **Definícia kategorizácie** slúži na výber algoritmu pre kategorizáciu. Podobne ako pri definícii klasifikácie výber metódy je závislý na charaktere vstupných dát.
8. **Kategorizácia** znamená vykonanie algoritmu kategorizácie.

Po vykonaní každého kroku, vyhodnotíme vykonaný krok a môžeme navrhnúť zmenu vybranej metódy s cieľom dosiahnuť lepšie čiastkové výsledky. Po vykonaní všetkých krokov a získaní výsledkov, sa môžeme vrátiť k ľubovoľnému kroku alebo k súboru krokov a zmeniť použité metódy s cieľom zlepšiť celkovú presnosť klasifikácie resp. kategorizácie. Na obr. 8 je znázornený blokový diagram procesu klasifikácie.

Pozn.: Body 1 – 4 predstavujú prácu s príznačkami obrazu a body 5 – 8 predstavujú rozpoznávanie (klasifikáciu a kategorizáciu).



Obr. 8 Blokový diagram procesu klasifikácie a kategorizácie

3.3. Práca s príznakmi

Ako už bolo spomenuté, rozpoznávanie je klasifikačná úloha, ktorá rozhoduje o tom, do ktorej triedy patrí neznámy objekt. Klasifikácia však neprebíha na základe objektu, ale na základe jeho popisu. Preto ak chceme klasifikovať objekt zobrazený v dvojrozmernom vizuálnom obraze, musíme tento obraz predspracovať, segmentovať a exaktne popísať.

Vizuálne obrazy sa popisujú pomocou príznakov. Príznačky sú merateľné atribúty (vlastnosti) obrazu alebo objektu v obraze. Z toho vyplýva, že cieľom práce s príznakmi je ich extrakcia, redukcia a vhodná reprezentácia tak, aby sme získali exaktný popis obrazu (objektu v obraze), ktorý môže byť predložený klasifikátoru na zatriedenie obrazu (objektu) do príslušnej triedy.

Príznačky extrahujeme z vizuálneho obrazu pomocou detektorov. Sú to nástroje, ktoré určujú body záujmu v obraze. Poznáme mnoho rôznych detektorov, napríklad hranový detektor, Cannyho detektor, gradientový detektor a iné. Viac informácií o detektoroch, ich vlastnostiach, princípoch a ich činnosti je možné nájsť v [16].

Body záujmu sú body v obraze, ktoré majú presne definovanú pozíciu v obraze a môžu byť ľahko detekované pomocou detektorov. Každému bodu záujmu prislúcha jeden alebo viacero lokálnych príznakov, ktoré popisujú vlastnosti jeho lokálneho okolia. Z toho vyplýva, že v lokálnom okolí bodu záujmu musí byť zakódovaná významná informácia.

Lokálne príznaky sú často extrahované na základe atribútov intenzity, farby alebo textúry bodov v okolí bodu záujmu. Lokálnymi príznakmi môžu byť hrany, rohy, konce línií príp. izolované body s veľmi odlišnou farbou alebo intenzitou oproti bodom vo svojom okolí a pod.

Každý detekovaný bod záujmu by mal byť invariantný voči pootočeniu, zmene mierky, geometrickej transformácii a iným parametrom. Invariantnosť znamená, že bod záujmu resp. jemu prislúchajúce lokálne príznaky sa nemenia aplikovaním určitej triedy transformácií.

Zjednotením všetkých vyextrahovaných lokálnych príznakov pre každý bod záujmu osobitne získame vektor príznakov daného záujmového bodu. Množina všetkých vektorov príznakov tvorí priestor príznakov. Avšak nie všetky vyextrahované príznaky sú užitočné pre ďalšie spracovanie. Tu sa otvára priestor na použitie algoritmu redukcie príznakov.

Pre ďalšie spracovanie musíme nájsť lokálne príznaky vhodne reprezentovať. Na to sa používajú deskriptory. Niektoré deskriptory sú konštruované pomocou príznakov extrahovaných na základe atribútu farby, iné zase používajú príznaky extrahované na základe intenzity bodov alebo textúry v okolí, hraníc objektov v obraze a pod. Deskriptory tvoria vstupy do klasifikátorov resp. algoritmov vykonávajúcich kategorizáciu.

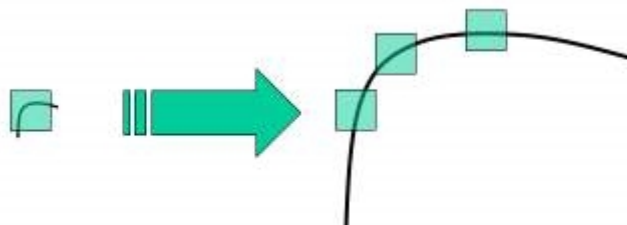
V literatúre je popísaných mnoho druhov deskriptorov a algoritmov ich extrakcie, pričom každý je vhodný na riešenie inej množiny úloh. My sa budeme venovať algoritmom SIFT a SURF na extrakciu deskriptorov.

3.3.1. Algoritmus SIFT

SIFT je skratka z anglického „scale invariant feature transform“, čo môžeme preložiť ako príznaková transformácia invariantná voči zmene mierky. Ako to napovedá názov algoritmus extrahuje deskripty, ktoré sú invariantné voči zmene mierky, voči otočeniu, zmene osvetlenia, šumom a malej zmene uhla pozorovania objektu. [2]

Algoritmus pracuje so šedo tónovým obrazom v štyroch základných krokoch:

3.3.1.1. Detekcia extrému pri rôznych mierkach (zväčšeniach) obrazu.



Obr. 9 Detekcia rohu pomocou rovnako veľkého lokálneho okolia bodu pri rôznych mierkach (zväčšeniach) obrazu [17]

Ako je vidieť na obrázku 9, nie je možné použiť rovnako veľké lokálne okolie na detekciu bodu záujmu pri rôznej mierke obrazu. V prípade zobrazenom v ľavej časti obrázku 9, by bol nájdený roh a označený za bod záujmu. V prípade zobrazenom napravo, by žiaden roh nebol nájdený, aj keď v skutočnosti sa tam nachádza, len obraz je zväčšený. Na detekciu takto zväčšeného rohu potrebujeme použiť väčšie lokálne okolie bodu. [17]

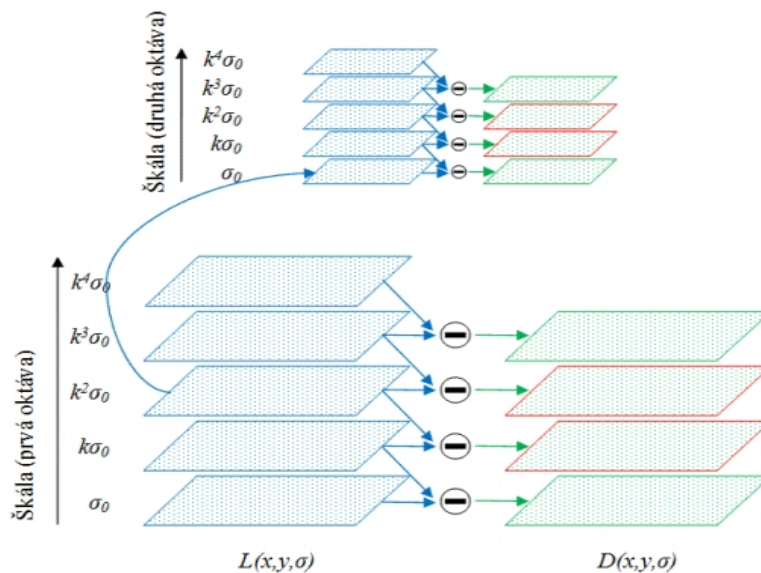
Body záujmu nájdené algoritmom SIFT zodpovedajú extrémom rozdielov Gaussových filtrov v celej množine mierok. Aby sme získali tento extrém je nutné najprv skonštruovať množinu mierok. Tá bude pozostávať z viacerých oktáv. [18] Množinu mierok môžeme definovať ako funkciu $L(x, y, \sigma)$, ktorá je tvorená konvolúciou variabilnej Gaussovej funkcie $G(x, y, \sigma)$ so vstupným obrazom $I(x, y, \sigma)$. [18] Tento vzťah vyjadruje rovnica (1). Znak „*“ predstavuje konvolúciu vstupného obrazu a Gaussovej funkcie na pozícií x, y .

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y, \sigma) \quad (1)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2)$$

Algoritmus SIFT detekuje body záujmu pomocou rozdielu Gaussových funkcií konvolvoaných s obrazom. Rozdiel Gaussových funkcií je označovaný $D(x, y, \sigma)$ a je možné ho vypočítať pomocou vzťahu (3). Tento vzťah predstavuje rozdiel Gaussových rozostrení obrazu pre mierky σ a $k\sigma$. [2]

$$G(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (3)$$



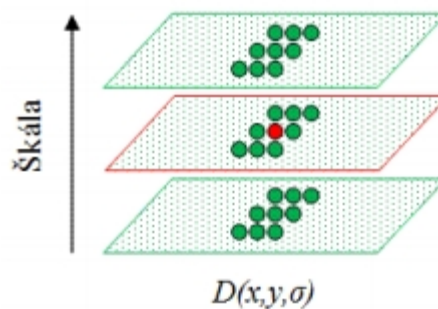
Obr. 10 Schéma zobrazuje rozostrené obrazy v rôznych mierkach a nasledovný výpočet rozdielov Gaussových rozostrení obrazu

Obr. 10 zobrazuje konštrukciu množiny mierok. Modré plochy na obrázku zodpovedajú rozostreným Gaussovým obrazom². Odčítaním vždy dvoch po sebe nasledujúcich obrazov získame rozdiel vyhladených Gaussových obrazov. Tieto rozdiely sú znázornené zelenou farbou. Tie rozdiely na ktorých je možné hľadať extrémny (body záujmu) majú okraje označené červenou farbou. Vždy hľadáme extrémny len na tých rozdielových obrazoch, ktoré majú suseda nad aj pod sebou a vždy len v rámci rovnakej oktávy. Nová oktáva sa začína konštruovať z obrazu, ktorý je vždy tretí zhora v oktáve o úroveň nižšie. V tomto obraze sa vynecháva každý druhý riadok aj stĺpec pixelov. Novovzniknutý obraz tak bude štyri krát menší ako pôvodný obraz a bude tvoriť prvý rozostrený obraz v novej oktáve so škálou σ_0 .

Rozostrené obrazy sú zoskupované v rámci oktáv. Každá oktáva obsahuje toľko rozostrených obrazov, aby sa dosiahlo zdvojnásobenie hodnoty škály σ_0 . Z toho vyplýva, že parameter k sa používa na získanie konštantného počtu rozostrených obrazov v rámci každej oktávy. Definovaním parametru k zároveň získame aj konštantný počet rozdielov rozostrených Gaussových obrazov pre každú oktávu.

² Namiesto rozostrených obrazov sa niekedy používa pojem vyhladené obrazy. Obe termíny sú ekvivalentné

Ako už bolo spomenuté vyššie v texte, body záujmu nájdené algoritmom SIFT zodpovedajú globálnemu extrému rozdielov Gaussových rozostrení obrazu v celej množine škál. Najprv hľadáme lokálny extrém v rozdieloch Gaussových rozostrení obrazov v rámci jednej oktávy. Lokálny extrém sa hľadá tak, že každý pixel rozdielu dvoch rozostrených obrazov sa porovná s ôsmimi susednými pixelmi rovnakého obrazu a z deviatimi korešpondujúcimi pixelmi v obrazoch s väčšou a menšou mierkou (v obrazoch rozdielov rozostrení nad ním a pod ním v rámci jednej oktávy). Hľadanie lokálneho extrému zobrazuje obr. 11. Nájdené lokálne extrémy sú potencionálnymi bodmy záujmu.



Obr. 11 Hľadanie lokálneho extrému v rozdieloch Gaussových rozostrení obrazu

3.3.1.2. Lokalizácia bodu záujmu.

Potenciálne body záujmu nájdené v predchádzajúcom kroku prechádzajú ďalším spracovaním. Je potrebné spresniť ich polohu a škálu. V tomto kroku sa odstraňujú body s nízkym kontrastom a body z okolia hrán, keďže rozdiel Gaussových funkcií je pozdĺž hrán síce veľký, ale samotné hrany sú veľmi háklivé na šum. V ďalších krokoch už nasleduje tvorba deskriptoru.

3.3.1.3. Priradenie orientácie

Pridanie orientácie slúži na získanie invariance voči pootočeniu. Každému nájdenému bodu záujmu sa priradí jeho orientácia na základe okolia v rozostrenom Gaussovom obraze (nie v ich rozdieloch ako v bode 1). Mierka nájdeného bodu záujmu nám posluží na zvolenie správneho rozostreného obrazu. Aby sme mohli vypočítať orientáciu bodu záujmu pre každý obrazový bod každého rozostreného obrazu, musíme vypočítať veľkosť gradientu $m(x, y, \sigma)$ a orientáciu $\theta(x, y, \sigma)$.

$$m(x, y, \sigma) = \sqrt{\frac{1}{2} (L(x+1, y, \sigma) - L(x-1, y, \sigma))^2 + \frac{1}{2} (L(x, y+1, \sigma) - L(x, y-1, \sigma))^2} \quad (4)$$

$$\theta(x, y, \sigma) = \tan^{-1} \left(\frac{L(x, y+1, \sigma) - L(x, y-1, \sigma)}{L(x+1, y, \sigma) - L(x-1, y, \sigma)} \right) \quad (5)$$

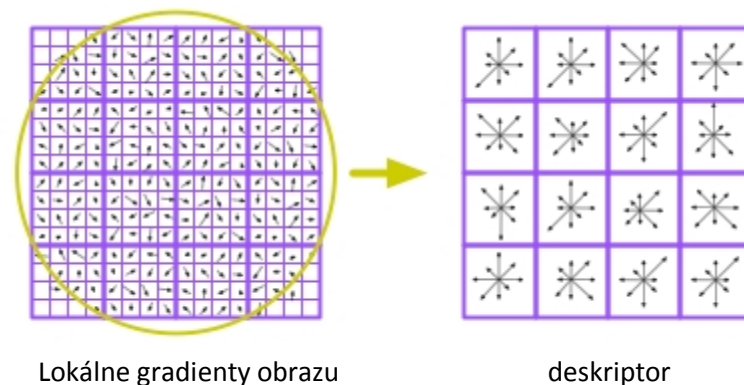
Tieto hodnoty nám slúžia na zostrojenie histogramu orientácií okolia nájdeného bodu záujmu. Okolie bodu záujmu má kruhový tvar a jeho priemer bude 4,5 násobkom mierky. Samotný

histogram je tvorený 36 indexmi. Orientácie každého pixelu vypočítané podľa (5) z tohto okolia sú násobené veľkosťou gradientu (4) a tiež sú vážené Gaussovým oknom σ . Veľkosť Gaussového okna je jeden a pol násobkom mierky bodu záujmu. Takto vypočítaná hodnota sa pripočíta k hodnote, ktorá je reprezentovaná príslušným indexom histogramu. Samotný index histogramu je určený iba orientáciou bodu θ . Vrchol zostrojeného histogramu predstavuje dominantnú orientáciu bodu záujmu. Pre všetky ostatné orientácie histogramu, ktoré sú väčšie ako 80% maximálnej hodnoty histogramu sú vytvorené osobitné body záujmu s prislúchajúcimi orientáciami. Vytváranie nových bodov záujmu zvyšuje presnosť vo fáze detekcie.

Môže nastať situácia, kedy histogram obsahuje viacero výherných vrcholov. V takom prípade sa vypočíta aritmetický priemer hodnoty každého z výherného vrcholu histogramu a dvoch ich najbližších susedov. Tento priemer sa pripočíta k prislúchajúcemu vrcholu. Potom výherné vrcholy už nebudú mať rovnaké hodnoty a my budeme môcť určiť orientáciu. [18]

3.3.1.4. Deskriptor bodu záujmu

Deskriptor sa vytvára z rovnakého obrazu, z ktorého sa počítala orientácia v predchádzajúcom kroku. Na konštrukciu SIFT deskriptorov sa používajú orientácie lokálnych gradientov otočené podľa orientácie bodu záujmu. Tieto gradienty sú vypočítané z okolia bodu záujmu veľkosti 16x16 pixelov. Okolie sa delí na 16 subokolí, každé o veľkosti 4x4 pixely. Pre každé subokolie sa vypočíta histogram obsahujúci osem indexov orientácií. Spolu je tak skonštruovaných 16 histogramov, ktorých hodnoty sú zoskupené v 128-rozmernom SIFT deskriptore. [17] Obr. 12 zobrazuje generovanie SIFT deskriptoru.



Obr. 12 Postup generovania SIFT deskriptoru

3.3.1.5. Porovnávanie bodov záujmu (fáza detekcie)

Objekt v obraze môže byť klasifikovaný do známej triedy pomocou metódy k-najbližších susedov. Počet nájdených bodov záujmu môže byť rôzny. Závisí od veľkosti obrazu, detailnosti obrazu a mnohých iných kritérií. Nie je výnimočné, ak je nájdených aj 2000 rôznych bodov záujmu.

3.3.2. Algoritmus SURF

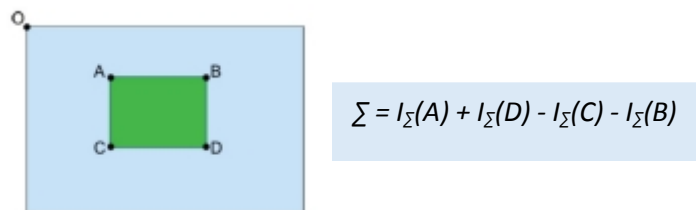
SURF je skratka z anglického Speed-Up Robust Features, čo sa dá voľne preložiť ako rýchly robustný detektor a deskriptor, ktorý má v počítačovom videní pomerne široké využitie napríklad na účely rozpoznávania objektov, 3D rekonštrukciu a pod. Čiastočne je inšpirovaný algoritmom SIFT. Pre urýchlenie výpočtu používa integrálny obraz a Hessové matice pre detekciu bodov záujmu

3.3.2.1. Integrálny obraz

Integrálny obraz je obraz, ktorý v každom svojom bode obsahuje súčet všetkých hodnôt obrazových bodov od začiatku obrazu až do daného bodu. [19] Matematický to môžeme vyjadriť vzťahom (6).

$$I_{\Sigma}(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad (6)$$

Výhodou integrálneho obrazu je, že na výpočet sumy intenzít obrazových bodov v ľubovoľnej veľkej oblasti postačujú štyri matematické operácie. Vďaka integrálnemu obrazu je výpočtový čas nezávislý od veľkosti oblasti z ktorej počítame sumu.



Obr. 13 Ilustrácia výpočtu sumy intenzít v integrálnom obraze

3.3.2.2. Hessián a Hessové matice

Hessová matica je štvorcová matica, ktorej prvky sú druhé parciálne derivácie skalárnej funkcie. Hessián je skrátenejší názov pre determinant Hessovej matice. Hessova matica pre skalárnu funkciu s dvoma premennými $f(x, y)$, bude mať veľkosť 2×2 a pre výpočet jej determinantu platí nasledujúci vzťah.

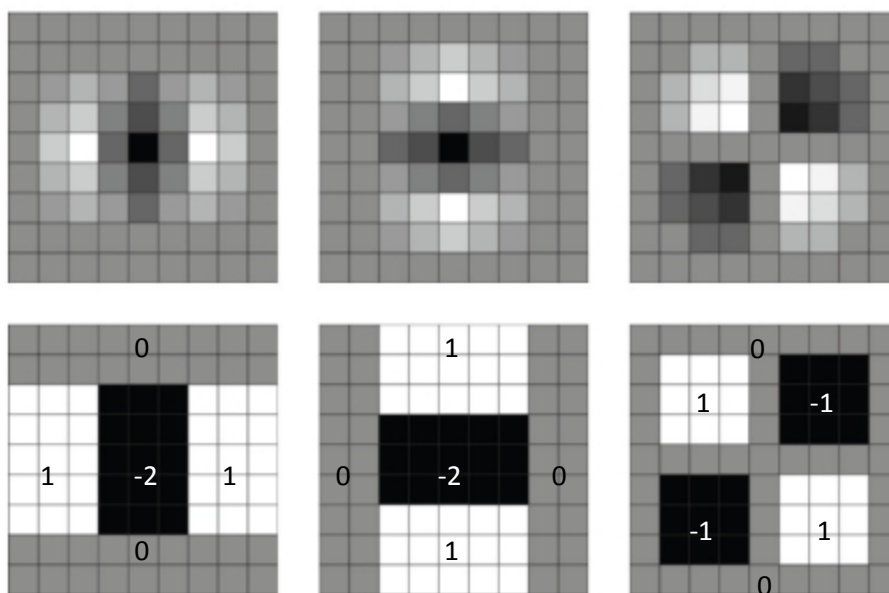
$$\mathcal{H}(x, y) = \det \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} = \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 \quad (7)$$

Hessova matica používaná ako detektor bodov záujmu v obraze je definovaná nasledovne. Pre daný bod x so súradnicami x, y v obraze I má Hessova matica $\mathcal{H}(x, \sigma)$ v bode x a s mierkou σ tvar:

$$\mathcal{H} = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{yx}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (8)$$

Kde $L_{xx}(x, \sigma)$ je konvolúcia druhej parciálnej derivácie $\frac{\partial^2}{\partial x^2} g(\sigma)$ v bode x . Podobný vzťah platí aj pre $L_{xy}(x, \sigma)$ a $L_{yy}(x, \sigma)$. [2] Zmenou parametra σ je možné meniť zväčšenie, na ktorom je počítaný determinant. My hľadáme bod, ktorý má maximálny determinant Hessovej matice pri rôznom zväčšení (tzv. Gaussové jadro).

Pre zjednodušenie a urýchlenie výpočtu druhých derivácií sa používa aproximácia gaussových jadier pomocou 2D filtrov s celočíselnými koeficientami. Obr. 14 zobrazuje túto aproximáciu. V prvom riadku sú diskretizované gaussové jadrá L_{xx} , L_{yy} a L_{xy} . V druhom riadku sú zobrazené aproximácie pomocou 2D filtrov D_{xx} , D_{xy} , D_{yy} . Čísla predstavujú hodnoty koeficientov, ktoré sú aplikované na príslušné oblasti obrazových bodov v Gaussových jadrách.



Obr. 14 Aproximácia Gaussových jadier pomocou 2D filtrov [19]

Najmenšia veľkosť 2D filtra je mriežka veľkosti 9x9. Aproximácia gaussových jadier pomocou 2D filtrov skresľuje hodnotu počítaného Hessiánu. Aby bola aproximácia použiteľná, skreslenie vznikajúce použitím 2D filtrov musí byť kompenzované pomocou vzťahu (9).

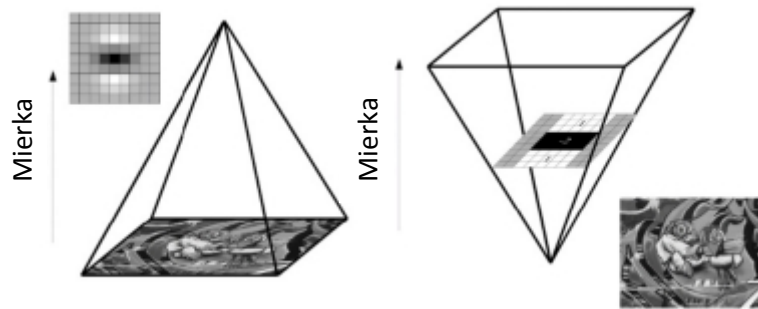
$$\mathcal{H} = D_{xx}D_{yy} - (0,9D_{xy})^2 \quad (9)$$

Aproximovaný Hessov determinant reprezentuje odozvu v obraze I na pozícií x . Tieto odozvy sú detekované naprieč rôznymi zväčšeniami.

3.3.2.3. Invariancia voči zmene mierky

Pre získanie invariance voči zmene mierky musíme hľadať body záujmu v rôznych mierkach (zväčšeniach) obrazu. Konštrukcia mierok a oktáv nám pôvodný obraz postupne znižuje. Je aplikovaný rovnaký princíp konštrukcie ako v algoritme SIFT.

Vďaka použitiu 2D filtrov a integrálnych obrazov, je možné aplikovať 2D filtre s rôznou veľkosťou priamo na pôvodný obraz. [2] Na základe toho, analýza obrazu jednej mierky znamená postupné zväčšovanie 2D filtrov a ich aplikovanie na tento obraz. Za základnú mierku je považovaný 2D filter veľkosti 9x9. Ďalšie veľkosti sú 15x15, 21x21, 27x27 atď.



Obr. 15 Zmenšovanie mierky pôvodného obrazu a zväčšovanie 2D filtru [19]

Každá oktáva reprezentuje sériu odoziev získaných porovnaním rovnakého vstupného obrazu (jednej mierky) s maskami zväčšujúcej sa veľkosti. Každá oktáva pozostáva z konštantného počtu úrovní mierok 2D filtru. [2]

3.3.2.4. Lokalizácia bodu záujmu

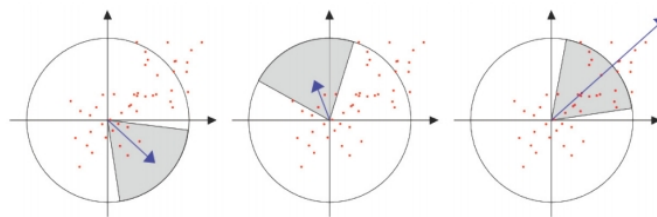
Pomocou výpočtov uvedených v predchádzajúcom texte sme získali odozvy (hodnoty Hessiánov) pre každý bod obrazu v rôznych zväčšeniach. Lokalizácia bodu záujmu je možné rozložiť do niekoľkých subkrokov. Prvým z nich je odstránenie tých Hessiánov, ktoré nepresahujú požadovaný prah. Prah sa určuje s ohľadom na konkrétnu aplikáciu. Po odstránení sa vykonáva hľadanie lokálnych maxím. Toto hľadanie prebieha rovnako ako pri algoritme SIFT (viď. podkapitulu 3.3.1.1, obr. 11). Posledným krokom je interpolácia pozícií maxím Hessoého determinantu v rámci jedného zväčšenia a v rámci všetkých zväčšení. [19]

3.3.2.5. Deskriptor bodu záujmu

Používaný deskriptor je 64 rozmerný vektor hodnôt, ktorý popisuje okolie lokalizovaného bodu záujmu. Veľkosť popisovaného okolia závisí od mierky (zväčšenia), na ktorej bol bod záujmu detekovaný. Vlastné generovanie deskriptoru sa skladá z dvoch krokov.

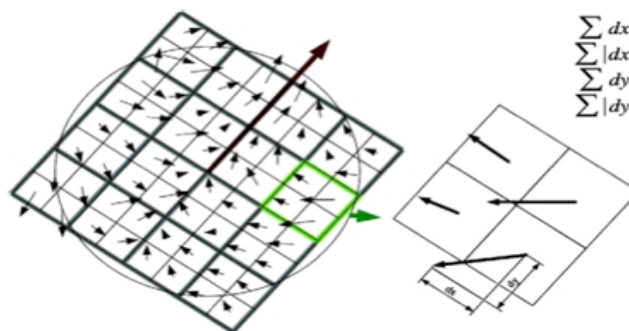
Najprv sa určuje orientácia okolia detekovaného bodu záujmu. Tento proces zabezpečuje invarianciu SURF deskriptorov voči rotácií. Na určenie orientácie bodu záujmu sa používa výpočet odoziev Haarových vlnkových filtrov (z angl. Haar Wavelets) v smeroch osí x a y . Odozvy Haarových vlnkových filtrov sa počítajú z kruhového okolia bodu záujmu s polomerom $6s$, pričom s je mierka, v ktorej bol bod záujmu detekovaný. Vypočítané odozvy sú ďalej vážené Gaussianom, centrovaným na bod záujmu. [2] V každom bode kruhového okolia tak získame váhovanú odozvu

Haarových vlnkových filtrov v smere osi x a y . Ak tieto váhované odozvy prehlásime za vektor, potom kruhové okolie bude vytvárať 2-rozmerný priestor. V ňom prvý rozmer predstavuje silu odozvy v horizontálnom smere (v smere osi x) a druhý rozmer predstavuje silu odozvy vo vertikálnom smere (v smere osi y). Celková orientácia je potom vypočítaná nasledovne. Necháme rotovať posuvné okno veľkosti $\pi/3$ s krokom 10° okolo kruhového okolia, pričom pre každú polohu posuvného okna počítame sumy všetkých horizontálnych odoziev a všetkých vertikálnych odoziev v danom okne. Výsledkom sčítania je lokálny vektor orientácie. Celková orientácia je totožná s uhlom vektoru, ktorý má maximálnu hodnotu zo všetkých lokálnych vektorov orientácie. [19]



Obr. 16 Princíp priraďovania dominantného smeru orientácie bodu záujmu [19]

Po získaní dominantnej orientácie bodu záujmu môžeme pristúpiť ku samotnej konštrukcii deskriptoru. Deskriptor sa generuje zo štvorcového okolia bodu záujmu veľkosti hrany $20s$ a orientácie totožnej s dominantnou orientáciou bodu záujmu. Toto okolie je delené na 16 rovnako veľkých štvorcových subokolí. Každé zo subokolia musí obsahovať 25 rovnomerne rozložených vzorkovacích bodov. Pre každý zo vzorkovacích bodov je počítaná odozva Haarových vlnkových filtrov v smere d_x a d_y . Je potrebné podotknúť, že smery d_x a d_y nezodpovedajú smerom osi x a y , ale d_x zodpovedá otočeniu deskriptoru a d_y zodpovedá smeru kolmému na d_x . Každé z týchto 16 subokolí je popísané 4 rozmerným vektorom $v = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|]$. Deskriptor je potom tvorený spojením vektorov všetkých 16 oblastí. [19] Celý postup je ilustrovaný na obr. 17.



Obr. 17 Ilustrácia konštrukcie deskriptoru SURF

3.4. Klasifikácia a kategorizácia

V tejto podkapitole uvedieme rozdiel medzi klasifikáciou a kategorizáciou, ponúkame delenie klasifikačných metód a bližšie vysvetlíme neurónové siete. Vzhľadom na ciele práce ponúkame úvod do adaptívnej rezonančnej teórie a ART sietí. Podrobne charakterizujeme neurónové siete typu MF Artmap.

3.4.1. Klasifikácia verzus kategorizácia

Rozdiel medzi klasifikáciou a kategorizáciou je veľmi malý a pre mnohých ľudí častokrát nejasný. V tejto krátkej podkapitole uvádzame rozdiel.

Podľa definície Elin K. Jacoba klasifikácia priraduje každej entite jednu triedu v rámci systému vzájomne neprekrývajúcich sa tried. [4] Pre porovnanie podľa definície prof. Paraliča, klasifikácia modeluje a predpovedá nominálne atribúty (triedy). [4] Triedy sú zvyčajne dobre pomenované, vopred pevne definované, nemenné a neprekrývajúce sa. Ak to zhrnieme, tak pri klasifikácii zatriedujeme objekty alebo javy do známych tried na základe atribútov klasifikovaných objektov.

Kategorizácia podľa Elin K. Jacoba je proces rozdeľovania sveta (univerza) do skupín entít, pričom entity v rámci skupiny sú si navzájom podobné. [4] Pierre Baldi definuje kategorizáciu ako proces zoskupovania entít do pevne určených tried alebo kategórií. [4] Kategórie nemusia byť dobre pomenované. Jedna entita môže byť kategorizovaná do viacerých tried. Jednoducho povedané kategorizácia je proces, v ktorom vytvárame množinu kategórií na základe známych entít.

3.4.2. Metódy klasifikácie objektov

Základom kvalitnej klasifikácie a kategorizácie je dobrá reprezentácia [20] znalostí. V podkapitole 3.3 sme sa venovali práci s príznakmi s cieľom získania presného popisu obrazu. Príznačky samé o sebe nie sú reprezentáciou znalostí, ale sú jej dôležitou časťou. [16] Príznačky sú organizované do vektorov príznakov a tie tvoria vstupy do klasifikačných a kategorizačných algoritmov. Metódy klasifikácie resp. kategorizácie pracujúce s príznakmi sa nazývajú príznakové (štatistické) metódy. Tu môžeme zaradiť rôzne druhy klasifikátorov, či už deterministické, stochastické (Bayesov klasifikátor) alebo neurónové siete.

Druhá možnosť reprezentácie znalostí je reprezentácia pomocou primitív. Znalostí sú reprezentované symbolicky. Pracuje s formálnymi gramatikami, produkčnými pravidlami, sémantickými sieťami, fuzzy logikou a pod.

Uvedené delenie klasifikačných metód je v literatúre uvádzane najčastejšie. Existujú aj iné delenia klasifikačných a kategorizačných metód, ktorým sa však nebudeme venovať.

3.4.3. Neurónové siete

Nasledujúci text vychádza z [20]. Neurónová sieť je biologický inšpirovaný výpočtový model, ktorý vznikol ako výsledok snahy simulovať ľudský mozog. Komunita vedcov pôsobiacich v oblasti neurónových sietí definuje tento výpočtový model ako masívne paralelný procesor, ktorý má sklon k uchovávaniu experimentálnych znalostí a ich ďalšiemu využívaniu. Neurónová sieť napodobňuje ľudský mozog v dvoch aspektoch, a to že znalostí získava v procese učenia a na ukladanie znalostí využíva medzi neurónové spojenia (tzv. synapsie).

Snáď najdôležitejšou vlastnosťou neurónových sietí je ich schopnosť aproximovať ľubovoľnú funkciu. Na základe tejto vlastnosti hovoríme, že sú univerzálnym aproximátorom funkcie. V praxi to znamená, že neurónová sieť dokáže nájsť neznámu funkciu medzi ponúknutými známymi vstupmi a k nim priradenými očakávanými výstupmi. Aproximovaná funkcia môže byť diskriminačnou funkciou, ktorá jednoznačne oddeľuje klasifikačné triedy. Teda neurónové siete sú okrem iného vhodne použiteľné pri úlohách klasifikácie.

Proces aproximácie neznámej funkcie sa nazýva učenie. Počas učenia sa adaptujú parametre neurónovej siete (synaptické váhy), ktoré slúžia na uloženie znalostí. Po ukončení učenia sieť prechádza do fázy života, kde sú získané znalosti využívané na riešenie nejakého problému. Napríklad na klasifikáciu, predikciu, optimalizáciu a pod. Tu je potrebné zdôrazniť, že neurónová sieť vo fáze života môže úspešne spracovávať aj také vstupné vzorky, ktoré jej neboli poskytnuté v procese učenia. Tomu hovoríme generalizácia.

Existujú dva základné druhy učenia neurónových sietí, a to kontrolované učenie a nekontrolované učenie. Kontrolovanému učeniu sa tiež hovorí aj učenie s učiteľom. V tomto prípade v procese učenia privádzame na vstup neurónovej siete trérovacie vzorky spolu s očakávaným výstupom z neurónovej siete. Sieť trénujeme dovtedy, kým nedosiahneme požadovaný výstup pre všetky trérovacie vzorky. Pri nekontrolovanom učení nepoznáme požadovaný výstup z neurónovej siete. Vstupné vzorky rozdeľujeme do zhlukov najčastejšie na základe podobností. Učenie končí, ak je zmena synaptických váh v čase t a v čase $t+1$ blízka nule.

Aplikačná oblasť neurónových sietí je obrovská. Pomocou nich je možné riešiť široké spektrum rôznych vedeckých a technických problémov. Okrem už spomenutej klasifikácie sú obzvlášť vhodne využiteľné na riešenie problémov predikcie, aproximácie funkcií, transformácií signálov a pod.

3.4.4. Neurónové siete typu ART

Ako sme spomínali v kapitole 3.4.3, neurónové siete vznikli zo snahy simulovať ľudský mozog. Jednou zo základných črt ľudského mozgu je schopnosť učiť sa nové veci bez toho, aby boli zabudnuté skôr naučené poznatky. V mnohých typoch neurónových sietí táto schopnosť chýba. Ako príklad uvidíme doprednú neurónovú sieť tréňovanú učiacim algoritmom backpropagation. Nech táto sieť je natréňovaná pomocou piatich tréňovacích vzoriek. Ak neskôr chceme dotréňovať novú vzorku, musíme celú sieť pretréňovať. Neurónové siete typu ART odstraňujú tento nedostatok a to tak, že umožňujú prispôbovať svoju topológiu na základe poskytovaných vstupných vzoriek.

ART je skratka pre adaptívnu rezonančnú teóriu (z anglického Adaptive Resonance Theory). Je to teória, ktorá navrhuje riešenie problému stability a plasticity vznikajúceho pri návrhu inteligentných systémov, schopných autonómnej adaptácie v reálnom čase na neočakávané zmeny v ich svete. [21] Problém stability a plasticity je možné veľmi zjednodušene charakterizovať ako problém učenia sa nových znalostí umelými inteligentnými systémami (plasticita) bez zabúdania už naučených znalostí (stabilita). Riešia sa mnohé otázky ako napríklad spôsob prepínania medzi stabilným a plastickým módom, ako uchrániť už naučené znalosti počas učenia nových znalostí a pod.

Neurónové siete typu ART sú jedným z možných riešení uvedeného problému. Rezonancia v úzko súvisí so spôsobom učenia sa neurónovej siete. Vstupná vzorka osciluje vo forme výstupných hodnôt medzi dvoma vrstvami siete až pokiaľ sa nedosiahne rezonancia. V tomto okamihu nastáva učenie, čiže adaptácia synaptických váh. Existujú dva spôsoby vzniku rezonancie. Prvý spôsob vzniku rezonancie sa objavuje ak sieť spracováva rovnakú alebo veľmi podobnú vstupnú vzorku, akú spracovávala v minulosti. Vtedy nastáva rezonancia okamžite. V druhom prípade je spracovaná vstupná vzorka výrazne odlišná od všetkých doposiaľ spracovaných vzoriek. V tomto prípade sa spúšťa porovnávanie všetkých doposiaľ naučených kódov a porovnáva sa ich podobnosť s práve spracovávanou vzorkou. Ak pre niektorý naučený kód je prekonaná definovaná prahová hodnota potom práve spracovávaná vzorka je priradená k triede reprezentovanej daným kódom. Pričom prahová hodnota je hodnota, ktorá definuje minimálnu prípustnú podobnosť vstupnej vzorky a daného kódu. Ak pre žiaden doposiaľ naučený kód nie je prekonaná prahová hodnota, potom sa vytvorí nová trieda identická s práve spracovávanou vzorkou. [22]

Veľkou výhodou ART sietí je ich schopnosť učiť sa bez zabúdania skôr naučených znalostí. Teda nerozlišujú medzi fázou tréňovania a fázou života neurónových sietí. Naopak, nevýhodami sú vysoké pamäťové nároky, citlivosť na šum, citlivosť na nastavenie učiacich parametrov a na

postupnosť vstupných vzoriek. Rôzna postupnosť vstupných vzoriek môže viesť k vytvoreniu rôznych zhlukov. [21]

ART neurónové siete môžu byť tréované pomocou kontrolovaného aj nekontrolovaného učenia. Základnou ART sieťou tréovanou pomocou nekontrolovaného učenia je sieť ART1. Časom vznikli rôzne derivácie a rozšírenia tejto siete ako napríklad ART2, ART2a, ART3, fuzzy ART a iné. Východiskovou sieťou neurónových ART sietí tréovaných pomocou kontrolovaného učenia je sieť ARTMAP. Podobne ako pri nekontrolovanom učení ART sietí, aj v tejto skupine vznikli mnohé upravené ARTMAP siete. Napríklad fuzzy ARTMAP, distribuovaný ARTMAP, Gaussovský ARTMAP, MF ARTMAP a mnohé ďalšie.

Ako sme už viackrát avizovali, vzhľadom na ciele tejto práce a obmedzený rozsah, našu pozornosť sústredíme len na neurónovú sieť MF ARTMAP.

3.4.5. MF ARTMAP

Názov MF ARTMAP pochádza z anglického Membership Function ARTMAP. Už názov napovedá, že táto neurónová sieť kombinuje teóriu fuzzy³ logiky s teóriou adaptívnej rezonančnej teórie. Motiváciou jej vzniku je fakt, že neurónové siete ARTMAP dosahujú obmedzenú presnosť klasifikácie, keďže nie vždy je možné jednoznačne určiť triedu vstupnej vzorky. MF ARTMAP vďaka využívaniu fuzzy množín umožňuje vypočítať stupeň príslušnosti vstupného vektora hodnôt (tzv. vstupnej vzorky) ku každej z tried. Výstup zo siete teda nie je explicitne vyjadrená trieda vstupnej vzorky, ale vektor stupňov príslušnosti vstupnej vzorky ku každej z tried.

3.4.5.1. Teória fuzzy množín v MF ARTMAP sieťach

Táto časť vychádza z [21]. Teória fuzzy množín v MF ARTMAP sietí sa využíva na reprezentáciu znalostí. Predpokladajme, že n -diemnzionálny vstupný priestor predstavuje univerzum, v ktorom sú definované fuzzy množiny. Vstupné vzorky sú zoskupované do fuzzy zhlukov. Z toho vyplýva, že je možné vypočítať hodnotu funkcie príslušnosti každej vstupnej vzorky ku každému fuzzy zhluoku. Fuzzy zhluky reprezentujú konkrétne klasifikačné triedy. Každá trieda môže byť reprezentovaná jedným alebo viacerými fuzzy zhlukmi.

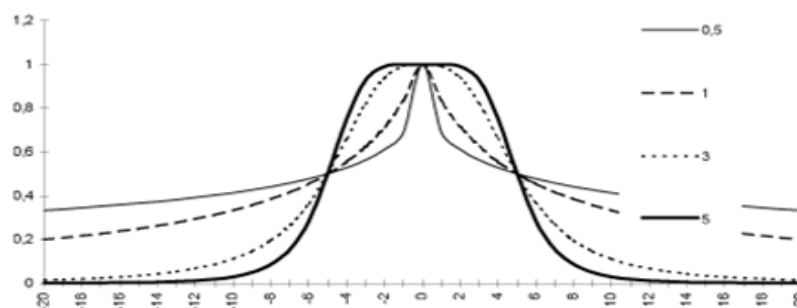
Každý fuzzy zhluk je v MF ARTMAPe definovaný pomocou fuzzy množín. Môžeme ho popísať pomocou fuzzy relácie s parametrami. V procese učenia sa adaptujú parametre fuzzy relácií. To spôsobí zmenu tvaru funkcií príslušnosti fuzzy zhlukov a to tak, aby čo najlepšie pokrývali vstupné vzorky patriace do tried nimi reprezentovanými. Fuzzy relácia je v MF ARTMAP neurónových sieťach definovaná podľa vzťahu (10).

³ Pochopenie teórie neurónovej siete MF ARTMAP predpokladá základné poznatky z fuzzy logiky, ktoré je možné získať vo voľne dostupnej elektronickej literatúre.

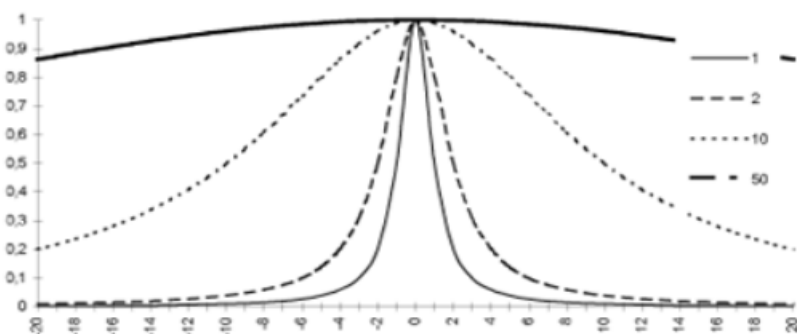
$$A(\bar{x}_i) = \int_U \frac{1}{1 + \left| \frac{\bar{x}_{s_k} - \bar{x}_i}{E_k} \right|^{F_k}} / \bar{x}_i \quad (10)$$

Kde m je počet dimenzií vstupného priestoru, $j=1, \dots, m$ predstavuje index dimenzie vstupného priestoru, $U=X_1 \times \dots \times X_m$ je označenie vstupného priestoru (univerza), n je počet vstupných vzoriek v tréningovej množine, i je index aktuálnej tréningovej vzorky, p je index doteraz vytvorených fuzzy zhlukov, $k=1, \dots, p$ je index aktuálneho fuzzy zhluku, $A(\bar{x}_i)$ zodpovedá príslušnosti vzorky \bar{x}_i k fuzzy zhluku k . Vektor \bar{x}_i je ľubovoľný bod v n -dimenzionálnom vstupnom priestore a na záver vektory x_{s_k} , E_k a F_k sú parametre vytvoreného fuzzy zhluku. Vektor x_{s_k} predstavuje stred daného fuzzy zhluku. [21]

Obrázky 18 a 19 zobrazujú vplyv vektorov E_k a F_k na tvar funkcie príslušnosti fuzzy zhluku. Obrázky sú vykreslené v jednorozmernom priestore, z dôvodu zachovania prehľadnosti a jednoduchosti ilustrácie. Zvyšovanie parametra E pre danú dimenziu spôsobí zmenšenie špicatosti funkcie príslušnosti fuzzy zhluku v tejto dimenzii. Zväčšovaním parametra F pre danú dimenziu spôsobíme rozšírenie fuzzy zhluku v tejto dimenzii.



Obr. 18 Vplyv parametra F na tvar funkcie príslušnosti fuzzy zhluku v jednorozmernom priestore [21]



Obr. 19 Vplyv parametra E na tvar funkcie príslušnosti fuzzy zhluku v jednorozmernom priestore [21]

Neurónová sieť MF ARTMAP počíta v porovnávacej vrstve hodnoty čiastkových funkcií príslušností pre každú dimenziu každého fuzzy zhluku (11). V rozpoznávacej vrstve zase sumarizuje tieto čiastkové hodnoty funkcií príslušnosti a počíta celkovú hodnotu funkcie príslušností vstupnej vzorky k jednotlivým fuzzy zhlukom (12). Spojenie týchto dvoch čiastkových výpočtov je ekvivalentné výpočtu hodnoty funkcie príslušností vstupnej vzorky ku fuzzy zhlukom podľa (10).

$$f_j(x_{i,j}) = \frac{1}{1 + \left(\frac{x_{s_{k,j}} - x_{i,j}}{E_{k,j}}\right)} \quad (11)$$

$$A(\bar{x}) = \frac{1}{1 + \left| \frac{1}{\sum_{j=1}^m f_j(x_{i,j})} - 1 \right|} \quad (12)$$

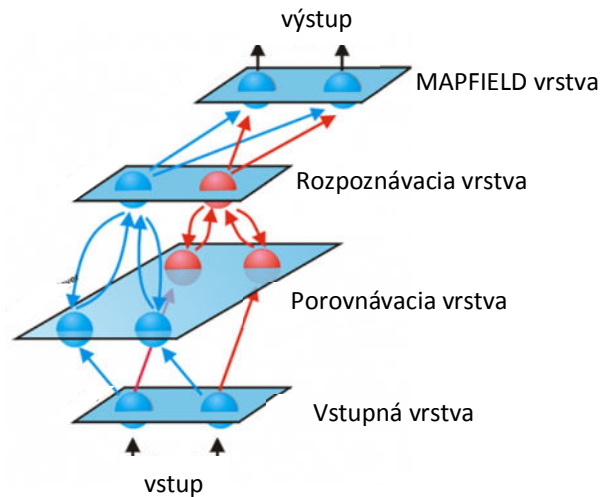
Ako bolo spomenuté skôr v tejto kapitole, trieda môže byť reprezentovaná viacerými fuzzy zhlukmi. Takáto trieda je potom definovaná ako zjednotenie tých fuzzy zhlukov, ktoré ju reprezentujú. Hodnota funkcie príslušností vstupnej vzorky ku fuzzy triede je totožná s maximálnou hodnotou funkcie príslušností spomedzi všetkých zhlukov tvoriacich danú triedu.

3.4.5.2. Topológia siete

Na obr. 20 je znázornená topológia MF ARTMAP siete. Vychádza z topológie ARTMAP sietí. Je tvorená zo štyroch vrstiev neurónov:

1. **Vstupná vrstva** slúži na priame mapovanie vstupnej vzorky na porovnávaciu vrstvu. Voliteľne sa tu môže vykonávať normalizácia vstupného vektora. Počet neurónov vo vstupnej vrstve je totožný s dimenzionalitou vstupného vektora.
2. **Porovnávacia vrstva** vykonáva výpočet čiastkovej funkcie príslušností vstupu ku každému fuzzy zhlukom podľa vzťahu (11). Neuróny v tejto vrstve sú organizované do dvojrozmernej mriežky, pričom jeden rozmer mriežky je totožný s počtom neurónov vo vstupnej vrstve a druhý rozmer je rovnaký ako počet neurónov v rozpoznávacej vrstve. Táto vrstva obsahuje vždy jeden neurón pre každú dimenziu vstupného vektora a pre každý fuzzy zhluk. Pridanie nového fuzzy zhluku (neurónu do rozpoznávacej vrstvy), spôsobí pridanie jedného radu neurónov v tejto vrstve.
3. **Rozpoznávacia vrstva** slúži na výpočet celkovej funkcie príslušnosti ku každému fuzzy zhlukom podľa vzťahu (12). Využíva čiastočné hodnoty príslušnosti vypočítané v porovnávacej vrstve. Vyberá víťazný zhluk t.j. zhluk, ku ktorému má vstupný vektor najvyššiu funkciu príslušnosti. V tejto vrstve každý neurón predstavuje jeden fuzzy zhluk.
4. **MAPFIELD** je vrstva neurónov, kde každý neurón zodpovedá jednej triede. Tu sa určuje výsledná hodnota funkcie príslušnosti vstupnej vzorky ku fuzzy triede, keďže jedna trieda môže byť tvorená viacerými fuzzy zhlukmi.

Na uloženie znalostí sa používajú synapsie medzi rozpoznávacou vrstvou a porovnávacou vrstvou. Tu sú zakódované parametre X , E a F pre všetky dimenzie všetkých fuzzy zhlukov. Znalostí sa ukladajú aj do synapsií medzi rozpoznávacou vrstvou a MAPFIELD vrstvou, kde sa ukladajú znalosti o tom, z ktorých fuzzy zhlukov sú tvorené jednotlivé triedy.



Obr. 20 Topológia MF ARTMAP siete

3.4.5.3. Učenie MF ARTMAP sietí

MF ARTMAP siete umožňujú inkrementálne učenie. Teda umožňujú učenie nových znalostí kedykoľvek, a to bez zabúdania skôr získaných poznatkov. Navyše fáza učenia a fáza života týchto sietí sa môže ľubovoľne prelínať.

Pri učení dochádza k dvom druhom adaptácie, ktoré spolu kooperujú. Jednou z možných adaptácií je adaptácia štruktúry siete. Táto adaptácia pridávaním nových neurónov do porovnávacej a rozpoznávacej vrstvy definuje nové fuzzy zhluky. Druhá možnosť adaptácie je adaptácia parametrov fuzzy zhlukov tak, aby definované zhluky čo najpresnejšie popisovali vstupné vzorky.

Vstupnými parametrami algoritmu učenia sú parametre fuzzy zhlukov E , F a prah pre rozpoznávaciu vrstvu. Celý algoritmus učenia prebieha podľa nasledujúcich bodov:

1. Na vstup sa privádza trénovacia vzorka, ktorá sa tu môže normalizovať. Odtiaľ je presunutá do porovnávacej vrstvy. Ak sme už natrénovali všetky vzorky v trénovacej množine, algoritmus končí.
2. V porovnávacej vrstve sa počítajú hodnoty čiastkových funkcií príslušností pre každú dimenziu a každý fuzzy zhluk.
3. Následne postupujeme do rozpoznávacej vrstvy, kde sa počíta celková hodnota funkcií príslušnosti vstupnej vzorky ku fuzzy zhlukov. Výpočet používa čiastočné hodnoty funkcií príslušností vypočítané v porovnávacej vrstve. Tu sa porovnáva celková vypočítaná

hodnota s prahom. Ak je niektorá celková hodnota nižšia ako prah, znamená to, že fuzzy zhuk reprezentovaný neurónom je nevyhovujúci. Tento neurón sa zmrazí. V prípade, že sú zmrazené všetky neuróny v rozpoznávacej vrstve, ani jeden fuzzy zhuk nie je vhodný. Vtedy pokračujeme bodom 4, ináč prechádzame na bod 5.

4. Definujeme nový fuzzy zhuk, ktorého stred je totožný so vstupnou vzorkou. Vstupné parametre E a F sú aplikované ako počiatočné parametre a parametre nového fuzzy zhluku. Nasleduje bod 1.
5. Z neurónov rozpoznávacej vrstvy, ktoré nie sú zmrazené vyberieme ten, ktorý reprezentuje fuzzy zhuk s maximálnou hodnotou funkcie príslušnosti. Tento fuzzy zhuk sa označí za výherný a následne sa vstupnej vzorke priradí trieda, ktorú víťazný fuzzy zhuk reprezentuje.
6. Ak zvolená trieda je totožná s triedou vstupnej vzorky, potom pokračujeme bodom 4.
7. Ak zvolená trieda je totožná s triedou vstupnej vzorky, potom nastáva adaptácia parametrov víťazného fuzzy zhluku. Adaptujú sa parametre každej dimenzie víťazného fuzzy zhluku uložené v synasiách medzi porovnávacou a rozpoznávacou vrstvou, a to podľa nasledujúcich vzťahov.

$$X_S^N = X_S^O - \frac{1}{q}(X_S^O - X) \quad (13)$$

$$E^N = E^O - \frac{1}{q}(E^O - (X_S^O - X)^2) \quad (14)$$

kde q je počet vzoriek v danom fuzzy zhluku, O je stará hodnota parametra, N je nová hodnota parametra a X je vstupná vzorka.

Učenie MF ARTMAP sietí je možné urýchliť zmenou prahu v rozpoznávacej vrstve prípadne pridaním nového prahu do porovnávacej vrstvy. Ten zabezpečí zmrazenie neurónu v prípade, že aspoň jedna čiastková hodnota funkcie príslušností daného fuzzy zhluku nedosiahla prahovú hodnotu.

4. Návrh systému pre rozpoznávanie objektov z obrazu v cloud robotike

Hlavným cieľom predkladanej práce je návrh a implementácia systému slúžiaceho na rozpoznávanie objektov v statických dvojrozmerných obrazoch. Navrhnutý systém má byť využiteľný v cloud robotike, teda má umožňovať spoluprácu s robotickými systémami. V nasledujúcich podkapitolách analyzujeme požiadavky na systém a poskytneme jeho teoretický návrh.

4.1. Požiadavky na systém rozpoznávania a jeho teoretický návrh

4.1.1. Výber klasifikačnej metódy

Keďže úloha rozpoznávania je vo svojej podstate klasifikačnou úlohou (viď. kapitolu 3.2), pre dosiahnutie nášho cieľa je potrebné implementovať vhodný klasifikátor. Vychádzajúc z analýzy vlastností jednotlivých klasifikačných metód sme navrhli použitie neurónových sietí.

Presnejšie navrhli sme použitie ART neurónových sietí. Tento návrh je podmienený predovšetkým inkrementálnosťou týchto neurónových sietí. Inkrementálnosť je veľmi vhodná vlastnosť predovšetkým pre použitie v prostredí cloudu, kde ktokoľvek a kedykoľvek môže trénovať neurónovú sieť na rozpoznávanie nových objektov, a to bez toho aby sieť zabudla klasifikovať skôr naučené objekty. Takýto prístup je vhodný aj v cloud robotike. Rozširuje sa tak množina objektov, ktoré je robot schopný správne klasifikovať.

V rámci skupiny ART sietí sme siahli po ARTMAP sieťach, ktoré sú trénované pomocou kontrolovaného učenia. Tento typ učenia nám vyhovuje, keďže neurónovú sieť trénujeme na základe statických obrazov, v ktorých používateľ pomenuje trénovaný objekt. Napokon v rámci ARTMAP sietí sme navrhli použitie MF ARTMAP siete. Tá kombinuje použitie teórií fuzzy logiky a adaptívnou rezonančnou teóriou, vďaka čomu je možné vyčíslenie hodnoty funkcie príslušnosti vstupnej vzorky ku každej naučenej triede. Navyše MF ARTMAP umožňuje vykonávať klasifikáciu a kategorizáciu súčasne.

4.1.2. Výber cloud platformy na vývoj a implementáciu

Z podmienky použiteľnosti systému v cloud robotike plyní ďalšia požiadavka a to, že navrhovaný systém musí byť vyvíjaný a implementovaný na niektorej z existujúcich cloud platforiem ako služba.

Ako sme uviedli v kapitole 2.1.3, existuje mnoho poskytovateľov cloud riešení. Napriek širokým možnostiam výberu poskytovateľa cloud riešenia sme sa rozhodli nami implementovaný systém

vyvíjať pre platformu Microsoft Windows Azure. Náš výber je podložený niekoľkými dôvodmi, medzi ktoré patrí možnosť vývoja cloud služby v profesionálnom vývojovom prostredí Visual Studio a v jazyku C#. Neoceniteľná je aj kvalitná a rozsiahla dokumentácia, tutoriály a oficiálne programátorské fóra nie len pre jazyk C#, ale aj pre platformu Microsoft Windows Azure. Vhodná je 30 dňová neobmedzená skúšobná verzia Azure cloudu a v neposlednom rade aj fakt, že študenti majú prístup ku plným verziám takmer všetkých vývojárskych nástrojov spoločnosti Microsoft zadarmo.

4.1.3. Výber frameworku a programovacieho jazyka

S výberom poskytovateľa cloud platformy pre implementáciu nepriamo súvisí aj zvolený programovací jazyk. Keďže sme sa rozhodli pre Microsoft Windows Azure platformu, aj pri programovaní cloud služby sme ostali verný produktom Microsoftu. Preto naša implementácia je vypracovaná vo vývojovom prostredí Microsoft Visual Studio 2013 vo verzií Professional. Používame ASP.NET MVC5 framework, ktorý slúži na vytváranie web aplikácií podľa návrhového vzoru Model-View-Controller (model, pohľad, kontrolér). Serverová časť implementovanej služby je programovaná v jazyku C#. Programovacie jazyky a technológie použité pri programovaní používateľského rozhrania sú popísané v nasledujúcej podkapitole. (kap. 4.1.4.)

4.1.4. Rozhrania cloud služby

Zvolenú neurónovú sieť sme navrhli implementovať ako cloud službu, ktorá má ponúkať dve rozhrania. Jedno je grafické a slúži pre interakciu človeka s touto službou. Druhé rozhranie je negrafické a slúži výlučne na komunikáciu robota Nao s implementovanou cloud službou.

4.1.4.1. Grafické rozhranie

Jednou z výhod cloud aplikácií je, že sú dostupné pomocou ľubovoľného zariadenia a internetového prehliadača. Aby sme zachovali túto vlastnosť, grafické rozhranie musí byť navrhnuté tak, aby bolo plne funkčné pri použití akéhokoľvek internetového prehliadača a zariadenia s rôznym operačným systémom a displejom s rôznym rozlíšením. Tiež nesmieme zabúdať na možnosť ovládania pomocou dotyku.

Operačný systém zariadenia pomocou ktorého používateľ pristupuje ku cloud aplikácií nie je podstatný, keďže cloud služba je umiestnená na vzdialenom serveri s vlastným operačným systémom. Na zabezpečenie ovládania cloud služby dotykom a kvalitné vykreslenie grafického rozhrania na displejoch s rôznym rozlíšením a v rôznych internetových prehliadačoch používame Bootstrap framework. Ten sa špecializuje práve na riešenie týchto problémov. Framework doplníme pomocou HTML5 a CSS3.

Obr. 21 zobrazuje grafické rozhranie zobrazené na rôznych zariadeniach a rôznym rozlíšením displeja.



Obr. 21 Zobrazenie grafického rozhrania na zariadeniach s rôznym rozlíšením displeja.

4.1.4.2. Negrafické rozhranie

Negrafické rozhranie umožňuje komunikáciu medzi robotom Nao a navrhnutou cloud službou. Robot sa tak môže stať zdrojom vizuálneho obrazu, ktorý je spracovaný cloud službou a naspäť mu je vracaný výsledok klasifikácie objektu v obraze. Tu sa už dostávame ku cloud robotike. Neustálym tréňovaním nových objektov sa rozširuje počet objektov, ktoré je robot schopný rozpoznať.

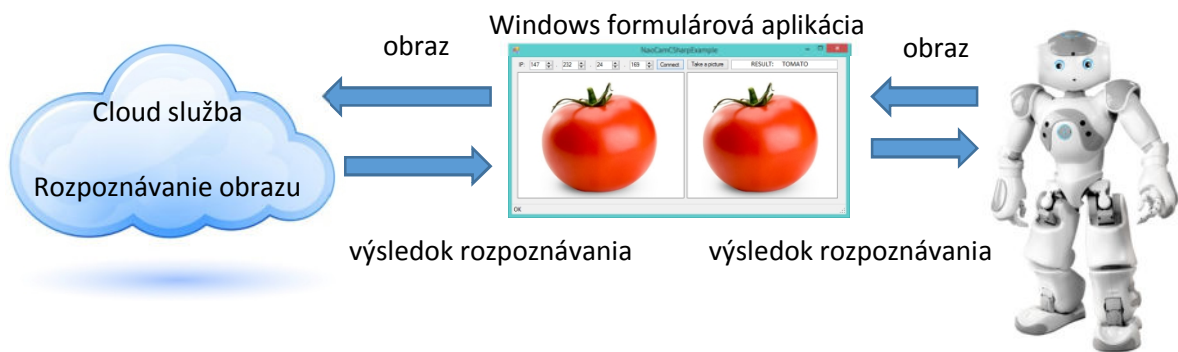
Na zabezpečenie komunikácie používame technológiu WCF (Windows Communication Foundation), ktorá je prístupná .NET⁴ frameworku od verzie 3.0. Je to technológia umožňujúca distribuované programovanie. Zabezpečuje prenos dát medzi klientom a serverom a obsahuje mechanizmy na ochranu prenášaných dát. Preto technológia WCF je ideálna pre komunikáciu robota Nao s cloud službou, kde robot Nao predstavuje klienta, komunikujúceho so vzdialeným serverom. V našom prípade vzdialeným serverom je cloud server. Viac o WCF je možné nájsť v na adrese <http://www.vyvojar.cz/Articles/454-wcf-zakladne-pojmy.aspx>.

Robot Nao môže vykonávať programy naprogramované v jazykoch Python, C++, Java a C#. Pričom robot priamo môže vykonávať len programy písané v jazyku Python alebo C++. Ostatné programy sú vykonávané na lokálnom počítači, ku ktorému je robot pripojený pomocou ethernetu alebo WiFi. Z lokálneho počítača sú robotovi odosielané inštrukcie, výsledky spracovania a všetky potrebné informácie. Program zase prijíma od robota vstupy.

Vzhľadom nato, že v celej implementácii používame Microsoft produkty a jazyk C#, rozhodli sme sa aj komunikáciu medzi robotom Nao a cloud službou programovať v tomto jazyku. Vytvorili sme

⁴ Používaný ASP.NET MVC5 framework je súčasťou .NET frameworku.

lokálnu Windows formulárovú aplikáciu, ktorá získava obraz z robota, ten spracuje a odošle vo forme statického obrazu do implementovanej cloud služby. Tam prebehne rozpoznávanie obrazu a do lokálnej aplikácie je vrátený výsledok rozpoznávania. Ten je ďalej preposlaný robotovi, ktorý prečíta výsledok používateľovi. Schéma tejto komunikácie je zobrazená na obr. 22



Obr. 22 Základná schéma komunikácie robota Nao s cloud službou

4.2. Implementácia navrhnutého systému

Proces rozpoznávania je finálnym krokom komplexnejšieho procesu spracovania statického vizuálneho obrazu. Preto pred samotnou implementáciou navrhnutého systému je nutné získať a predspracovať vizuálne obrazy do podoby vhodnej na klasifikáciu objektov, ktoré na sa na týchto obrazoch nachádzajú.

4.2.1. Príprava reprezentatívnej množiny obrazov

Prípravu reprezentatívnej množiny obrazov je možné stotožniť s prvým krokom spracovania obrazu, a to so získaním obrazu. Reprezentatívna množina obrazov je taká množina obrazov, kde objekt v každom obraze je klasifikovaný do určitej triedy. Reprezentatívna množina sa delí na tréningovú a testovaciu množinu. Pomocou nich je neurónová sieť tréňovaná a testovaná.

Obrazy tvoriace reprezentatívnu množinu sme získali z internetu a štandardných databáz obrazov používaných na vykonávanie experimentov v počítačovom videní. Existuje mnoho takýchto databáz. Každá z nich je vhodná na testovanie iných úloh počítačového videnia.

Existujú databázy obsahujúce logá firiem, dopravných značiek, automobilov, lietadiel, zvierat, rôznych druhov psov, koní, rastlín, listov stromov a kríkov, databázy kvetov, ľudských tvári a pod. Často krát obsahujú rovnaký obraz fotografovaný pod rôznym zorným uhlom a pri rozličných svetelných podmienkach. Tieto databázy ponúkajú naozaj širokú paletu obrazov použiteľných na riešenie prakticky ľubovoľnej úlohy počítačového videnia.

Zoznam mnohých databáz, rozdelených podľa vhodnosti použitia v rôznych úlohách počítačového videnia, so stručným popisom a odkazmi na domovské stránky daných databáz je možné nájsť na internetovej stránke s adresou <http://www.cvpapers.com/datasets.html>.

Pre účely experimentálnych prác sme vytvorili dve reprezentatívne množiny. Prvá obsahuje obrázky objektmi, ktoré sú tvorené jednoduchšími geometrickými tvarmi a sú ľahko rozpoznateľné. Každý obraz v tejto množine obsahuje práve jeden objekt umiestnený na bielom pozadí. Ten patrí vždy do práve jednej z piatich tried. Reprezentatívna množina obsahuje 50 obrazov (objektov). Každú triedu tvorí 10 tréningových obrazov. Triedy tejto reprezentatívnej množiny sú fľaša piva, mrkva, paradajka, logo reštaurácií rýchleho občerstvenia KFC a Mc Donalds. Obrazy tejto množiny boli získané z internetu. V ďalšom texte bude označovaná ako 1. RM

Druhá reprezentatívna množina je tvorená 57 obrazmi. Každý obraz obsahuje práve jeden objekt zobrazený na bielom pozadí. Tieto objekty sú v porovnaní s objektmi prvej reprezentatívnej množiny sú o niečo zložitejšie a komplexnejšie. Táto množina obsahuje objekty patriace vždy do jednej zo štyroch tried. Triedy sú baterka (na svietenie, nie akumulátor), ďalekohľad, monitor a pištoľ. Prvé tri triedy obsahujú po 15 tréningových obrazov a trieda pištoľ obsahuje 12 tréningových obrazov. Všetky obrazy boli získané z databáz počítačového videnia. Táto reprezentatívna množina bude v nasledujúcom texte označovaná ako 2. RM.

4.2.2. Predspracovanie a segmentácia obrazov reprezentatívnych množín

Proces predspracovania vizuálneho statického obrazu v počítačovom videní znamená úpravu obrazu a jeho zlepšenie pre potreby ďalšieho spracovania. Vzhľadom na obmedzenia kladené na obrazy vyberané do reprezentatívnych množín, sú vyžadované len minimálne úpravy. Tie pozostávajú predovšetkým zo zmenšenia obrazu a jeho prevodu z farebného na šedo-tónový obraz. Algoritmy extrakcie príznakov SIFT a SURF pracujú so šedo-tónovými obrazmi.

Vzhľadom nato, že každý obraz v reprezentatívnej množine obsahuje práve jeden objekt umiestnený na bielom pozadí, nie je potrebná žiadna segmentácia obrazu. Taktiež pri tréningu nových obrazov, ktoré nie sú v reprezentatívnej množine je vyžadovaný obraz obsahujúci práve jeden objekt umiestnený na bielom pozadí. Toto obmedzenie by mohlo byť odstránené použitím skriptu napísaného v jazyku JavaScript a implementovanom do grafického rozhrania cloud aplikácie. Takýto skript umožní aby používateľ označil objekt na obraze. Označený objekt ostane nezmenený a všetky ostatné časti obrazu, nachádzajúce sa mimo označenej oblasti, budú zafarbené na bielo.

Na predspracovanie obrazov reprezentatívnych množín a ich rozdelenia do tréningových a testovacích množín sme implementovali samostatnú triedu.

4.2.3. Detekcia príznakov a konštrukcia deskriptorov

V predchádzajúcich kapitolách bolo viac krát spomenuté, že neurónové siete na rozpoznávanie objektov sme trénovali pomocou troch rôznych typov deskriptorov. Každý typ deskriptorov bol získaný pomocou iného algoritmov počítačového videnia. Na tento účel boli použité algoritmy SIFT, SURF a algoritmus tvorby jednoduchých deskriptorov tvorené len farbou rozloženou do RGB zložiek pre každý pixel obrazu.

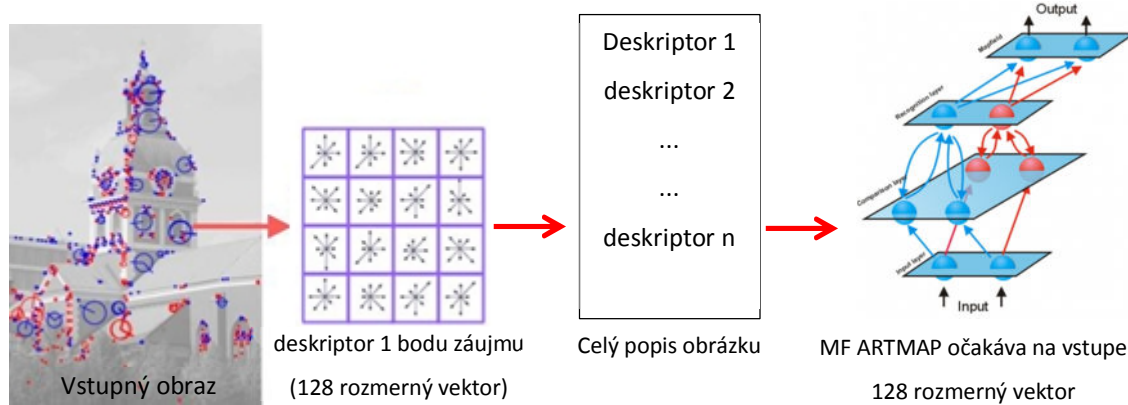
Deskriptory SIFT a SURF pracujú s textúrnymi príznakmi vyskytujúcimi sa v lokálnom okolí bodu záujmu. (viď. kapitoly 3.3.1. a 3.3.2.). Teda pracujú vo texturálnej oblasti obrazu. Naopak deskriptory tvorené farebnými zložkami každého pixelu obrazu pracujú s farebným spektrom, čiže pracujú v spektrálnej oblasti. To je hlavný rozdiel medzi použitými deskriptormi. Vhodnosť jednotlivých typov deskriptorov pri trénovaní neurónovej siete a následnej klasifikácii objektov je predmetom experimentálnej činnosti popísanej v kapitole 5.

Implementácie algoritmov extrakcie deskriptorov SIFT a SURF preberáme z knižnice EmguCV. EmguCV je knižnica, umožňujúca využívanie funkcií knižnice OpenCV pre jazyky frameworku .NET. Napríklad pre jazyk C#. OpenCV je knižnica, implementujúca algoritmy počítačového videnia. Správne pracuje s programovacími jazykmi C a C++. Algoritmy extrakcie príznakov sme implementovali ako samostatnú triedu.

4.2.4. Implementácia a trénovanie neurónovej siete typu MF ARTMAP

Neurónovú sieť typu MF ARTMAP sme implementovali ako dátovú štruktúru, ukladanú do tabuľky v cloud dátovom úložisku. Pre prácu s touto dátovou štruktúrou sme implementovali samostatnú triedu. Táto obsahuje funkcie pre vykonávanie algoritmu trénovanie neurónovej siete a pre vykonávanie klasifikácie objektu z neznámeho obrazu na už natrénovanej neurónovej sieti.

V tejto časti implementácie navrhnutého systému pre rozpoznávanie obrazu sme narazili na veľký problém. Ten spočíva v tom, že neurónová sieť očakáva na vstupe vždy konštantne rozmerný vektor. Algoritmy SIFT a SURF síce konštruujú deskriptory s konštantným počtom dimenzií, avšak ten nepopisuje celý vstupný obraz. Jeden deskriptor vždy popisuje len okolie jedného nájdeného bodu záujmu vo vstupnom obraze. Celkový popis obrazu pomocou algoritmov SIFT a SURF pozostáva z množiny deskriptorov pre nájdené body záujmu. Z toho vyplýva, že celkový popis vstupného obrazu je vektor s premenlivou dimenzionalitou závislou na počte nájdených bodov záujmu vo vstupnom obraze. Teda nie je možné trénovať celkový popis vstupného obrazu. Problém ilustruje obr. 23. Pre SIFT deskriptory. Rovnaký princíp platí aj pre SURF deskriptory s tým rozdielom, že deskriptor bodu záujmu nie je 128 rozmerný vektor ako pri SIFT deskriptore, ale len 64 rozmerný.

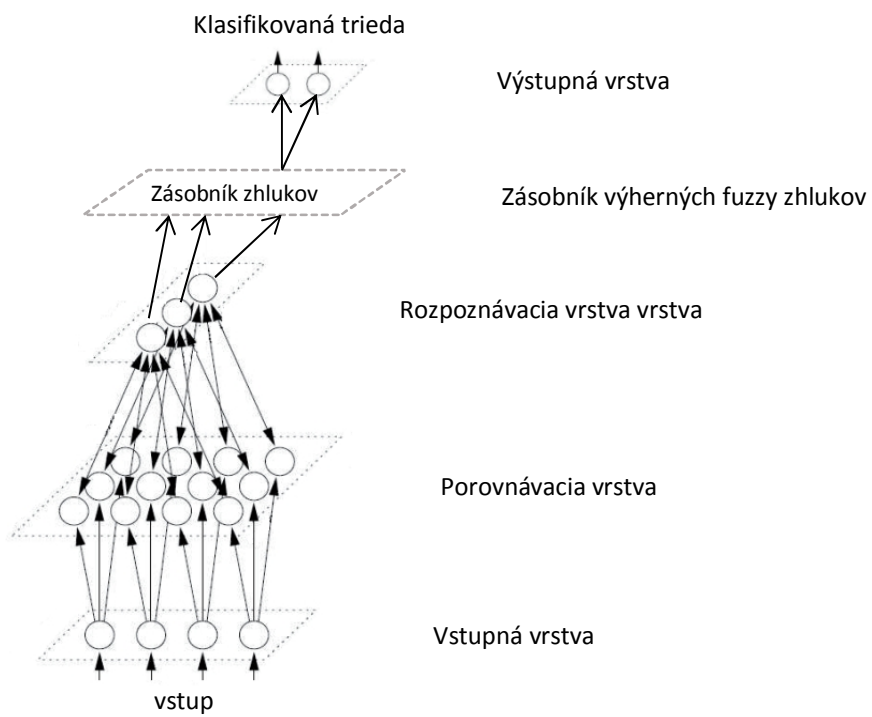


Obr. 23 Ilustrácia problému tréningu neurónovej siete

Sú dva možné prístupy k riešeniu problému. Jedným z nich je zanechanie deskriptorov SIFT a SURF, návrat do spektrálnej oblasti a tréning neurónovej siete pomocou farebných zložiek každého obrazového bodu. Tento prístup sme síce implementovali, avšak ako bude popísané v kapitole 5, výsledky klasifikácie týmto spôsobom nie sú uspokojivé. Týmto spôsobom trénujeme neurónovú sieť v podstate iba na rozpoznávanie pomeru farieb na obraze. Teda v prípade, že máme dva objekty, ktoré majú podobné farby a v obraze zaberajú približne rovnakú plochu, potom ich klasifikátor nedokáže klasifikovať. Príkladom je červené jablko a paradajka. Iba na základe farby sa nevieme rozhodnúť, či neznámy objekt je paradajka alebo jablko.

Druhým riešením je tréning neurónovej siete tak, že postupne budeme privádzať na vstup deskriptory jednotlivých nájdených bodov záujmu vo vstupnom obraze. Výstup z neurónovej siete nebude výsledná trieda, do ktorej je zaradený vstupný obraz, ale vektor víťazných neurónov (víťazných fuzzy zhukov). Rozmer výstupného vektora zodpovedá počtu nájdených deskriptorov vo vstupnom obraze. Tomuto vektoru výherných fuzzy zhukov musíme ďalej priradiť triedu objektu, ktorý je zobrazený na vstupnom obraze. Tento prístup vedie k zmene topológie MF ARTMAP siete, čím vznikne nový modifikovaný typ MF ARTMAP sietí. Modifikácia spočíva v pridaní zásobníka medzi rozpoznávaciu vrstvu a mapfield vrstvu, v ktorom sa budú odpamätávať výherné fuzzy zhuky pre všetky tréňované (testované) deskriptory v rámci jedného obrazu. Obr. 24 zobrazuje navrhovanú zmenu topológie MF ARTMAP sietí.

Pri tréningu sa následne celý obsah zásobníka priradí k triede tréňovaného objektu. V prípade testovania sa obsah zásobníka porovná s fuzzy zhukmi, ktoré boli určené ako výherné pre jednotlivé triedy v procese tréningu. Vstupný obraz (postupnosť deskriptorov jeho záujmových bodov) je klasifikovaný do triedy, ku ktorej je získaný zásobník najbližšie v zmysle maximálneho počtu zhodných výherných fuzzy zhukov medzi aktuálnym zásobníkom a triedou.



Obr. 24 Návrh zmeny topológie MF ARTMAP siete

5. Experimenty a analýza získaných výsledkov

Po implementácii neurónovej siete typu MF ARTMAP na cloude a naprogramovaní spracovania vstupných statických obrazov pomocou troch rôznych algoritmov počítačového videnia, je možné pristúpiť k vykonaniu experimentov.

Všetky experimenty boli navrhnuté a vykonané s cieľom porovnania presnosti klasifikácie neurónových sietí typu MF Artmap trénovaných pomocou deskriptorov získaných algoritmom SIFT, algoritmom SURF a pomocou RGB farebných zložiek každého pixelu vo vstupných obrazoch. Každý experiment bol vykonaný na dvoch rôznych trénovacích množinách (viď. kapitolu 4.2.1.)

Pri každom experimente sme zachovali konštantné rozdelenie reprezentatívnej množiny na množinu trénovaciu a testovaciu. Tento pomer je nasledovný: 60% všetkých vstupných obrazov z reprezentatívnej množiny pripadlo do trénovacej množiny. Zvyšných 40% vstupných obrazov bolo použitých na testovanie naučenej neurónovej siete a výpočet presnosti klasifikácie. Zachovali sme podmienku, aby obrazy jednotlivých tried boli v trénovacej aj testovacej množine zastúpené rovnomerne.

Vždy pri novom trénovaní neurónovej siete boli vytvorené nové trénovacie a testovacie množiny, pričom obrazy z reprezentatívnej množiny boli vyberané náhodne.

Presnosť klasifikácie vyjadrujeme pomocou kontingenčných a konfúzných⁵ tabuliek. Obe uvádzame spolu ako jednu tabuľku. V každej bunke tejto spojenej tabuľky sú dve hodnoty. Prvá zodpovedá hodnote z kontingenčnej tabuľky a druhá hodnote z konfúznej tabuľky. Tabuľky sú počítané pre presnosť klasifikácie vstupných obrazov z trénovacej množiny.

Pre jednoduchšie porovnanie presnosti uvádzame aj percentuálne vyjadrenú presnosť klasifikácie pre príklady z testovacej množiny, z trénovacej množiny a pre všetky príklady z reprezentatívnej množiny.

Ostatné parametre vytvorených neurónových sietí a presnosť ich klasifikácie budú prezentované v nasledujúcich podkapitolách. Celkový súhrn dosiahnutých výsledkov a ich analýza bude popísaná na tejto konci kapitoly.

5.1. Výsledky získané použitím reprezentatívnej množiny s jednoduchými obrazmi

Použitá reprezentatívna množina pozostáva z 50 obrazov rozdelených do 5 rôznych tried. Pre každú z tried je v množine 10 obrazov. Každý obraz patrí vždy len do jednej triedy. Teda triedy sú

⁵ Konfúzna tabuľka vyjadruje hodnoty kontingenčnej tabuľky ako percentuálny pomer hodnôt v rámci každého riadka tabuľky. Popis kontingenčnej tabuľky je možné nájsť na internete.

disjunktné. Triedy vyskytujúce sa v tejto reprezentatívnej množine sú fľaša piva, mrkva, paradajka a logá reštaurácií rýchleho občerstvenia KFC a Mc Donald's.

Vo všetkých nasledujúcich experimentoch je testovacia množina tvorená 30 vstupnými obrazmi a testovacia množina 20 vstupnými obrazmi.

5.1.1. Presnosť klasifikácie pri použití SIFT deskriptorov a 1. RM

Maximálnu presnosť klasifikácie na jednoduchšej trénovacej množine pri použití SIFT deskriptorov sme dosiahli použitím parametrov neurónovej siete zhrnutých v tab. 1. Tabuľka 2 predstavuje spojenú kontingenčnú a konfúznú tabuľku pre neurónovú sieť trénovanú pomocou SIFT deskriptorov a testovanú obrazmi z testovacej množiny.

V tabuľke 2, prvý riadok zodpovedá triedam, do ktorých boli objekty z testovacích obrazov klasifikované. Triedy v prvom stĺpci tabuľky sú triedy, ku ktorým reálne patrili objekty zobrazené na testovacích obrazoch. V ideálnom prípade, by sa nenulové hodnoty mali vyskytovať len na diagonále tejto tabuľky.

Parameter	Hodnota
E	2,00
F	2,10
Prah	0,50
Max. vzdialenosť vstupnej vzorky od stredu fuzzy zhlukov	0,40
Počet nájdených zhlukov	2161
Presnosť klasifikácie na testovacej množine	70,00%
Presnosť klasifikácie na trénovacej množine	100,00%
Presnosť klasifikácie na celej reprezentatívnej množine	85,00%

Tab. 1 Parametre a presnosť neurónovej siete pri použití SIFT deskriptorov a 1. RM

Reál. \ klas.	fľaša piva		mrkva		KFC		Mc Donald's		paradajka		neznáma trieda		Suma	
fľaša piva	3	75%	0	0%	1	25%	0	25%	0	0%	0	0%	4	100%
mrkva	0	0%	3	75%	0	0%	0	25%	1	25%	0	0%	4	100%
KFC	0	0%	0	0%	4	100%	0	0%	0	0%	0	0%	4	100%
Mc Donald's	0	0%	0	0%	0	0%	4	100%	0	0%	0	0%	4	100%
paradajka	0	0%	0	0%	0	0%	0	25%	0	0%	4	100%	4	100%

Tab. 2 Kontingenčná a konfúzna tab. pre testovaciu množinu pri použití SIFT deskriptorov a 1. RM

5.1.2. Presnosť klasifikácie pri použití SURF deskriptorov a 1. RM

Maximálnu presnosť klasifikácie na jednoduchšej trénovacej množine pri použití SURF deskriptorov sme dosiahli použitím parametrov neurónovej siete zhrnutých v tab. 3. Tabuľka 4 opäť predstavuje spojenú kontingenčnú a konfúznú tabuľku pre neurónovú sieť trénovanú pomocou SURF deskriptorov a testovanú obrazmi z testovacej množiny.

Tabuľka 4 sa číta rovnakým spôsobom ako je popísané v podkapitole 5.1.1.

Parameter	Hodnota
E	8,00
F	2,00
Prah	0,70
Max. vzdialenosť vstupnej vzorky od stredu fuzzy zhlukov	0,20
Počet nájdených zhlukov	3075
Presnosť klasifikácie na testovacej množine	70,00%
Presnosť klasifikácie na tréningovej množine	100,00%
Presnosť klasifikácie na celej reprezentatívnej množine	85,00%

Tab. 3 Parametre a presnosť neurónovej siete pri použití SURF deskriptorov a 1. RM

Reál. \ klas.	fľaša piva		mrkva		KFC		Mc Donald's		paradajka		neznáma trieda		Suma	
fľaša piva	3	75%	0	0%	0	0%	0	0%	1	25%	0	0%	4	100%
mrkva	3	75%	0	0%	0	0%	1	25%	0	0%	0	0%	4	100%
KFC	0	0%	0	0%	4	100%	0	0%	0	0%	0	0%	4	100%
Mc Donald's	0	0%	0	0%	1	0%	3	75%	0	0%	0	0%	4	100%
paradajka	0	0%	1	25%	0	0%	0	0%	3	75%	0	0%	4	100%

Tab. 4 Kontingenčná a konfúzna tab. pre testovaciu množinu pri použití SURF deskriptorov a 1. RM

5.1.3. Presnosť klasifikácie pri použití RGB farebných zložiek obrazových bodov a 1.

RM

Najlepšiu presnosť klasifikácie sme dosiahli pri použití RGB farebných zložiek každého pixelu vstupného obrazu a jednoduchšej tréningovej množiny zadaním parametrov zhrnutých v tabuľke 5. Tab. 6 predstavuje kontingenčnú a konfúznu tabuľku pre neurónovú sieť tréningovanú pomocou RGB farebných zložiek jednotlivých pixelov vo stupnom obraze. V tejto podkapitole uvádzame aj spojenú kontingenčnú a konfúznu tabuľku (tab. 7) pre testovanie tých vstupných obrazov, pomocou ktorých bola neurónová sieť tréningovaná. Tým demonštrujeme nemožnosť siete naučiť sa správne klasifikovať objekty zobrazené na vstupných obrazoch, a to ani pri zmene parametrov siete.

Parameter	Hodnota
E	6,00
F	2,25
Prah	0,60
Max. vzdialenosť vstupnej vzorky od stredu fuzzy zhlukov	0,20
Počet nájdených zhlukov	798
Presnosť klasifikácie na testovacej množine	70,00%
Presnosť klasifikácie na tréningovej množine	90,00%
Presnosť klasifikácie na reprezentatívnej množine	80,00%

Tab. 5 Parametre a presnosť neurónovej siete pri použití RGB farebných zložiek pixelov vstupného obrazu a 1. RM

Reál. \ klas.	fľaša piva		mrkva		KFC		Mc Donald's		paradajka		neznáma trieda		Suma	
fľaša piva	2	50%	1	25%	1	25%	0	0%	0	0%	0	0%	4	100%
mrkva	2	50%	2	50%	0	25%	0	0%	0	0%	0	0%	4	100%
KFC	0	0%	0	0%	4	100%	0	0%	0	0%	0	0%	4	100%
Mc Donald's	1	25%	0	0%	0	0%	3	75%	0	0%	0	0%	4	100%
paradajka	0	0%	0	0%	1	25%	0	0%	3	75%	0	0%	4	100%

Tab. 6 Kontingenčná a konfúzna tabuľka pre testovaciu množinu pri použití RGB farebných zložiek pixelov vstupného obrazu

Reál. \ klas.	fľaša piva		mrkva		KFC		Mc Donald's		Paradajka		neznáma trieda		Suma	
fľaša piva	5	83,33%	0	0%	1	16,67%	0	0%	0	0%	0	0%	6	100%
mrkva	0	0%	6	100%	0	0%	0	0%	0	0%	0	0%	6	100%
KFC	0	0%	0	0%	6	100%	0	0%	0	0%	0	0%	6	100%
Mc Donald's	0	0%	0	0%	0	0%	6	100%	0	0%	0	0%	6	100%
paradajka	2	33,33%	0	0%	0	0%	0	0%	4	66,67%	0	0%	6	100%

Tab. 7 Kontingenčná a konfúzna tabuľka pre trénovaciu množinu pri použití RGB farebných zložiek pixelov vstupného obrazu

5.2. Výsledky získané použitím reprezentatívnej množiny so zložitejšími obrazmi

Objekty zobrazené na obrazoch tejto reprezentatívnej množiny sú v porovnaní s objektmi predchádzajúcej reprezentatívnej množiny o niečo zložitejšie a detailnejšie. Množina je tvorená 57 obrazmi patriacich vždy do jednej zo 4 tried. Z toho vyplýva, že triedy objektov sú disjunktné. Triedy sú baterka (prenosný zdroj svetla, nie akumulátor), ďalekohľad, monitor a pištoľ. Pre prvé tri spomenuté triedy sa v reprezentatívnej množine nachádza po 15 obrazov a pre triedu „pištoľ“ je pripravených 12 trénovacích obrazov. Aj tu zachovávame podmienku z predchádzajúcej reprezentatívnej množiny, že na každom obraze môže byť len jeden objekt umiestnený na bielom pozadí.

Vo všetkých nasledujúcich experimentoch je testovacia množina tvorená 34 vstupnými obrazmi a testovacia množina 23 vstupnými obrazmi.

5.2.1. Presnosť klasifikácie pri použití SIFT deskriptorov a 2. RM

Maximálnu presnosť klasifikácie pri použití druhej (o niečo zložitejšej) trénovacej množiny a pri trénovaní neurónovej siete pomocou SIFT deskriptorov, dosiahneme ak zvolíme parametre neurónovej siete zaznamenané v tab. 8.

Parameter	Hodnota
E	0,50
F	5,00
Prah	0,50
Max. vzdialenosť vstupnej vzorky od stredu fuzzy zhluku	0,50
Počet nájdených zhlukov	2165
Presnosť klasifikácie na testovacej množine	65,22%
Presnosť klasifikácie na trénovacej množine	100,00%
Presnosť klasifikácie na celej reprezentatívnej množine	82,61%

Tab. 8 Parametre a presnosť neurónovej siete pri použití SIFT deskriptorov a 2. RM

Reál. \ klas.	baterka		Ďalekohľad		monitor		pištoľ		neznáma trieda		Suma	
baterka	5	83,33%	0	0%	0	0%	1	16,67%	0	0%	6	100%
ďalekohľad	0	0%	5	83,33%	1	16,67%	0	0%	0	0%	6	100%
Monitor	2	33,33%	0	0%	1	16,67%	3	50%	0	0%	6	100%
pištoľ	1	20%	0	0%	0	0%	4	80%	0	0%	5	100%

Tab. 9 Kontingenčná a konfúzna tab. pre testovaciu množinu pri použití SIFT deskriptorov a 2. RM

5.2.2. Presnosť klasifikácie pri použití SURF deskriptorov a 2. RM

Najlepší klasifikátor pri použití druhej trénovacej množiny a pri trénovaní neurónovej siete pomocou SURF deskriptorov sme získali nastavením vstupných parametrov neurónovej siete na hodnoty zhrnuté v tab. 10.

Parameter	Hodnota
E	8,00
F	2,00
Prah	0,70
Max. vzdialenosť vstupnej vzorky od stredu fuzzy zhluku	0,20
Počet nájdených zhlukov	2895
Presnosť klasifikácie na testovacej množine	65,22%
Presnosť klasifikácie na trénovacej množine	100,00%
Presnosť klasifikácie na celej reprezentatívnej množine	82,61%

Tab. 10 Parametre a presnosť neurónovej siete pri použití SURF deskriptorov a 2. RM

Reál. \ klas.	baterka		Ďalekohľad		monitor		pištoľ		neznáma trieda		Suma	
baterka	4	66,67%	0	0%	1	16,67%	1	16,67%	0	0%	6	100%
ďalekohľad	1	16,67%	2	33,33%	3	50%	0	0%	0	0%	6	100%
Monitor	0	0%	0	0%	6	100%	0	0%	0	0%	6	100%
pištoľ	0	0%	1	20%	1	20%	3	60%	0	0%	5	100%

Tab. 11 Kontingenčná a konfúzna tab. pre testovaciu množinu pri použití SURF deskriptorov a

2. RM

5.2.3. Presnosť klasifikácie pri použití RGB farebných zložiek obrazových bodov a 2.

RM

Podobné výsledky ako pri prvej reprezentatívnej množine sme získali aj použitím druhej reprezentatívnej množiny a trénovaním neurónovej siete pomocou RGB farebných zložiek obrazu.

Najvyššia presnosť, ktorú sme dosiahli a parametre neurónovej siete sú zhrnuté v tab. 12.

Pre porovnanie s prvou trénovacou množinou uvádzame aj spojenú kontingenčnú a konfúzna tabuľku (tab. 14) zobrazujúcu presnosť klasifikovania tých vstupných obrazov, ktoré boli použité na trénovanie neurónovej siete.

Parameter	Hodnota
E	8,00
F	2,50
Prah	0,60
Max. vzdialenosť vstupnej vzorky od stredu fuzzy zhlukov	0,50
Počet nájdených zhlukov	681
Presnosť klasifikácie na testovacej množine	56,52%
Presnosť klasifikácie na trénovacej množine	91,18%
Presnosť klasifikácie na celej reprezentatívnej množine	73,85%

Tab. 12 Parametre a presnosť neurónovej siete pri použití RGB farebných zložiek pixelov vstupného obrazu a 2. RM

Reál. \ klas.	baterka		Ďalekohľad		monitor		pištoľ		neznáma trieda		Suma	
baterka	5	83,33%	0	0%	1	16,67%	0	0%	0	0%	6	100%
ďalekohľad	3	50%	3	50%	0	0%	0	0%	0	0%	6	100%
Monitor	1	16,67%	1	16,67%	3	50%	1	16,67%	0	0%	6	100%
pištoľ	2	40%	0	0%	1	20%	2	40%	0	0%	5	100%

Tab. 13 Kontingenčná a konfúzna tabuľka pre testovaciu množinu pri použití RGB farebných zložiek pixelov vstupného obrazu a 2. RM

Reál. \ klas.	baterka		Ďalekohľad		monitor		pištoľ		neznáma trieda		Suma	
baterka	8	88,89%	1	11,11%	0	0%	0	0%	0	0%	9	100%
ďalekohľad	1	11,11%	7	77,78%	1	11,11%	0	0%	0	0%	9	100%
Monitor	0	0%	0	0%	9	100%	0	0%	0	0%	9	100%
pištoľ	3	42,86%	0	0%	0	0%	4	57,14%	0	0%	7	100%

Tab. 14 Kontingenčná a konfúzna tabuľka pre tréningovú množinu pri použití RGB farebných zložiek pixelov vstupného obrazu a druhej reprezentatívnej množiny

5.3. Analýza výsledkov

V tabuľke 15 je uvedený sumárny prehľad dosiahnutej presnosti všetkých experimentálnych neurónových sietí.

		Použitie 1. RP			Použitie 2. RP		
		SIFT	SURF	RGB	SIFT	SURF	RGB
Presnosť klasifikácie	tréningová množina	100,0%	100,0%	90,0%	100,0%	100,0%	91,2%
	testovacia množina	70,0%	65,0%	70,0%	65,2%	65,2%	56,5%
	reprezentatívna množ.	85,0%	82,5%	80,0%	82,6%	82,6%	73,9%
Počet nájdených zhlukov		2161	3075	798	2165	2895	681
Schopnosť generalizácie NN		0,223	0,491	0,999	0,149	0,423	0,998

Tab. 15 Sumárny prehľad dosiahnutých výsledkov

Dosiahnuté výsledky nepreukázali dosiahnutie výrazne lepších výsledkov pri tréningu a testovaní jednoduchších, ľahko rozlíšiteľných objektov. Z toho vyplýva, že neurónová sieť MF ARTMAP použitá ako klasifikátor dokáže rovnako kvalitne klasifikovať ľubovoľne zložité objekty.

Naše pozorovania preukázali, že príliš jednoduché objekty tvorené jednou dominantnou farbou, neobsahujúce hrany alebo rohy sú ťažko rozpoznateľné a klasifikovateľné. Sú to napríklad paradajka, mrkva a pod. Popis takýchto objektov pomocou SIFT alebo SURF deskriptorov je nepostačujúci.

Ak sa zameriame na rozdiely medzi zvolenými druhmi tréningu neurónových sietí, môžeme konštatovať, že neurónové siete tréningované pomocou SIFT deskriptorov sa dokážu učiť nové objekty a neskôr ich aj klasifikovať rýchlejšie ako ostatné dve metódy.

Doba učenia a klasifikácie nového objektu pri použití MF ARTMAP siete tréningovanej pomocou zvyšných dvoch metód je vyššia. Pri tréningu pomocou SURF deskriptorov je dlhšia doba

trénovania/testovania spôsobená vyšším počtom nájdených deskriptorov v každom vstupnom obraze. Taktiež sieť MF ARTMAP trénovaná SURF deskriptormi nájde približne 1000 fuzzy zhlukov viac v porovnaní s trénovaním pomocou SIFT deskriptorov. Samozrejme pri použití rovnakej trénovacej množiny.

Napriek tomu, že neurónové siete trénované pomocou RGB farebných zložiek každého pixelu nájdu najmenej fuzzy zhlukov, ich doba učenia sa a klasifikácie je pomerne vysoká. Je to spôsobené tým, že každý obrazový bod vstupného obrazu je považovaný za trojrozmerný deskriptor obsahujúci jednotlivé farebné zložky daného bodu. Už pri obraze s rozlíšením 100x100 pixelov dostávame 10 000 deskriptorov, ktoré je potrebné trénovať resp. klasifikovať. Je tu však možná optimalizácia, ktorá spočíva s popise oblastí vstupného obrazu tvorených približne rovnakými farbami. Následne by boli trénované resp. klasifikované už len popisy nájdených oblastí.

Na základe vykonaných experimentov sa neurónové siete trénované pomocou SIFT deskriptorov javia ako najvhodnejšie na klasifikáciu a kategorizáciu objektov zo statických obrazov. Pričom vychádzame z kratšieho času učenia sa a klasifikácie nových objektov, z nižšieho počtu nájdených deskriptorov vo vstupných obrazoch, z nižšieho počtu nájdených fuzzy zhlukov a z porovnateľnej presnosti klasifikácie v porovnaní so sieťami trénovanými pomocou SURF deskriptorov.

Neurónové siete trénované pomocou RGB farebných zložiek pixelov sú vhodnými sieťami na klasifikáciu v špeciálnych úlohách. Vzhľadom nato, že tieto siete sa učia len farby pixelov, výsledkom ich učenia sú pomery farieb v jednotlivých trénovacích obrazoch. Preto sú vhodné použiteľné na úlohy, kde rozoznávame dominantu farbu v obraze. Napríklad rozpoznanie farieb automobilov prechádzajúcich cez mýtnu bránu na diaľnici. Tieto siete zlyhávajú ak rozoznávajú dva rôzne objekty s rovnakou farbou. Príkladom takéhoto zlyhania v našich experimentoch je klasifikácie až 50% všetkých testovacích objektov triedy ďalekohľad do triedy baterka len na základe podobností farieb týchto objektov.

MF ARTMAP neurónové siete použité v našich experimentoch sa vyznačujú pomerne veľkou náhodnosťou pri učení, pričom nezáleží na zvolenej metóde tréovania. Presnosť klasifikácie neurónovej siete závisí od poradia trénovacích obrazov (resp. od poradia deskriptorov), ktoré privádzame na vstup. Ak v našich experimentoch viackrát pretrénujeme neurónovú sieť s rovnakými parametrami, takmer vždy dostávame inú presnosť klasifikácie. Tento jav súvisí s nízkou hodnotou generalizácie neurónových sietí, ktorú sme dosiahli. Generalizácia⁶ neurónových znamená schopnosť neurónových sietí zovšeobecňovať. Pri nízkych hodnotách

⁶ Generalizácia sa určuje ako pomer nájdených fuzzy zhlukov ku všetkým trénovaným deskriptorom.

generalizácie je schopnosť zovšeobecňovania slabá a neurónová sieť vytvára fuzzy zhluky špecifické pre takmer každý trénovaný deskriptor. Potom nový neznámy obraz nemôže byť správne klasifikovaný, keďže ani jeden z jeho deskriptorov nedokáže presiahnuť požadovaný prah pre zaradenie do niektorého z nájdených fuzzy zhlukov. Pri neurónových sieťach trénovaných pomocou RGB farebných zložiek pixelov sme získali vysokú mieru generalizácie, avšak presnosť klasifikácie tam znižovali problémy nemožnosti rozpoznania objektov s podobnou farbou.

Problém nízkej generalizácie MF ARTMAP neurónových sietí by mohol byť riešený zmenou funkcie fuzzy relácie používanej v sieti MF ARTMAP. Súčasne používaná funkcia fuzzy relácie (vzťah 10) nedokáže dostatočne presne popísať tvar zložitého fuzzy zhluku. Nájdené fuzzy zhluky tak pokrývajú aj vstupné vzorky, ktoré do zhluku v skutočnosti nepatria resp. nepokrývajú vstupné vzorky, ktoré by do nich mali patriť. Pri snahe dosiahnuť vyššiu presnosť klasifikácie sme museli nastaviť inicializačné parametre fuzzy zhlukov tak, aby sa získavali čo najšpecifickejšie zhluky, čím sme rapídne znížili schopnosť generalizácie. Takto naučené neurónové siete klasifikujú správne všetky trénovacie obrazy a obrazy veľmi podobné trénovacím, ale iné obrazy už nie sú klasifikované správne.

Záver

V tejto práci sme čitateľovi poskytli prehľad cloudových technológií, servisných modelov cloudu, cloud robotiky a jej aplikačnom potenciály.

Taktiež sme sa venovali problematike spracovania obrazu, procesu rozpoznávania, klasifikácií a kategorizácií a spôsobu popisu vizuálnych obrazov pomocou deskriptorov SIFT a SURF.

Spojenie oboch častí nám vytvorilo teoretický základ pre návrh a implementáciu systému slúžiaceho na rozpoznávanie objektov zo statických obrazov v prostredí cloudu. Na účely rozpoznávania objektov z obrazov sme implementovali neurónovú sieť typu MF ARTMAP, ktorú sme trénovali pomocou troch rôznych metód a to pomocou deskriptorov obrazov získaných algoritmi SIFT, SURF a pomocou RGB farebných zložiek každého obrazového bodu vstupného obrazu.

Systém sme vyvíjali na platforme Microsoft Windows Azure. Počas vývoja sme narazili na problém trénovania siete MF ARTMAP pomocou popisov obrazov tvorených rôznym počtom SIFT resp. SURF deskriptorov. Ten sme vyriešili zmenou topológie neurónovej siete MF ARTMAP. Presnejšie pridali sme zásobník medzi rozpoznávaciu vrstvu a mapfield, čo umožnilo trénovať a testovať vstupné obrazy ako množiny ich deskriptorov.

Implementáciou navrhnutého systému vznikol priestor na vykonávanie experimentov. Tie preukázali, že trénovanie neurónovej siete pomocou deskriptorov získaných algoritmom SIFT je najvhodnejšie. Taktiež sme zistili, že neurónová sieť typu MF ARTMAP nie je vhodná pre účely rozpoznávania objektov z obrazov, keďže využíva funkciu fuzzy relácie, ktorá nedokáže presne popísať tvar zložitejších fuzzy zhlukov. Zámena tejto funkcie za novú si však vyžaduje hlbší a dlhodobjší výskum, ktorý by mal byť predmetom nášho ďalšieho štúdia.

Zoznam použitej literatúry

- [1] **PRANTL, M. et al.** : *Glossary of Computer Vision Term in Connection to Information Fusion* [online]. [cit. 2014-4-25]. Dostupné na internete: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.1327&rep=rep1&type=pdf>>.
- [2] **SMOLÁR, P. - Sinčák, P. - Virčíková, M.** : *Intelligent Image Categorization*. Košice : LAMBERT Academic Publishing, 2012. 181s. ISBN 978-3-659-24795-8.
- [3] **Sinčák, P. - Andrejková, G.** : *Neurónové siete, Inžiniersky prístup (1. diel)*. Košice : ELFA press, 1996. 107s. ISSN 80-8878-638-X.
- [4] **Vojterk, P.** : *Terminológia - klasifikácia, kategorizácia, zhlukovanie : PeWe seminár*. [online]. Bratislava : STU, FIIT, 2007. [cit. 2014-4-27]. Dostupné na internete: <<http://wiki.fiit.stuba.sk/research/seminars/pewe/terminology/vojtek-class-cat-clust.pps>>.
- [5] **CICMAN, D.** : *Cloud computing ako model vývoja a používania IT technológií v podnikoch*. In: *Key Competencies in Knowledge Society: Today trends*. Zborník z medzinárodnej konferencie, Brusel : EuroScientia, 2012. ISBN 978-90-818529-5-1.
- [6] **autor neznámy**: *Oxford Dictionaries : Definition of cloud computing in English*. [Online]. Oxford : Oxford University Press, 2014. [cit. 2014-4-15]. Dostupné na internete: <<http://www.oxforddictionaries.com/definition/english/cloud-computing>>.
- [7] **IBM Slovensko** : *IBM Cloud Computing*. [Online]. Bratislava : IBM Slovensko. [cit. 2014-4-18]. Dostupné na internete: <<http://www-05.ibm.com/sk/cloud/>>.
- [8] **Lorenčík, D. et al.** : *Cloudová robotika, vplyv cloudového computingu na budúcnosť robotiky (1)*. In: *ATP journal*. roč. 21, 2014, č. 3, s. 40 - 42. 3/2014, ISSN 1335-2237.
- [9] **Microsoft Slovensko** : *Windows Azure*. [Online]. Bratislava : Microsoft Slovensko [cit. 2014-4-17]. Dostupné na internete: <<http://www.microsoft.com/slovakia/azure/products/>>.
- [10] **GUIZZO, E.** : *Cloud Robotics: Connected to the Cloud, Robots Get Smarter*. In *IEEE Spectrum* [Online]. 2011. [cit. 2014-25]. Dostupné na internete: <<http://spectrum.ieee.org/automaton/robotics/robotics-software/cloud-robotics>>.
- [11] **HU, G. - Tay, W. P. - Wen, Y.** : *Cloud Robotics: Architecture, Challenges and Applications*.

- Singapore : IEEE Network Magazine, 2012, Zv. 26. s. 21 - 28. ISSN 0890-8044.
- [12] **Waibel, M et al.** : *RoboEarth*. In: *Robotics & Automation Magazine, IEEE*. 2011, zv. 18, s. 69 - 82.
- [13] **Goldberg, K.** : *Cloud Robotics and Automation* [Online]. Berkeley: University of California. [cit 2014-4-28]. Dostupné na internete: <<http://goldberg.berkeley.edu/cloud-robotics/>>.
- [14] **Kehoe, B. et. al.** : *Cloud-Based Robot Grasping with the Google Object Recognition Engine*. In: IEEE International Conference on Robotics and Automation, 2010. s. 3084 - 3089. ISSN 1050-4729
- [15] **Arumugam, R. et al.** : *DAvinCi: A cloud computing framework for service robots*. In: *IEEE Robot. 2011, zv. 18, s. 69 - 82*. Anchorage : IEEE, 2010. 1050-4729.
- [16] **Tomori, Zoltán.** *Počítačové videnie (elektronické učebné texty pre poslúchačov FEI, TUKE)* [Online]. Košice : TUKE, FEI, 2010. [cit. 2014-4-26]. Dostupné na internete: <<http://neuron.tuke.sk/jelen/PV/PV.pdf>>.
- [17] **autor neznámy** : *OpenCV dokumentácia: Introduction to SIFT (Scale-Invariant Feature Transform)* [Online]. [cit. 2014-4-27]. Dostupné na internete: <http://docs.opencv.org/trunk/doc/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html>.
- [18] **Reiff, Tomáš:** *Inkrementálny systém pre rozpoznávanie obrazových vzorov*: Diplomová práca. Košice : TUKE FEI, 2008, 76 s.
- [19] **Šváb, Ján:** *Akcelerace zpracování obrazu hradlovým polem*: Bakalárska práca. Praha : ČVÚT, FE, 2009. 35 s.
- [20] **Ondo, Jaroslav:** *Využitie umelej inteligencie v inteligentných budovách*: Bakalárska práca. Košice : TUKE, FEI, 2009.
- [21] **Smolár, Peter:** *Rozpoznávanie obrazu neurónovými sieťami typu ART*: Dizertačná práca. Košice : TUKE, FEI, 2010. 57 s.
- [22] **Sinčák, P. - Andrejková, G.** *Neurónové siete, Inžiniersky prístup (2. diel)*. Košice : ELFA Press, 1996. 63s.

Prílohy

Príloha A: Používateľská príručka

Príloha B: Systémová príručka

Príloha C: CD médium