VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÉ TŘÍDĚNÍ FOTOGRAFIÍ PODLE OBSAHU

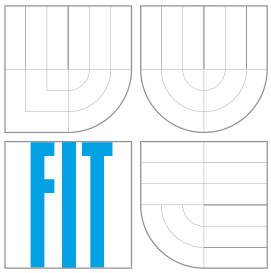
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN VEĽAS

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÉ TŘÍDĚNÍ FOTOGRAFIÍ PODLE OBSAHU

AUTOMATIC PHOTOGRAPHY CATEGORIZATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN VEĽAS

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL ŠPANĚL, Ph.D.

BRNO 2011

Abstrakt

Tato práce se zabývá automatickou kategorizací fotografií podle obrazového obsahu. Cílem práce bylo vytvořit aplikaci, která je schopna s dostatečnou přesností a rychlostí tuto úlohu naplnit. Základní řešení obnáší detekci význačných bodů a extrakci lokálních příznaků, tvorbu vizuálního slovníku shlukováním metodou k-means a jeho reprezentaci pomocí k-dimenzionálního stromu. Fotografie je reprezentována pomocí histogramu početnosti výskytu vizuálních slov (bag of words). Úlohu vlastního klasifikátoru plní SVM (support vector machines). Dále je základní řešení obohaceno o dělení obrazu na části se samostatným zpracováním, využití barevných korelogramů pro doplňkový popis obrazu, extrakci lokálních příznaků v opponent color space a měkké přiřazení vizuálních slov k extrahovaným příznakovým vektorům. Závěr práce je věnován experimentům se zmíněnými technikami a vyhodnocování výsledků kategorizace při jejich použití.

Abstract

This thesis deals with content based automatic photo categorization. The aim of the work is to create an application, which is would be able to achieve sufficient precision and computation speed of categorization. Basic solution involves detection of interesting points, extraction of feature vectors, creation of visual codebook by clustering, using k-means algorithm and representing visual codebook by k-dimensional tree. Photography is represented by bag of words – histogram of presence of visual words in a particular photo. Support vector machines (SVM) was used in role of classifier. Afterwards the basic solution is enhanced by dividing picture into cells, which are processed separately, computing color correlograms for advanced image description, extraction of feature vectors in opponent color space and soft assignment of visual words to extracted feature vectors. The end of this thesis concerns to experiments of of above mentioned techniques and evaluation of the results of image categorization on their usage.

Klíčová slova

kategorizace fotografii, tagování, lokální příznaky, význačné body, SURF, experimenty, vizuální slovník, vizuální slovy, shlukování, k-means, k-dimenzionální strom, bag of words, support vector machines, klasifikátor, dělení obrazu, barevné příznaky, barevné korelogramy, opponent color space, měkké přirazení, knihovna OpenCV

Keywords

photo categorization, tagging, local features, interesting points, SURF, experiments, visual codebook, visual word, clustering, k-means, k-dimensional tree, bag of words, support vector machines, classifier, dividing of image, color features, color correlograms, opponent color space, soft assignment, OpenCV library

Citace

Martin Veľas: Automatické třídění fotografií podle obsahu, bakalářská práce, Brno, FIT VUT v Brně, 2011

Automatické třídění fotografií podle obsahu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Španěla. Všechny literární zdroje a publikace, které jsem použil, jsou uvedeny v seznamu použité literatury.

.....
Martin Veľas
11. mája 2011

Poděkování

Chci se poděkovat svému konzultantovi Ing. Michalovi Španělovi za pomoc a cenné rady při tvorbě této bakalářské práce.

Obsah

Kapitola 1

Úvod

Ľudia v dnešnej dobe žijú v informačnej spoločnosti - každý deň sa stretávame s informáciami, ktoré prijímame, spracúvame a využívame. Tieto informácie k nám prichádzajú v rôznej podobe a už tradične s nimi pracuje s využitím výpočtovej techniky. Nevyhnutnosťou je informácie zotriediť, označiť či iným spôsobom organizovať.

Táto práca sa zaoberá *automatickým triedením fotografií* do jednotlivých kategórií na základe ich obrazového obsahu. Okrem tohto faktoru môžu byť fotografie organizované na základe dátumu ich vytvorenia, prípadne podľa iných prívlastkov, ktoré im priradia manuálne samotní užívatelia (napr. na sociálnych sietiach). Takáto organizácia je však irelevantná, pokiaľ je fotografia vyhľadávaná s ohľadom na jej obsah a taktiež automatizácia organizácia oproti manuálnej predstavuje úsporu času užívateľa.

Základná kostra práce vychádza z článku *Automatic photo tagging* od *Š.Rosu*[15]. Tá je upravená pre potreby tejto práce a ďalej doplnená o pokročilejšie techniky kategorizácie.

Hlavným cieľom práce je vytvoriť aplikáciu vo vybranom programovacom jazyku, s využitím dostupných knižníc, ktorá bude schopná zaraďovať fotografie do vopred zvolených kategórii len na základe samotnej fotografie bez dodatočných informácií o nej. Hlavné požiadavky na túto aplikáciu sú spoľahlivosť a taktiež efektivita behu, keďže sa predpokladá veľké množstvo fotografií roztriedených za rozumný čas.

Pri riešení boli naštudované techniky spracovania obrazu, extrakcie a práce s obrazovými príznakmi. Tie vychádzajú buď z tradičných postupov kategorizácie obrazu, alebo z novších publikácií. Moderné prístupy zvyšujú robustnosť, spoľahlivosť i rýchlosť aplikácie. Experimentovanie s nimi je teda cestou pre získanie optimálneho riešenia.

V nasledujúcej kapitole *Kategorizácia obsahu* budú predstavené jednotlivé použité techniky a taktiež popis ich teoretického základu. *Kapitola 3 – State of art* obsahuje zhrnutie postupov, ktoré sú v súčasnej dobe využívané v oblasti kategorizácie obrazu. *Kapitola 4 – Návrh riešenia* obsahuje výber metód použitých v tejto práci, ich prepojenie a návrh testov ich fungovania. V *kapitole 5 – Implementácia* budú predstavené použité programovacie prostriedky, využité knižnice, spôsob implementácie algoritmov a popis najdôležitejších modulov/tried. *Kapitola 6 – Výsledky práce* obsahuje vyhodnotenie výsledkov jednotlivých experimentov – výber rôznych techník a ich optimalizácie. Zhrnutie dosiahnutých výsledkov a zhodnotenie miery splnenia vytýčených cieľov sa nachádza v *kapitole 7 – Záver*.

Kapitola 2

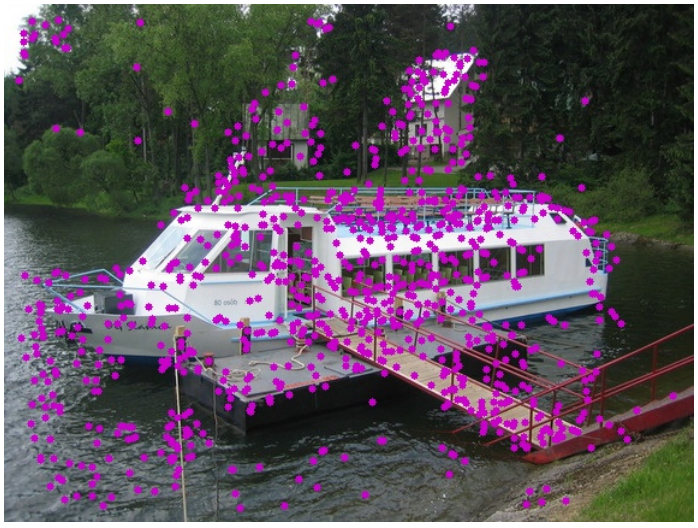
Kategorizácia obrazu

2.1 Príznaky obrazu

Prvým krokom kategorizácie obrazu je extrakcia príznakov (*features*) pomocou ktorých je charakterizovaný a popisovaný obrazový obsah. Tie môžu mať buď lokálny charakter alebo popisujú obrázok ako celok. V konečnom dôsledku majú príznaky charakter číselného vektoru.

Lokálne príznaky (SIFT, SURF, MSER, BRIEF, ...) popisujú jednotlivé body obrazu. Vyhľadanie odpovedajúcich diskrétnych bodov obrazu môže byť rozdelené do troch hlavných krokov [2].

Najprv sú vyhľadané *význačné body* na miestach, ktoré sú pre obraz určujúce, ako napríklad prechody, bloby alebo T-križovatky. najdôležitejšou vlastnosťou *detektoru* význačných bodov je jeho opakovateľnosť (repeatibility). Opakovateľnosť znamená, že daný detektor vie spoľahlivo vyhľadať fyzicky rovnaké význačne body bez ohľadu na rôzne podmienky pohľadu.



Obr. 2.1: Príklad detekovaných význačných bodov obrázku metódou SURF

Ďalej je okolie každého bodu je reprezentované príznakovým vektorom. Tento *deskriptor* musí mať rozlišovaciu schopnosť a zároveň robustný voči:

- šumu

- posunutiu
- geometrickým a
- fotometrickým deformáciám

Nakoniec sú vyhľadávané odpovedajúce príznakové vektory medzi odlišnými obrázkami. Hľadanie zhody je založené na vzdialenosti medzi vektormi v priestore – napr. Mahalanobisova alebo Euklidova vzdialenosť.

Dimenzia vektorov ma priamy dopad na na čas, ktorý tieto činnosti zaberú a teda nižšie dimenzie sú žiadúce pre rýchle spracovanie. Tieto nízko-dimenzionálne príznakové vektory maju vo všeobecnosti nižšiu rozlišovaciu schopnosť ako ich vysoko-dimenzionálne náprotivky.

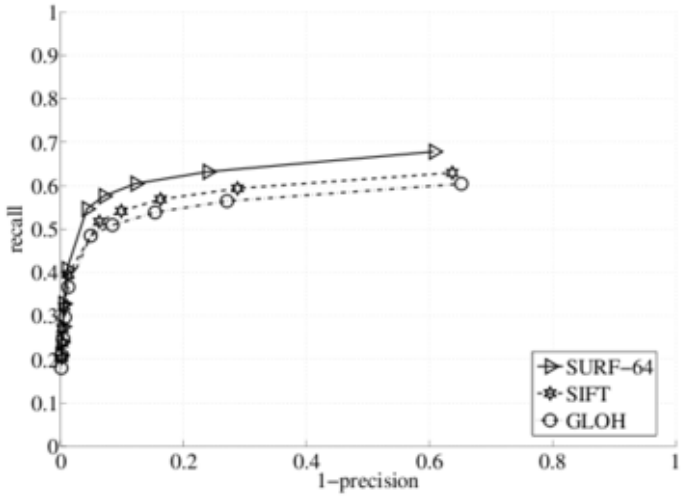
Na druhej strane existujú príznaky, ktoré *obraz popisujú ako celok*. Ide predovšetkým o príznaky založené na analýze farebnosti obrazu (farebné histogramy, korelogramy, . . .).

2.1.1 SURF

SURF (*Speeded-Up Robust Features*) patrí do kategórie lokálnych príznakov. Zahŕňa jednak detektor význačných bodov, ale aj extraktor samotného príznakového vektoru.

Cieľom autorov bolo vyvinúť detektor i deskriptor tak, aby boli v porovnaní s modernými technikami výpočtovo rýchle, ale popri tom nestrácali na výkonnosti [2]. V zmysle naplnenia týchto cieľov bolo potrebné nájsť rovnováhu medzi požiadavkami – zjednodušiť schému detekcie význačných bodov pri zachovaní jej presnosti a zároveň redukovať veľkosť príznakového vektoru pri zachovaní dostatočnej rozlišovacej schopnosti.

V experimentoch na porovnávacích sadách dát SURF detektor i deskriptor dosahoval nielen vyššiu rýchlosť, ale aj vyššiu opakovateľnosť a na základe toho aj vyššiu rozlišovaciu schopnosť.



Obr. 2.2: Porovnanie úspešnosti hľadania odpovedajúcich dvojíc bodov metódou hľadania najbližšieho suseda pre rôzne schémy extrakcie deskriptorov. Vyhodnotené na základe význačných bodov určených metódou SURF. Priemerné hodnoty boli určené podľa 8 párov obrázkov Mikolajczykovej databázy[2].

Detekcia význačných bodov a extrakcia deskriptor metódy SURF je zameraná na invarianciu voči zmene mierky obrazu a rovinnému otáčaniu. Tento fakt sa zdá byť vhodným

kompromisom medzi zložitosťou príznakov a robustnosťou voči bežne sa vyskytujúcim deformáciám.

Skosenie obrazu, anizotrópna zmena mierky a perspektívne efekty sú považované za druhotriedne efekty ktoré sú pokryté do určitej miery celkovou robustnosťou deskriptorov.

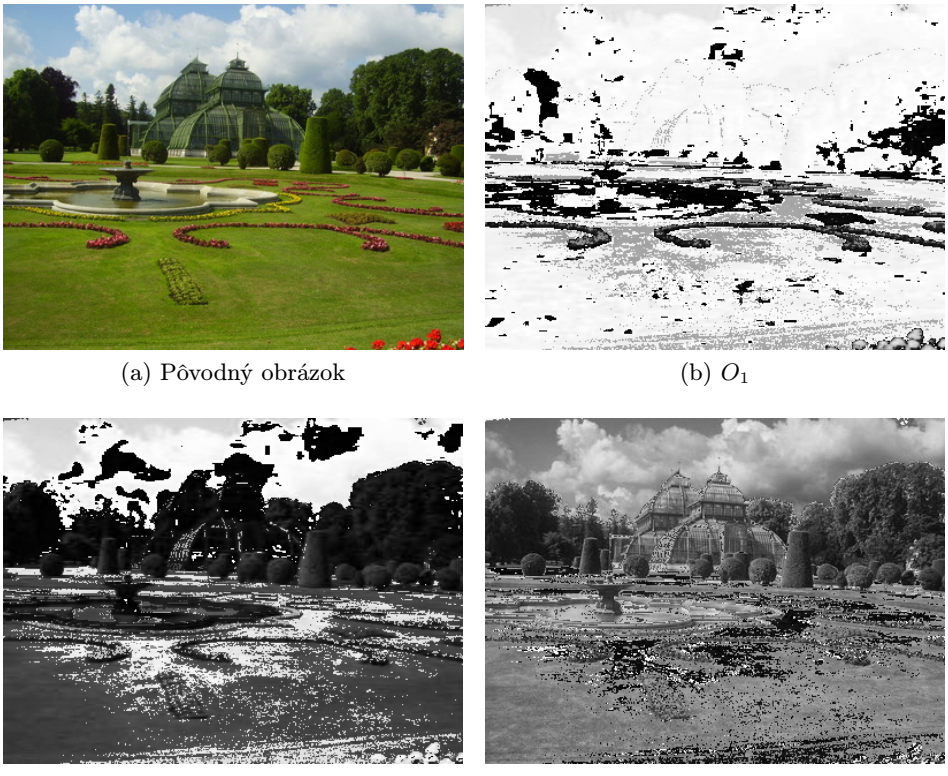
Kvôli možným fotometrickým deformáciám je zvolený jednoduchý lineárny model s ofsetom a kontrastnou zmenou. Detektor ani deskriptor metódy SURF nepoužíva informáciu o farbe obrazu.

2.1.2 Opponent color features extraction

Výber obrazových deskriptorov má veľký vplyv na na presnosť rozpoznávania. Tradične používané príznaky sú založené na intenzite obrazu vo význačných bodoch. Farba je však ignorovaná [17]. Ako už bolo spomenuté, ignorancia farebnosti obrazu vylepšuje robustnosť metódy voči fotometrickým deformáciám. Je však otázne, či by informácia o farbe nedokázala vylepšiť výsledky kategorizácie.

Bežne používané lokálne deskriptory (SIFT, SURF, ...) používajú len jedno-kanálovú infromáciu o obraze. Prirodzeným rozšírením je zahrnúť priestor protikladných farieb (*opponent color space*). Pri tomto postupe dochádza k rozloženiu tohoto priestoru do troch kanálov:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{2.1}$$



(a) Pôvodný obrázok (b) O_1

(c) O_2 (d) O_3

Obr. 2.3: Mapovanie farebého priestoru RGB do *opponent color space*.

Každý z kanálov je popísaný pomocou lokálneho deskriptoru (SIFT, SURF, ...). Informácia kanálu O_3 je rovná intenzite, kým ostatné kanály popisujú informáciu o farbe v obraze. Tieto kanály síce obsahujú informácie o intenzite, avšak nie sú invariantné voči zmenám v intenzite osvetlenia. Obrázok 2.3 demonštruje mapovanie farebného priestoru RGB do opponent color space.

2.1.3 Farebné korelogramy

Popri lokálnych príznakoch existujú metódy, ktoré sú schopné popisu obrazu ako celku. Ide o metódy založené na histogramoch. Farebné histogramy sú jednou zo starších metód indexovania a vyhľadávania založenom na obsahu obrazu [18].

Jeden zo známych problémov farebných histogramov je to, že neobsahujú žiadne informácie o priestorovom rozložení v obraze, čo veľmi obmedzuje ich rozlišovaciu schopnosť. Novšie výskumy vyvinuli lepšiu techniku s vyššou rozlišovacou schopnosťou zvanú *farebné korelogramy*. Tie preukázali výrazné vylepšenie oproti tradičným farebným histogramom.

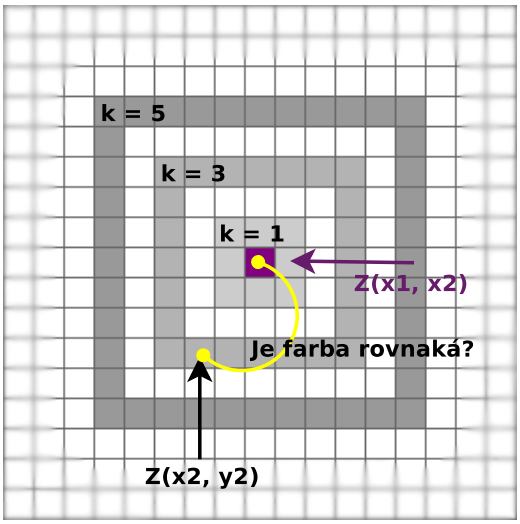
Formálne je farebný korelogram C obrazu $Z(x, y), x = 1, 2, ..., M, y = 1, 2, ..., N$ definovaný ako pravdepodobnosť P:

$$C(i, j, k) = P(Z(x_1, x_2) \in C_1 | Z(x_2, y_2) \in C_j), \tag{2.2}$$

$$k = max\{|x_1 - x_2|, |y_1 - y_2|\}, \tag{2.3}$$

kde obrázok $Z(x, y)$ je kvantovaný na fixný počet farieb C_1, C_1, ..., C_L a vzdialenosť medzi dvoma bodmi $k \in \{1, 2, ..., K\}$ je a priori fixná. Inak povedané je farebný korelogram pravdepodobnosť spojeného výskytu dvoch bodov, pričom jeden patrí k farbe C_i a druhý k farbe C_j.

Veľkosť korelogramu je $O(L^2K)$. Pre redukciu úložného priestoru sa možno zamerať na *farebné auto-korelogramy* kde $i = j$ a ich veľkosť je teda $O(LK)$ [14]. Obrázok 2.4 demonštruje konštrukciu farebného auto-korelogramu.



Obr. 2.4: Konštrukcia farebného auto-korelogramu.

2.2 Vizuálny slovník

Automatická kategorizácia podľa obrazového obsahu, či vyhľadávanie fotografií na jeho základe predstavujú paralelu voči úlohám spracovania prirodzeného jazyka, kde sa na základe textového obsahu snažíme určiť, o čom ten-ktorý článok či dokument pojednáva.

Jednotlivé fotografie predstavujú *dokumenty* a deskriptory význačných bodov prestavujú *vizuálne slová*. Množina všetkých vizuálnych slov tvorí *vizuálny slovník*. Podľa výskytu vizuálnych slov fotografiu popíšeme a pri požiadavku na kategorizáciu fotografie je na základe tohto popisu určený *tag* – čiže kategória.

Existujú dva extrémne prístupy k vytváraniu vizuálneho slovníku a teda aj k jeho používaniu[7]. Jeden extrém je ten, že každý deskriptor kategorizovanej fotografie je porovnávaný so všetkými deskriptormi trénovacej sady dát. To je dosť nepraktické, keďže počet trénovacích deskriptorov je obrovský (závisí na veľkosti trénovacej sady, ale vo všeobecnosti siaha rádovo na stovky tisíc). Ďalším extrémom je pokus o identifikovanie malého počtu veľkých „zhlukov“, ktoré majú veľkú rozlišovaciu silu (avšak vyššie výpočtové náklady). V reálnych aplikáciach existuje snaha o kompromis medzi presnosťou a výpočtovou efektivitou, ktorý sa dosahuje strednou veľkosťou zhlukovania.

2.2.1 Term frequency × inversed document frequency

Technika váhovania *tf-idf* je veľmi často používaná v oblasti získavaní informácií a dolovaní textov[19]. Táto váha je štatisticky určená pre vyjadrenie faktu, ako je určité slovo dôležité v rámci kolekcie dokumentov. Významnosť slova rastie s počtom výskytov tohto slova, ale zároveň je ovplyvnená frekvenciou jeho výskytu v sade dokumentov. Napríklad slovo, ktoré sa vo veľkom počte vyskytuje v dokumentoch istej témy, je pre určenie tejto témy významné. Ak sa však toto slovo vyskytuje často v rôznych dokumentoch rôznych kategórií (napr. predložky, spojky, ...), nie je pre kategorizáciu významné.

Frekvencia výskytu *tf* určitého slova t_i v dokumente d_j je daná vzťahom 2.4, pričom $n_{i,j}$ je počet výskytov tohto slova v dokumente, ktorý obsahuje k slov.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2.4}$$

Inverzná frekvencia slova t_i idf_i v množine dokumentov D je definovaná vzťahom 2.5.

$$idf_i = \log \frac{|D|}{|\{d : d \in D, t_i \in d\}|} \tag{2.5}$$

Toto váhovanie je rovnako použiteľné pri kategorizácii obrazových dokumentov (fotografií) na základe výskytu vizuálnych slov.

2.3 Zhlukovanie v k-dimenzionálnom priestore

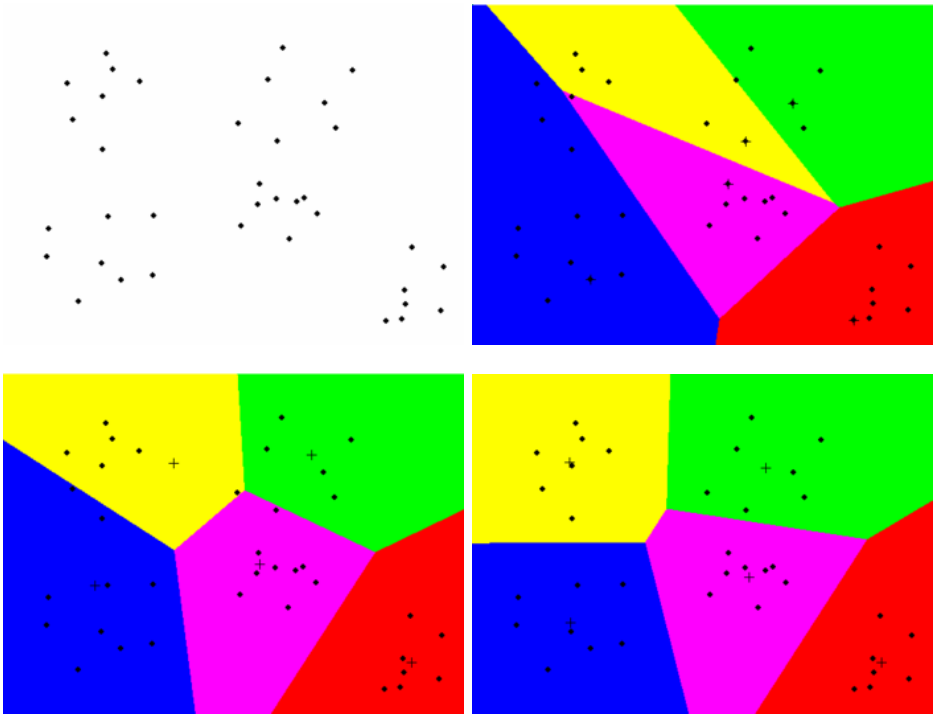
Ako už bolo uvedené v 2.2, pre vytváranie praktických vizuálnych slovníkov je vhodné deskriptory trénovacích dát zoskupiť do optimálne zvoleného počtu zhlukov. Stredy týchto zhlukov budú následne reprezentovať vizuálne slová.

Úloha hľadania podobnosti medzi prvkami trénovacej množiny a ich zaradenie prvkov s podobnými charakteristikami do zhlukov rieši počítačové učenie bez učiteľa. Systém nedostáva žiadnu informáciu o správnosti klasifikácie – jedinou informáciou môže byť počet zhlukov, do ktorých sa majú prvky trénovacej množiny zhlukovať. Prvky trénovacej množiny sú reprezentované číselnými vektormi príznakov objektov. [20]

2.3.1 K-means

Táto metóda je najznámejšia a najpoužívanejšia metóda klasifikácie príkladov trénovacej množiny do vopred daného počtu k zhlukov.

Algoritmus zaradí každý vektor do toho zhluku, ktorého stred je k danému vektoru najbližšie. Učenie prebieha týmto spôsobom (viď. 2.5):



Obr. 2.5: Postup metódy *k-means*[20]

1. náhodne je vybraných k vektorov z trénovacej množiny, ktoré sú považované za stredy zhlukov. Inicializácia stredov môže prebiehať týmto spôsobom náhodne, prípadne môže byť využitý deterministický postup – napr. algoritmus kmeans++ od autorov *Arthur a Vassilvitskii.*

2. každý prvok trénovacej množiny je priradený k najbližšiemu stredu zhluku

3. stredy zhlukov sú prepočítané a priradenie sa opakuje

4. učenie končí, ak sú všetky vektory zaradené do rovnakých zhlukov, prípadne dosiahnutím inej koncovej podmienky (napr. maximálny počet iterácií alebo dosiahnutie určitej presnosti)[20, 1]

2.3.2 Kd-tree

Vo fáze trénovania sú na základe extrahovaných lokálnych príznakov a následne pomocou algoritmu k-means vytvorené vizuálne slová tvoriace slovník. Ten je potrebné počas vlastnej kategorizácie prehľadávať a určovať, ktoré slová sa v kategorizovanej fotografii nachádzajú.

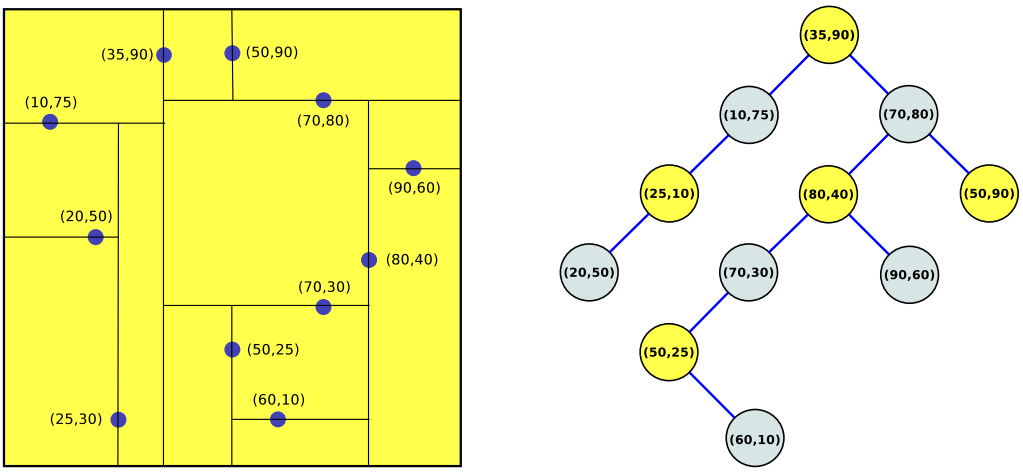
Slovník je teda potrebné reprezentovať vhodnou dátovou štruktúrou – k - *dimenzionálnym stromom.* Ide o multi-dimenzionálny vyhľadávací strom, kde k predstavuje dimenziona-

litu prehľadávaného priestoru[3]. Používa sa pre ukladanie informácií, pre ktoré je potrebné asociatívne vyhľadávanie v kolekcii záznamov. Každý záznam je usporiadaná *k-tica* $v_1, v_2, \ldots, v_k$ hodnôt, ktoré sú kľúčmi a atribútmi záznamu.

Každý uzol stromu pozostáva z:

- dvoch ukazateľov na synovské uzly
- vyhľadávacieho kľúča – jeden či viac čísel s plávajúcou desatinnou čiarkou, ktoré reprezentujú umiestnenie záznamu či bodu v priestore
- dodatočné informácie (napr. pomenovanie záznamu a pod.)

Príklad štruktúry k-dimenzionáneho stromu (pre $k = 2$) je uvedený na 2.6. Pri konštrukcii takéhoto binárneho stromu je vybraný jeden z rozmerov priestoru vybraného bodu, ktorý delí zvyšné body na dve skupiny. Napríklad pre koreňový uzol vyberieme os x – všetky body, ktoré majú menšiu hodnotu x-ovej súradnice, budú patriť do ľavého podstromu a zvyšné do pravého podstromu. V nasledujúcej úrovni je priestor delený na základe y-ových súradníc atď.[6]



Obr. 2.6: Príklad *kd-tree* pre $k = 2$[6]

Požiadavok na vyhľadanie záznamu sa nazýva *dopyt – query*[3]. Pri dopyte sú špecifikované isté podmienky, ktorých splnenie je požadované od vyhľadávaného záznamu. Na základe týchto požiadavkov rozlišujeme dopyty na prienik (intersection queries), dopyty na úplnú či čiastočnú zhodu, dopyty na určité okolie (region queries) alebo na *najbližšieho suseda – neares neighbour queries*.

Posledný zmienený dopyt je relevantný pre úlohu kategorizácie obrazu, keďže vo fáze samotnej kategorizácie potrebujeme vyhľadať výskyt vizuálnych slov v obraze (záznamov kd-tree). Hľadáme slovo (prípadne slová), ktoré sú najbližšie – a teda najpodobnejšie – extrahovanému príznakovému vektoru kategorizovanej fotografie. Hľadanie najbližšieho suseda má empiricky určenú časovú zložitosť $O(\log n)$ pre n záznamov.

Ak máme danú funkciu vzdialenosti dvoch záznamov D, kolekciu záznamov B v k-dimenzionálnom priestore a bod P v ňom, požadovaný *najbližší sused* Q *je definovaný* ako:

$$(\forall R \leq B)\{(R \neq Q) \Rightarrow [D(R, P) \geq D(Q, P)]\} \tag{2.6}$$

2.3.3 Soft assigment

Pri kategorizácii fotografie dochádza ku kvantifikovaniu príznakových vektorov na množinu vizuálnych slov, ktorá je tradične vytváraná cestou učenia bez učiteľa (k-means)[12]. Podobné deskriptory teda končia „tvrdo priradené“ – hard assigned – k rovnakému vizuálnemu slovu. Tento prístup je problematický z dôvodu veľkého množstva možných vzhľadov rovnakého objektu, rôznych uhlov pohľadu, svetelných podmienok, ale i kvôli samotnej variácii objektov tej istej triedy.

Jeden z možných riešení tohto problému je tzv. „mäkké priradenie“ – *soft assignment*[5]. Pri tejto metóde namiesto hľadania jedného najbližšieho suseda (viď. 2.3.2) vyhľadáme najbližších susedov niekoľko.

Významnosť jednotlivých susedov a teda aj ich príspevok ku konečnej klasifikácii môže byť ďalej váhovaný. Možná váhovacia rovnica je vyjadrená vzťahom 2.7 z [12]

$$w_i = \frac{\exp\left(\frac{||x - c_i||_2}{m}\right)}{\sum_{j=1}^{k} \exp\left(\frac{||x - c_j||_2}{m}\right)} \tag{2.7}$$

kde c_i predstavuje najbližšieho suseda z k vyhľadaných najbližších susedov bodu (príznakového vektoru) x. $||.||_2$ je L_2 norma. Skalár m môže byť interpretovaný ako okraj hranice. Nízke hodnoty m predstavujú „ostré“ hrancie – naopak vysoké hodnoty znamenajú „mäkšiu“ klasifikáciu.

2.4 Bag Of Words a Support Vector Machines

Pri trénovaní ale i pri samotnej klasifikácii je fotografia reprezentovaná číselným vektorom, ktorý je predaný ďalej klasifikátoru. Tento vektor predstavuje histogram početnosti výskytu vizuálnych slov v danej fotografii – *bag of words (BOW)*. Početnosti môžu by ďalej váhované na základe relevantonosti vizuálneho slova (napr. idf), alebo váhou na základe vzdialenosti extrahovaného deskriptoru od samotného vizuálneho slova pri použití soft assignment-u.

BOW môže byť ďalej spojený s iným číselným vektorom popisujúcim fotografiu – napr. farebným histogramom, korelogramom, ... atď. Klasifikátorom, ktorý určí na základe tohto vektoru kategóriu, môže byť neurónová sieť, systém založený na určitom štatistickom hlasovaní alebo najčastejšie *support vector machines*.

2.4.1 Support Vector Machines

SVM je technika výhodne použiteľná v oblasti klasifikácie, ktorá je považovaná za jednoduchšiu ako neurónove siete[11]. Úloha klasifikácie zahŕňa rozdelenie dát do trénovacej a testovacej sady. Každá inštancia trénovacej sady pozostáva z „cieľovej“ hodnoty (napr. čísla kategórie) a niekoľkých atribútov (napr. príznaky, pozorované premenné). Cieľom SVM je vytvoriť model založený na trénovacích dátach, ktorý je schopný predikovať cieľové hodnoty pre testovacie dáta založené na ich atribútoch.

Daná trénovacia sada dát obsahuje dvojice $(\mathbf{x}_i, y_i), i = 1, \ldots, l$ kde $x \in \mathbb{R}^n$ a $\mathbf{y} \in 1, -1^l$. Použitie SVM vyžaduje potom riešenie optimalizačného problému, pri ktorom sú trénovacie vektory $\mathbf{x_i}$ namapované do vyššie-dimenzionáneho priestoru funkciou ϕ. SVM hľadá lineárnu deliacu nadrovinu v tomto vyššie-dimenzionálnom priestore. $K(\mathbf{x_i}, \mathbf{x_j}) \equiv \phi(\mathbf{x_i})^T \phi(\mathbf{x_j})$ sa nazýva *jadrom*. SVM používa tieto štyri základné *jadrá* – *kernel-y*:

- lineárne: $K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i}^T \mathbf{x_j}$.
- polynomiálne: $K(\mathbf{x_i}, \mathbf{x_j}) = (\gamma \mathbf{x_i}^T \mathbf{x_j} + r)^d, \gamma > 0$.
- s radial basis funkciou (RBF): $K(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma ||\mathbf{x_i} - \mathbf{x_j}||^2), \gamma > 0$.
- sigmoid: $K(\mathbf{x_i}, \mathbf{x_j}) = \tanh(\gamma \mathbf{x_i}^T \mathbf{x_j} + r)$.

Kde γ, r, d sú parameter jadra. Tieto parameter by mali byť vhodne zvolené vzhľadom na *cross validáciu*, ktorej cieľom je identifikovať tieto parametre pre primerané určovanie cieľovej hodnoty – neznámych dát.

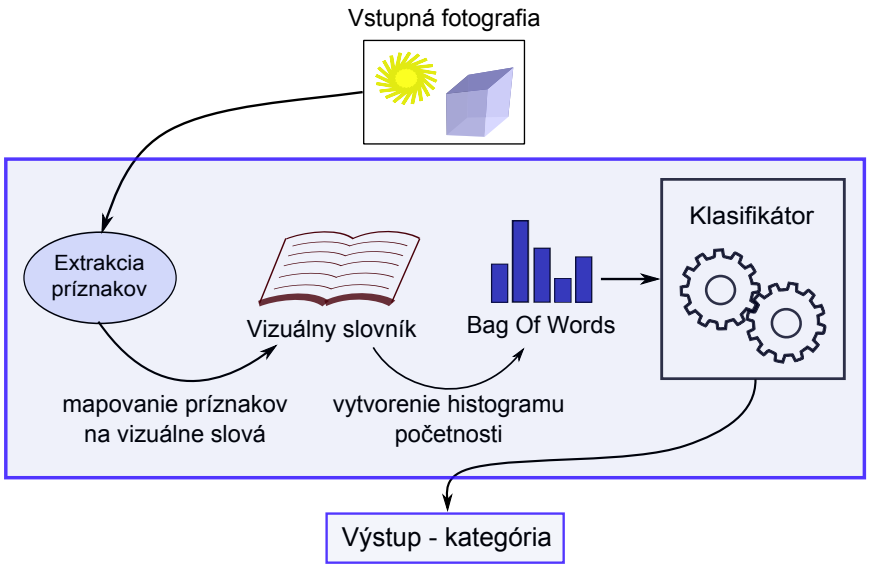
Vo všeobecnosti je vhodné zvoliť RBF jadro, keďže toto jadro mapuje vzorky do viacdimenzionálneho priestoru nelineárne, takže sa napríklad na rozdiel od lineárneho jadra dokáže vysporiadať s nelineárnymi vzťahmi medzi označením kategórií a atribútami dát. RBF model je tiež jednoduchší ako napríklad polynomiálny model. Existujú samozrejme aj prípady, kedy je vhodné voliť iný typ jadra. Lineárne jadro je napríklad vhodné v prípade veľkého množstva použitých príznakov. v tomto prípade nelineárne mapovanie výkonnosť nevylepší. Taktiež je pri jeho použití nutné optimálne urči menší počet parametrov jadra.

Kapitola 3

State of the Art

V tejto kapitole budú popísané typické prístupy k úlohám kategorizácie obrazových dát. Tie je možné rozšíriť o pokročilejšie metódy pre zvýšenie robustnosti či efektivity riešenia.

3.1 Tradičný postup kategorizácie



Obr. 3.1: Schéma systému kategorizácie obrazu.

Systémy kategorizácie fotografií podľa obrazového obsahu sa v súčasnej dobe riadia modelom, ktorý je znázornený na obrázku 3.1. *Vstupom* je fotografia, ktorá nenesie žiadne dodatočné informácie o svojom obsahu. *Výstupom* je informácia o jedenej alebo viacerých kategóriách, do ktorých fotografia patrí. Tejto informácii sa v anglickej literatúre hovorí aj *tag* – čiže určitý „štítok“ fotografie. Proces samotnej kategorizácie obsahuje tieto základné kroky:

1. Načítanie fotografie, detekcia a extrakcia *príznakov* význačných bodov.
2. Priradenie každého príznakového vektoru ku konkrétnemu slovu *vizuálneho slovníka* (na základe vzdialenosti vektorov).

3. Vytvorenie histogramu početnosti výskytu vizuálnych slov vo fotografii – *Bag Of Words* (skr. *BOW*), pričom významnosť slov môže byť ovplyvnená váhami.

4. predanie BOW – ktorý môže byť opäť chápaný ako vektoru – *klasifikátoru*. Ten mapuje vektor popisujúci celú fotografiu na konkrétne číslo kategórie (resp. na množinu kategórií). Toto mapovanie môže byť taktiež charakterizované pravdepodobnosťou, s ktorou klasifikátor začlenil fotografiu do danej skupiny.

3.2 Inovácie štandardného postupu

3.2.1 Delenie obrazu

Pri tradičnom prístupe dochádza k detekcii význačných bodov v celej fotografii. Následne sú v týchto bodoch extrahované príznakové vektory. Alternatívne prístupy obmedzujú detekciu význačných bodov na určitú vymedzenú časť fotografie.

Extrakcii príznakov (či už lokálnych alebo výpočtu farebných histogramov/korelogramov) môže predchádzať vyhľadanie *oblasti záujmu – region of interest (ROI)*[4]. Hľadáme teda oblasť resp. časť fotografie „kde niečo je“. Obmedzenie extrakcie príznakov či výpočtu farebných histogramov len na ROI môže mať za následok nielen zrýchlenie zvyšných výpočtov ale aj zvýšenie úspešnosti klasifikácie. Bežne sa využívajú oblasti obdĺžnikového tvaru.

Trénovacie fotografie musia byť pre takúto klasifikáciu predspracované (anotované vymedzením ROI). V praxi to znamená, že trénovaciu množinu rozdelíme na malé podmnožiny (cca. 3-4) fotografií obsahujúce rovnaký alebo veľmi podobný objekt. V každej takejto pod-množine musí existovať jedna anotovaná fotografia s vyznačeným ROI. Ostatné trénovacie fotografie danej množiny budú anotované automaticky hľadaním podobnosti – *angl. matching* – s využitím lokálnych príznakov.

Pri klasifikácii istých obrázkov môže byť s výhodou použitá opačná technika – tzv. *dense sampling*, ktorá je demonštrovaná na obrázku 3.2. Obraz je delený na malé pravidelné časti v ktorých sú extrahované lokálne príznaky[16]. Toto delenie má za následok rovnomernejšie rozprestrenie význačných bodov po fotografii. To môže byť užitočné, ak sú pre určitú fotografiu charakteristické homogénne plochy (napr. nebo, voda, more, pláž, . . .), keďže klasická detekcia význačných bodov takého plochy diskriminuje.

Obr. 3.2: Dense sampling.

V praxi sa ďalej využíva delenie obrazu na niekoľko častí s oddeleným spracovaním (viď. 3.3). Používajú sa delenia 2x2 alebo 3x1[16]. Pre tieto časti je vypočítaný BOW či iné (napr. farebné) histogramy úplne oddelene. Informácia sa spojí pred vstupom do klasifikátoru či až na základe jeho rozhodnutia o kategóriách jednotlivých častí. Delenie obrazu 3x1 – t.j. tri rovnobežné horizontálne pásy môže byť obzvlášť prínosné, keďže vo veľmi veľa fotografiách

je možné pozorovať takéto (alebo podobné) rozloženie objektov – v spodnej časti zem, voda či pláž, v strednej nejaký samotný objekt a v najvyššej časti typicky obloha, stena alebo stromy.

Obr. 3.3: Delenie obrazu pred spracovaním podľa pravidelnej mriežky.

3.2.2 Reprezentácie vizuálneho slovníka

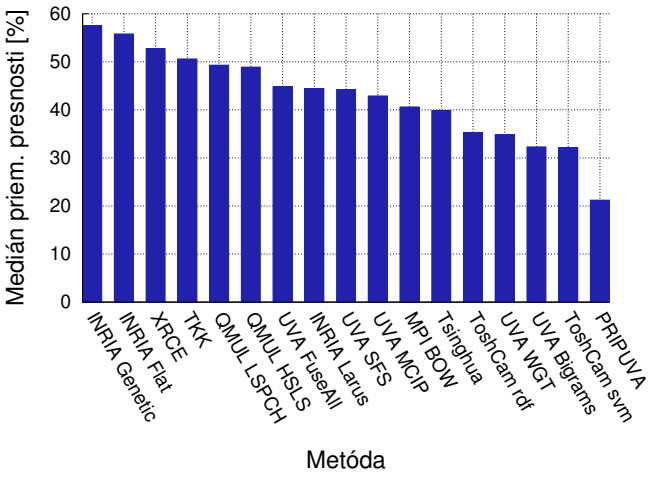
Vizuálny slovník býva najčastejšie reprezentovaný viac-dimenzionálnym stromom (viď. 2.3.2). Alternatívne môže byť namiesto jedného stromu vytvorený index, ktorý je tvorený viacerými náhodne generovanými stromami – *randomized trees* pričom vznikajú tzv. *randomized forests*. Tieto stromy môžu byť prehľadávané paralelne, čo môže zvýšiť rýchlosť vyhľadania vizuálneho slova.

Kd-tree môže byť taktiež vytvorený aplikáciou *hierarchického k-means*, pri ktorom je vytváraná stromová štruktúra už pri samotnom zhlukovaní.[1, 13, 4]

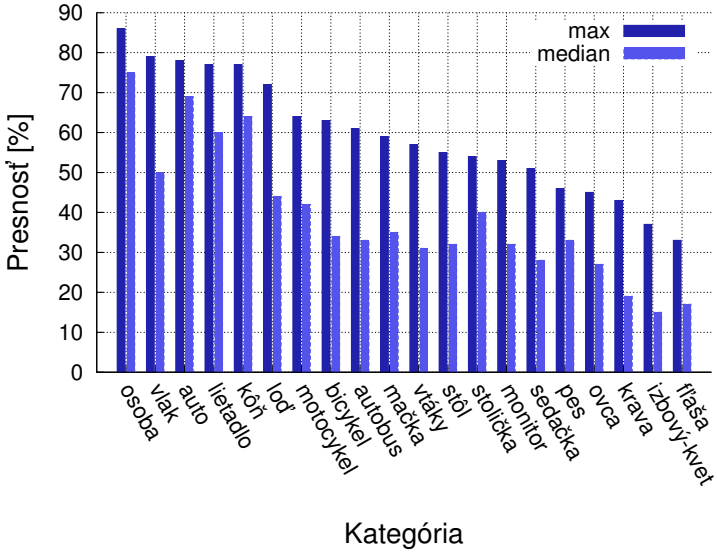
3.3 The Pascal Visual Object Classes Challenge

Táto súťaž je zameraná na hodnotenie najnovších metód a prístupov spracovania obrazových dát. Zameriava sa na dve konkrétne oblasti – a to *klasifikáciu* fotografií a *detekciu* objektov v obraze. Od roku 2006 je každoročne zverejnená testovacia sada fotografií, na ktorej si tímy z celého sveta porovnávajú svoje aplikácie. Obr. 3.4 ukazuje výsledky najlepších tímov z roku 2007 (kategorizácia prebiehala do 20-tich uvedených kategórií). Táto súťaž je zameraná na systémy, ktoré dokážu klasifikovať obraz do viacerých kategórii. Výsledky jednotlivých tímov hodnotené na základe priemernej presnosti kategorizácie, médiánu priemernej presnosti cez všetky kategórie a taktiež na základe precision/recall krivky.[9]

Zdrojom testovacích dát je anotovaná databáza fotografií *flickr*. Na rozdiel od iných databáz ako napr. Caltech101[10] alebo Caltech256 má väčšiu rozmanitosť fotografií v rámci jednotlivých kategórií (rozne objekty, ich rozmiestnenie v scéne, natočenie, počet, . . .).[9]



(a) Sumárne výsledky podľa použitých metód. Za každú metódu je vybraný medián cez všetky kategórie



(b) Sumárne výsledky klasifikácie podľa kategórií. Ukázané sú maximálne hodnoty priemernej presnosti a medián priemernej presnosti zo všetkých použitých metód.

Obr. 3.4: Výsledky VOC Challenge z roku 2007 [9]

Kapitola 4

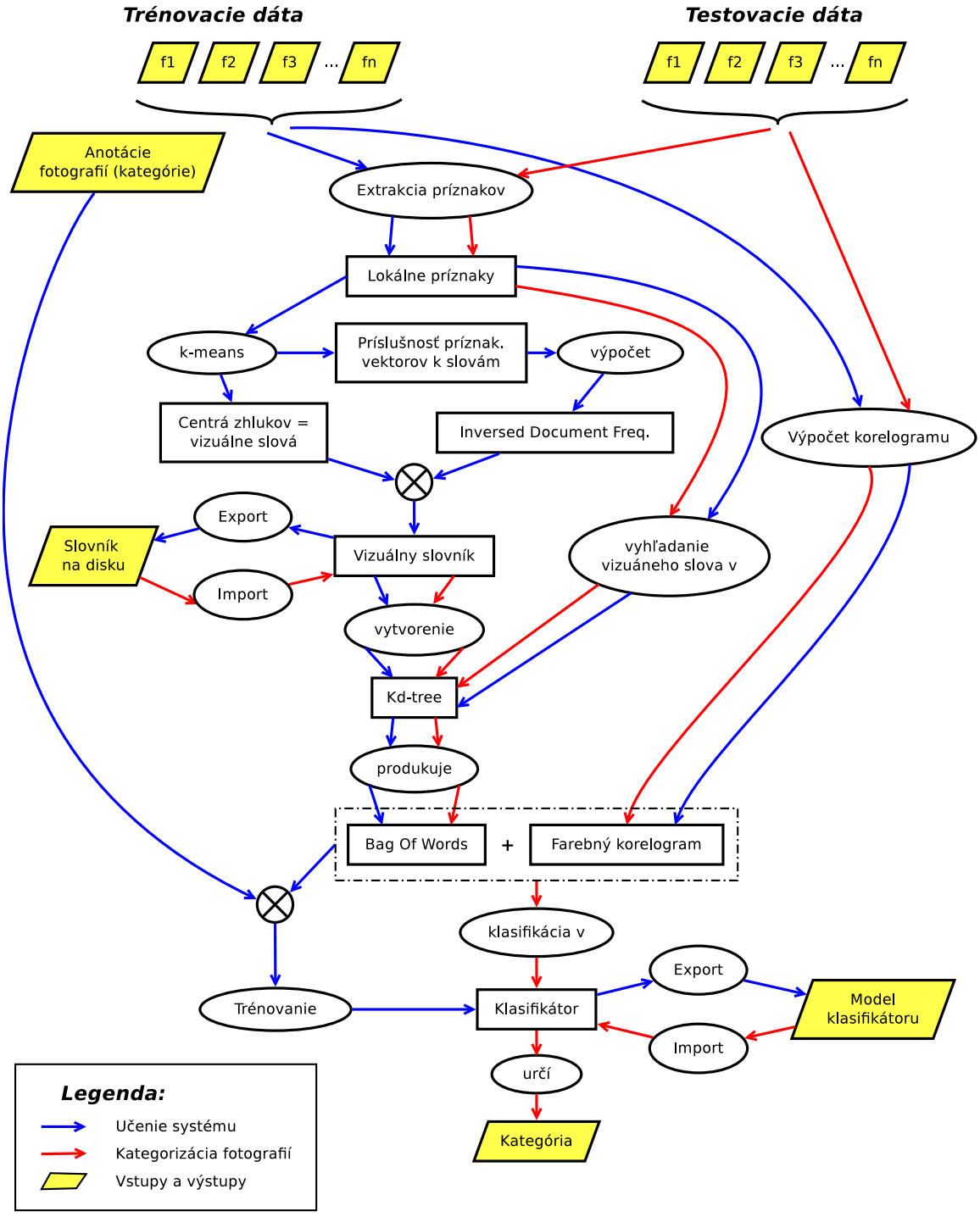
Návrh riešenia

Na začiatku vypracovávania návrhu riešenia tejto práce boli identifikované základné časti výsledného systému, vymedzená ich činnosť, určené vstupy a výstupy. V rámci systému klasifikácie obrazu je možné sledovať tieto hlavné fázy:

1. fáza *učenia*
   - *vstupy:* trénovacie fotografie, anotácie fotografií (informácia o kategórii)
   - *činnosti:* extrakcia príznakov, zhlukovanie do vizuálnych slov, tvorba a export vizuálneho slovníku, trénovanie a export klasifikátoru
   - *výstupy:* súborý obsahujúce exportovaný vizuálny slovník a model klasifikátoru
2. fáza vlastnej *klasifikácie*
   - *vstupy:* testovacie fotografie, súbory obsahujúce vizuálny slovník a natrénovaný model klasifikátoru
   - *činnosti:* extrakcia príznakov, vyhľadanie príznakových vektorov vo vizuálnom slovníku a výsledná klasifikácia fotografie využitím klasifikátoru
   - *výstupy:* kategórie, ktoré pre dané fotografie určí klasifikátor

V tejto práci sú použité nasledujúce metódy a algoritmy klasifikácie obrazu:

- lokálne príznaky *SURF*, extrakcia týchto príznakov v *opponent color space*
- reprezentácia fotografie pomocou *bag of words* a *farebných korelogramov*
- vytvorenie *vizuálneho slovníku* zhlukovaním metódou k-means a reprezentácia pomocou náhodne vytváranej sady k-dimenzionálnych stromov
- váhovanie relevantonosti vizuálnych slov pomocou *inversed document frequency*
- klasifikácia s využitím *support vector machines*
- *delenie obrazu* pri extrakcii lokálnych príznakov a výpočte korelogramov
- *mäkké priradenie* vizuálnych slov k extrahovanému príznakovému vektoru



Obr. 4.1: Vývojový diagram aplikácie tejto práce.

4.1 Podrobný návrh systému

Základná kostra tejto práce vychádza z článku Š. Rosu – *Automatic photo tagging*[15]. Na obrázku 4.1 je pomocou vývojového diagramu podrobnejšie popísaný celý systém klasifikácie. Jednotlivé fáze (učenie a vlastná kategorizácia), vstupy a výstupy sú oddelené farebne

(viď. legenda). V tejto sekcii bude tento návrh podrobnejšie popísaný.

4.1.1 Extrakcia príznakov

Prvým krokom spracovania každej fotografie je extrakcia príznakov. V tejto práci sú zvolené tri druhy príznakov.

Pre detekciu význačných bodov a následne extrakciu lokálnych príznakov je použitá metóda *SURF* (viď. 2.1.1), ale taktiež varianta s využitím transformácie obrazu z priestoru RGB do *opponent color space* (viď. 2.1.2) a následná extrakcia príznakov SURF v každom kanály tohto priestoru.

Posledný tretí typ príznakov, použité v tejto práci sú *farebné korelogramy* (viď. 2.1.3). Korelogram je spojený spolu s BOW do jedného spoločného vektoru pred vstupom do klasifikáciu.

Pre zvýšenie robustnosti použitých metód získavania obrazových príznakov je využitá možnosť delenia fotografie na časti výpočtom príznakov v každom regióne oddelene. To znamená, že počas popisu fotografie vzniká niekoľko histogramov (BOW- s alebo farebných korelogramov), ktoré sú opäť spojené pred vstupom do klasifikátoru.

Inou možnosťou kombinácie viacerých príznakov (resp. príznakov získaných z rôznych častí obrazu) by bolo trénovanie viacerých klasifikátorov – pre každý typ histogramu zvlášť – a následne pri kategorizácii štatisticky spojiť výstupy jednotlivých klasifikátorov. Tento spôsob však nie je využitý, keďže spájanie na úrovni vstupu do klasifikátoru je jednoduchšie a naviac pri ňom nie je potrebné riešiť situácie, keď by jednotlivé klasifikátory produkovali rozporuplné výsledky.

4.1.2 Vizuálny slovník

Vizuálny slovník tejto práce pozostáva z vizuálnych slov, ktoré sú reprezentované jednak k-dimenzionálnym vektorom (pričom k je dĺžka príznakového vektoru) ale taktiež významnosťou slova – váhou. Je použitá metóda *IDF* (viď. 2.2.1), ktorá reflektuje dôležitosť daného slova v množine dokumentov.

K-dimenzionálne vektory popisujúce vizuálne slová sú získané zhlukovaním metódou *k-means* z extrahovaných príznakových vektorov trénovacích fotografií. Popri vektorovej kvantifikácii sú zároveň získané aj príslušnosti jednotlivých príznakových vektorov k centrám zhlukov – vizuálnym slovám. Keďže je taktiež neustále udržovaná informácia o tom, ktorý príznak pochádza z ktorej fotografie, je možné vypočítať inverznú frekvenciu slova v dokumentoch (IDF).

Po výpočte vizuálneho slovníku je jeho obsah exportovaný do súboru, aby mohol byť využitý pri klasifikácii bez nutnosti jeho ďalšieho zostavovania.

Vizuálny slovník je nevyhnutnou súčasťou pre vytvorenie histogramu výskytu slov – BOW (viď. 2.4). Príznakové vektory sú v slovníku vyhľadávané pre určenie slov, ktoré ich budú reprezentovať. Keďže ide o vyhľadávanie vo viac-dimenzionálnom priestore, je pre tento účel vhodné využiť stromovú štruktúru *k- dimenzionálny strom* (viď. 2.3.2), ktorý je vytvorený na základe obsahu slovníka a je jeho vnútornou reprezentáciou. Pre vyhľadávanie je použitý algoritmus *najbližšieho suseda*.

Použitý ke variant k-dimenzionálneho stromu (viď. 3.2.2), ktorý umožňuje paralelné spracovanie. Tzv. *randomized trees*, predstavujú náhodné rozdelenie solitérnej stromovej štruktúry na viacero menších stromov, ktoré je možné prehľadávať súčasne.

Voliteľnou súčasťou systému je taktiež využitie mäkkého priradenia (*soft assignemnt*) viacerých najbližších vizuálnych slov ku každému extrahovanému príznakovému vektoru (viď. 2.3.3).

4.1.3 Klasifikátor

Koncovým modulom trénovania systému ale aj samotnej klasifikácie je *klasifikátor*. Vstupom do tohoto modulu je vždy histogram (resp. množina histogramov), ktorý vzniká spojením BOW a farebného korelogramu. Každý histogram teda reprezentuje jednu fotografiu.

Vo fáze trénovania prináleží histogramu taktiež anotácia o kategórii fotografie, ktorú reprezentuje. Účelom vstupu takto anotovaných histogramov je trénovanie klasifikátoru a vytvorenie jeho modelu. Ten je možné trvalo uložiť na disk – exportovať do súboru.

Tento model je vo fáze klasifikácie fotografií importovaný a využitý pre priradenie kategórie ku každej fotografii. Tá je charakterizovaná obdobným (spojeným) histogramom, aké boli použité pri trénovaní. Keďže sa táto práca zaoberá kategorizáciou len na základe obrazového obsahu, testovacie dáta (klasifikované fotografie) neobsahujú žiadne dodatočné anotácie.

Úlohu klasifikátoru tradične zohrávajú neurónove siete, support vector machines (SVM) či iné techniky založené napríklad na štatistickom spracovaní (viď. 2.4.1). V tejto práci sú kvôli spoľahlivosti a jednoduchosti využité *support vector machines*. Pri použití SVM je potrebné vyriešiť niekoľko otázok, ako napríklad výber vhodného jadra – *kernel-u* – (v zásade lineárneho alebo RBF), či vhodné nastavenie konštánt klasifikátoru. Riešenie týchto otázok bude cieľom experimentovania a teda hodnotenia SVM klasifikátoru na základe *cross-validácie* pri trénovaní a dosiahnutých výsledkov pri klasifikácii.

4.2 Návrh testovania

4.2.1 Výber dát

Pre testovanie kategorizácie fotgrafií – či už základného riešenia alebo pokročilejších prístupov – je potrebná množina trénovacích a testovacích fotografií. Výber fotografií bol založený na známych anotovaných obrazových databázach, ktoré sú určené pre testovanie uloh v oblasti počítačového videnia alebo úloh samotnej kategorizácie obrazu.

Použité sú databázy *Caltech101*[10] a súbor testovacích dát *VOC Challenge*[8] z roku 2007, ktoré sú voľne dostupné a ktorých zdrojom je server pre zdieľanie fotografií *Flickr*[1].

Z týchto databáz sú zostavené tri sady trénovacích a testovacích množín:

1. Podmnožina databázy Caltech101 (vynechané sú kategórie s príliš nízkym počtom fotografií), ktorá obsahuje 68 kategórií – každá okolo 70-100 fotografií a je určená pre experimentovanie s jednotlivými vylepšeniami základného riešenia.

2. Podmnožina testovacej sady VOC Challenge (vynechané sú fotografie, ktoré sú anotované viacerými kategóriami, keďže v tejto práci je použitá klasifikácia do jedinej kategórie). Obsahuje 20 kategórií – každá po cca 100 fotografií. Táto sada bude použitá na približné porovnanie s výsledkami VOC Challenge 2007.

3. Výber 20 kategórií databázy Caltech101, ktoré sa približne zhodujú s kategóriami použitými vo VOC Challenge 2007 pre ďalšie približné porovnanie.

[1] `www.flickr.com`

Porovnanie výsledkov tejto práce s výsledkami VOC Challenge bude pochopiteľne len približné, keďže pri sade 2 vylúčením fotografií, ktoré sú anotované viacerými triedami je úloha klasifikácie o niečo zjednodušená. Pri sade 3 je zasa využitá databáza Caltech101, ktorá podľa [9] disponuje nižšou rozmanitosťou objektov, natočení a obsadení scény. Keďže ale cieľom tohto testovania nie je „súperiť“ s existujúcimi metódami, ale len overiť výsledky tejto práce, mali by uvedené sady fotografií postačovať.

4.2.2 Priebeh testov

Testovanie výsledkov tejto práce bude prebiehať inkrementálne. To znamená, že najprv bude experimentálne určené vhodné nastavenie základného riešenia (k-means, veľkosť slovníku, parametre SVM). Naň budú postupne aplikované pokročilejšie techniky (s rôznymi parametrami) popísané v tejto práci. Týmto postupom bude určené najlepšie riešenie, ktoré bude následne porovnané s výsledkami VOC Challenge 2007.

Jednotlivé experimenty budú vyhodnotené na základe presnosti (*precission*) p a odozvy (*recall*) r. Vzorce (viď. 4.1) uvádzajú výpočet týchto hodnôt pre požiadavok o nájdenie všetkých fotografií danej kategórie.

$$p = \frac{correct\_results}{all\_results\_for\_cathegory} \qquad r = \frac{correct\_results}{all\_files\_in\_cathegory} \tag{4.1}$$

Hodnoty presnosti a odozvy je možné určiť pre každú kategóriu danej testovacej množiny. Presnosť je teda reprezentovaná podielom počtu správnych výsledkov pre danú kategóriu a počtu fotografií, ktoré boli do tejto kategórii priradené. Pri výpočte odozvy je menovateľ nahradený počtom všetkých fotografií, ktoré sa v testovacej sade vyskytujú v danej kategórii. Na základe týchto údajov bude určená *priemerná presnosť* a *medián presnosti* cez všetky uvedené kategórie. Keďže je v tejto práci použitá klasifikácia fotografií vždy do práve jednej kategórie, priemerná odozva bude mať rovnakú hodnotu ako priemerná presnosť. Toto štatistické vyhodnotenie bude základom pre porovnanie jednotlivých metód

Kapitola 5

Implementácia

Implementácia práce vychádza z návrhu uvedeného v *Kapitole 4*. Keďže už samotný návrh pozostáva z 2 fáz (učenie systému a vlastná kategorizácia), výstupom implementácie sú dve aplikácie *teacher* a *tagger*, ktoré zdieľajú rovnaké moduly a triedy.

Aplikácie sú napísané v jazyku C++. Ten bol zvolený hlavne kvôli faktu, že knižnica *OpenCV*, ktorá je vo veľkej miere v implementácii využitá, ponúka rozhrania v jazykoch C, C++ a Python, pričom C++ predstavuje vhodný kompromis medzi pohodlnosťou vývoja a rýchlosťou spracovania a naviac umožňuje objektovú orientáciu programovania.

5.1 OpenCV

Knižnica OpenCV (*Open source Computer Vision*) je voľne dostupná knižnica, ktorá implementuje celú škálu algoritmov a postupov počítačového videnia. Zahŕňa v sebe taktiež rozhrania pre čítanie/zápis obrazových dát a definuje potrebné dátové štruktúry pre ich vhodnú reprezentáciu v pamäti. Informácie o jednotlivých triedach, funkciách a štruktúrach boli čerpané z oficiálnej dokumentácie.[1]

Pôvodne bola napísaná v jazyku C, ale obsahuje plne funkčné objektové rozhranie pre C++ a Python. Dátové štruktúry poskytujú vhodné metódy pre prevod dát z/do štruktúr knižnice *STL – Standard Template Library* jazyka C++, čo umožňuje pohodlnú a pružnú prácu s dátami.

Väčšina programovej časti tejto práce bola implementovaná ešte pred zverejnením verzie *2.2* a využíva teda verziu *2.1*. Z tohoto dôvodu je vizuálny slovník, tvorba BOW a práca spojená s k-dimenzionálnym stromom programovaná samostatne, bez využitia noviniek ponúkaných vo verzii 2.2. Avšak pokročilejšie techniky, ktoré zvyšujú robustnosť základného riešenia a sú cieľom experimentov, využívajú nové triedy a metódy verzie 2.2.

5.1.1 Detekcia a extrakcia príznakov

Ako už bolo uvedené, knižnica OpenCV ponúka rozhranie pre čítanie a zápis obrazových dát. Na začiatku behu aplikácie sú teda načítané vstupné fotografie pomocou funkcie `imread()`, ktorá vracia maticu (trieda `Mat`) bodov. Na túto maticu je následne aplikovaný detektor a neskôr extraktor význačných bodov.

Knižnica OpenCV implementuje viacero druhov lokálnych príznakov (SURF, SIFT, MSER, STAR, ...). V tejto práci sú kvôli výhodným vlastnostiam využité príznaky SURF implementované triedou `SURF`. Kvôli menšej pamäťovej náročnosti je použitá varianta príznakových vektorov dĺžky 64. OpenCV obsahuje univerzálne rozhranie pre extrakciu a de-

tekciu lokálnych príznakov, ktoré zapúzdruje detektory a extraktory konkrétnych metód. Tieto univerzálne rozhrania (triedy) sú ich schopné adaptovať na obraz špecifickým spôsobom.

V tejto práci sú použité dve konkrétne rozhrania. Trieda `GridAdaptedFeatureDetector` delí obraz podľa zvolenej rovnomernej mriežky. V jednotlivých bunkách potom nezávisle detekuje lokálne príznaky na základe metódy, ktorú adaptuje (v tejto práci SURF). Ďalej je využité univerzálne rozhranie `OpponentColorDescriptorExtractor`, ktoré prevádza obraz do opponent color space a následne extrahuje lokálne príznaky v jeho jednotlivých kanáloch.

5.1.2 Zhlukovanie a vyhľadávanie v k-dimenzionálnom priestore

Extrahované príznaky sú zoskupené do jedinej matice – trieda `Mat` – ktorá je ďalej predmetom zhlukovania. Knižnica OpenCV priamo implementuje metódu *k-means* funkciou `kmeans()`. Funkcia určí stredy výsledných zhlukov, ktorých počet (čiže veľkosť slovníku) je definovaný parametrom. Taktiež príslušnosť príznakových vektorov ku konkrétnym zhlukom. Pomocou flag-u je možné určiť spôsob inicializácie centier zhlukov. Použitá je voľba `KMEANS_PP_CENTERS`, ktorá spôsobí použitie algoritmu k-means++ pre vhodné rozostavenie centier do priestoru. Pomocou štruktúry `TermCriteria` je definovaná ukončujúca podmienka – maximálny počet iterácií, ktorý bude určený na základe experimentov.

Uloženie vizuálneho slovníku – čiže *k-dimenzionálny strom* – implementuje trieda `flann::Index`. Pomocou štruktúry `IndexParams` predanej pri volaní konštruktoru je možné definovať usporiadanie stromu. V tejto práci je použité náhodné vytváranej sady viacerých stromov prehľadávaných súčasne. Trieda `flann::Index` má implementuje priamo algoritmus *hľadania najbližšieho suseda* metódou `knnSearch()`, ktorá pre zadaný príznakový vektor vyhľadá podľa parametra $1 - n$ najbližších susedov. Viacerý susedia sú vyhľadávaný pri použití mäkkého priradenia *soft assignment-u*.

5.1.3 Support vector machines

Posledným veľmi podstatným modulom knižnice OpenCV, ktorý je využitý v tejto práci, je modulom strojového učenia. Konkrétne implementácia *support vector machines* triedou `CvSVM`. Táto implementácia zahŕňa všetky známe a používané jadrá, pričom v tejto práci je výber zameraný predovšetkým na *lineárne* a *RBF* jadro. Pre voľbu vhodných parametrov a konštánt SVM a vlastné trénovanie klasifikátoru je použitá metóda `train_auto()`, ktorá automaticky zvolí vhodné parametre na základe *cross-validácie*. Táto trieda tiež implementuje metódu `save()`, ktorá umožňuje uloženie modelu klasifikátoru do súboru na disk. Pre klasifikáciu na základe vytvoreného modelu je možné použiť metódu `predict()`, ktorá na základe vektoru (histogramu) popisujúceho fotografiu určí kategóriu.

5.2 Farebné korelogramy

Výpočet farebného korelogramu nemá podporu v knižnici OpenCV, preto je jeho implementácia súčasťou tejto práce. Implementovaný je konkrétne jeho variant *auto-korelogram*, ktorý sa využíva v praxi kvôli menšej pamäťovej náročnosti. Algoritmus 1 jeho výpočtu vychádza priamo z definície (viď. 2.1.3). Tento algoritmus je implementovaný triedou `ColorCorrelogram`. Priebeh výpočtu je ilustrovaný na obr. 2.4.

Pre výpočet je potrebné zvoliť dva základné parametre a to veľkosť okolia a počet farieb. Veľkosť okolia k predstavuje vzdialenosť medzi analyzovaným bodom obrazu a jeho

Algoritmus 1: *Výpočet farebného auto-korelogramu*

Input: RGB maticová reprezentácia obrazu O, veľkosť okolia k a počet farieb pre kvantovanie obrazu f
Output: Farebný auto-korelogram

```
O := O.kvantovanie(f)
histogram := Array[colors]
denominator := Array[colors]
foreach O as bod do
    foreach bod.okolie(k) as sused do
        if sused.farba == bod.farba then
            histogram[bod.farba]++;
        end
        denominator[bod.farba] += bod.okolie.veľkosť;
    end
end
for i = 1 to f do
    histogram[i] /= denominator[i];
end
return histogram;
```

susedom. Počet farieb je podstatný pre kvantifikáciu obrazu pred samotným výpočtom. Vychádzajúc z práce [14] je zvolená kvantifikácia na 64 farieb. Následne sú vypočítané korelogramy pre viacero okolí – $k \in \{1, 3, 5, 7\}$ – a tieto časti sú spojené do jedného výsledného korelogramu.

Keďže výpočet korelogramu je časovo dosť náročná operácia (je potrebné preskúmať všetkých susedov pre viacero okolí každého bodu obrazu) určite má zmysel hovori o optimalizáciách algoritmu, keďže cieľom kategorizácie nie je len vysoká presnosť ale taktiež časová efektivita. Optimalizácie vychádzajú z faktu, že korelogram vyjadruje pravdepodobnosť, že susediace body obrazu majú (nejakú konkrétnu) rovnakú farbu. Táto pravdepodobnosť by mohla byť aproximovaná pri zachovaní presnosti výpočtu a zvýšení jeho efektivity

1. obmedzením poštu vyšetrovaných bodov – t.j. v obraze budú rovnomerne rozmiestnené body, v ktorých bude prebiehať výpočet korelogramu

2. znížením počtu porovnávaných susedov pri vyšetrovaní každého bodu a pri zachovaní veľkostí a počtu okolí – t.j. porovnávaný bude len *každý druhý* sused okolia (väčšie obmedzenie by vzhľadom na veľkosťou najmenšieho okolia malo negatívny dopad)

Konkrétne hustoty rozmiestnenia bodov v obraze, „vynechávanie“ susedov každého bodu pri porovnávaní farby a dopad týchto optimalizácií na úspešnosť klasifikácie bude predmetom experimentov.

5.3 Popis tried

5.3.1 Teacher

Trieda implementuje všetky prvky *učenia systému*. Obsahuje metódu pre načítanie hierarchie vstupných súborov (viď. 5.4.2). Taktiež implementuje extrakciu lokálnych príznakov,

zhlukovanie príznakových vektorov, výpočet inverznej frekvencie slov v dokumentoch (IDF) a trénovanie klasifikátoru. Využitý je *multiclass* variant SVM klasifikátoru, čo znamená, že klasifikátor sám určí jednu konkrétnu kategóriu z množiny viacerých kategórií (tried).

5.3.2 Codebook

Trieda `Codebook` implementuje *vizuálny slovník a prácu s ním*. Zapúzdruje k-dimenzionálny strom, ktorý je využitý pre vyhľadávanie vizuálnych slov podľa príznakových vektorov. Obsahuje metódu pre výpočet *bag of words*. Taktiež implementuje export a import vizuálneho slovníku z/do súboru na disk.

5.3.3 Detector

Táto trieda zapúzdruje `GridAddaptedFeatureDetector` pre *detekciu význačných bodov* metódou SURF z obrazu. Okrem samotnej detekcie bodov v rovnomernej mriežke (ktorú už implementuje uvedený zapúzdrený objekt) poskytuje informáciu o príslušnosti jednotlivých význačných bodov ku konkrétnej bunke mriežky.

5.3.4 Histogram

Trieda implementuje jednoduchý histogram. Predstavuje vnútornú reprezentáciu *bag of words*. Obsahuje metódy pre spájanie viacerých histogramov, pripojenie korelogramu a rozhranie pre `std::vector` knižnice *STL*, ktorý trieda `Histogram` interne používa pre reprezentáciu histogramu.

5.3.5 ColorCorrelogram

Táto trieda je zdedená od triedy `Histogram`. Okrem reprezentácie *farebného auto-korelogramu* obsahuje metódy pre kvantifikáciu obrazu a následný výpočet tohto korelogramu. Implementuje *delenie obrazu* do rovnomernej mriežky a výpočet korelogramu v jednotlivých bunkách oddelene.

5.4 Rozhranie aplikácie

5.4.1 Uloženie vnútorných dát

Počas fázy učenia systému je vytvorený vizuálny slovník a model klasifikátoru. Tieto dva „produkty“ je potrebné uložiť do súboru (resp. do súborov), aby mohli byť využité vo fáze klasifikácie, keďže ich výpočet je časovo náročná záležitosť.

Trieda `CvSVM` priamo implementuje ukladanie *modelu klasifikátoru* a jeho opätovný import zo samostatného súboru, pričom je použitý formát *XML*. Vďaka tejto možnosti je teda ponechaný export a import klasifikátoru čisto na metódy tejto triedy.

Vizuálny slovník je teda uložený do samostatného súboru. Rovnako ako vytvorenie tak aj jeho export a import je vykonávaný plne v réžii implementácie aplikácie. Prvotnou myšlienkou bolo taktiež využiť formát *XML*, ktorý má silnú vyjadrovaciu schopnosť a bol by pre človeka zrozumiteľnejší. Avšak kvôli jednoduchosti ukladaných dát, nižšej pamäťovej náročnosti a rýchlosti spracovania je použitý formát *CSV*. Vizuálny slovník je uložený nasledujúcim spôsobom:

```
<počet-kategórií>;<názov-kat.-1>;...;<názov-kat-n>;<dĺžka-príznakov>
<IDF-slova-1>;<vektor-slova-1[0]>;...;<vektor-slova-1[d]>
<IDF-slova-2>;<vektor-slova-2[0]>;...;<vektor-slova-2[d]>
...
```

5.4.2 Rozhranie aplikácie

Výsledkom tejto práce sú dve konzolové aplikácie s jednoduchým *CLI rozhraním*, ktoré bolo navrhnuté vzhľadom na potreby testovania a experimentovania rôznych prístupov ku kategorizácii obrazu. Umožňuje jednoduchú aktiváciu/de-aktiváciu jednotlivých vylepšení štandardného postupu.

Za učenie systému je zodpovedná aplikácie `teacher` a vlastnú kategorizáciu fotografií vykonáva aplikácia `tagger`. Použitie aplikácií je nasledovné:

```
./teacher [PREPÍNAČE] <TRÉNOVACÍ PRIEČINOK>
./tagger [PREPÍNAČE] <FOTOGRAFIA-1> ... <FOTOGRAFIA-n>

    PREPÍNAČE:
      -k slov,iterácií    veľkosti slovníku a počet iterácií k-means
      -b cesta            umiestnenie súboru vizuálneho slovníka
      -m cesta            umiestnenie modelu SVM
      -c R,S,vzdialenosť  použitie farebných korelogramov v mriežke RxS
                          s daným rozložením bodov podľa danej vzdialenosti
      -o                  použitie OpponentSURF
      -s počet            použitie mäkkého priradenie k danému počtu slov
      -g R,S              delenie obrazu pre extrakciu lokálnych príznakov
```

Všetky uvedené prepínače (okrem `-k`, ktorý je určený len pre aplikáciu `teacher`) je možne použiť u oboch aplikácií. Pre úspešnosť klasifikácie je potrebné, aby bol systém používaný s rovnakými nastaveniami, s akými bol trénovaný. Všetky prepínače sú nepovinné a pri vynechaní bude program bežať s predvolenými nastaveniami:
`./teacher|tagger -k 1000,4 -b codebook.csv -c svmmodel.xml <fotografie>`

Jediný povinný argument pre učenie systému je *trénovací priečinok*. Ten obsahuje jednoúrovňovú adresárovú štruktúru, pričom názov každého adresára udáva kategóriu a obsahuje práve a len trénovacie fotografie tejto kategórie. Takáto forma vstupu trénovacích dát je volená hlavne kvôli jednoduchosti.

Kapitola 6

Výsledky práce

V tejto kapitole budú zhrnuté všetky použité metódy a ich výsledky. Úspešnosť jednotlivých metód je hodnotená prevažne na základe priemernej presnosti, mediánu presnosti a priemerného času kategorizácie jednej fotografie (do tohto času nie je započítaná doba inicializácie aplikácie – t.j. import slovníku a modelu klasifikátora, keďže tieto činnosti sú vykonané jednorázovo pre celú sadu testovacích fotografií).

Experimenty boli vykonávané na notebooku *HP Pavilion s dvoj-jadrovým procesorom AMD Turion (2,2GHZ)*. Testy prebiehali na nezaťaženom stroji – t.j. všetky prostriedky boli takmer plne dostupné testovacej aplikácii.

Výpočet *presnosti* a *odozvy* klasifikácie fotografií pre jednotlivé kategórie vychádza z rovníc popísaných v 4.1. *Medián presnosti* je určený na základe zoradených dosiahnutých presností pre všetky kategórie. Keďže je v tejto práci použitá kategorizácia každej fotografie do práve jednej kategórie, hodnota *priemernej presnosti* je určená ako podiel správne kategorizovaných fotografií a všetkých fotografií. Doba kategorizácie fotografie je vypočítaná na základe rozdielov hodnoty vrátenej funkciou `clock()` pred a po klasifikácii fotografie.

V prvej časti experimentov je určené základné riešenie, bez prídavných „vylepšení“. Ďalej nasledujú experimenty s pokročilejšími technikami klasifikácie obrazu. V poslednej časti tejto kapitoly budú výsledky tejto práce porovnané s výsledkami súťaže VOC Challenge.

Experimenty boli vyhodnocované využitím automatických testov s využitím skriptov napísaných v *BASH-i* a v jazyku *Perl*. Pre vizualizáciu výsledkov bola použitá aplikácia *Gnuplot*.

6.1 Základné riešenie

6.1.1 Optimalizácia slovníku

Najprv budú optimalizované parametre tvorby vizuálneho slovníka. To znamená počet zhlukov (slov) a počet iterácií k-means. Pri týchto testoch bolo použité RBF jadro pre SVM klasifikátor. Centrá zhlukov sú inicializované na základe metódy k-means++. Výsledky experimentu sú uvedené v tabuľke 6.1.

Na základe tohto experimentu je veľkosť slovníku stanovená na *2000 slov* a metóda k-means bude svoj algoritmus zhlukovania *opakovať 4-krát*. Vyššie hodnoty výsledky klasifikácie nevylepšujú – naopak, väčší slovník zvyšuje časovú náročnosť klasifikácie kvôli dlhšiemu vyhľadávaniu vo väčšom k-dimenzionálnom strome.

Počet zhlukov	Počet iterácií	Priemerná úspešnosť [%]	Rýchlosť klasifikácie [ms]
300	8	35,88	166
600	8	36,99	169
1000	8	36,17	172
2000	2	38,27	182
2000	3	37,86	182
2000	4	39,39	181
2000	6	38,56	181
2000	8	38,73	185
3000	8	38,36	189
5000	8	38,77	204
7000	8	36,99	220
10000	8	36,17	247

Tabuľka 6.1: Výsledky klasifikácie pre rôzne nastavenia k-means. Vybrané riešenie je vyznačené.

6.1.2 Support Vector machines

Ako už v tejto práci bolo uvedené, OpenCV ponúka všetky bežne používané druhy jadier pre SVM. Pri experimentoch bola hodnotená úspešnosť pre *lineárne*, *RBF* a *polynomiálne* jadro. Na základe výsledkov uvedených v tabuľke 6.2 bolo vybrané jadro RBF. Experimenty boli vykonávané na základnom riešení, uvedenom a vyznačenom v tabuľke 6.1.

Jadro	Priemerná úspešnosť [%]	Medián [%]	Celková doba učenia
Lineárne	39,28	36,33	1h 53min
RBF	39,39	36,36	2h 36min
Polynomiálne	39,37	36,34	19h21min

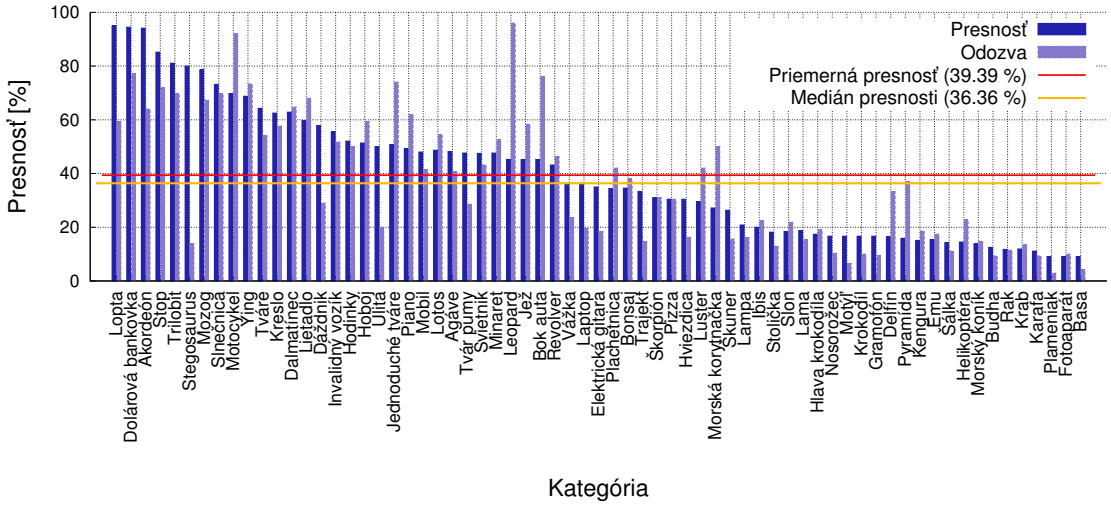
Tabuľka 6.2: Výsledky pri použití rôznych jadier SVM.

Na základe uvedených experimentov bolo určené základne riešenie, ktorého podrobné výsledky sú pre jednotlivé kategórie uvedené v grafe 6.1

6.2 Extrakcia lokálnych príznakov v opponent color space

Pri tejto metóde je pre každý bod extrahovaný príznakový vektor v každom z troch kanálov tohto farebného priestoru (viď. 2.1.2). Tie sú následne spojené, čím vzniká *3-krát dlhší* príznakový vektor. Tím je zároveň spôsobené, že jednotlivé vektory a aj vizuálne slová sú v priestore rozmiestnené „chaotickejšie“.

Tento fakt zrejme spôsobil, že spomínaná metóda celkovo zhoršila výsledky klasifikácie, ako je uvedené v tabuľke 6.3. Výsledky sa nezlepšili ani pri výraznom zvýšení veľkosti slovníku, či pri vyššom počte behov algoritmu k-means.



Obr. 6.1: Výsledky základného riešenia cez všetky použité kategórie.

Iterácie k-means	Veľkosť slovníku	Priemerná presnosť [%]	Medián presnosti [%]
4	2000	27,33	21,62
8	5000	26,92	22,22
16	10000	26,30	23,81

Tabuľka 6.3: Výsledky pri použití opponent color space

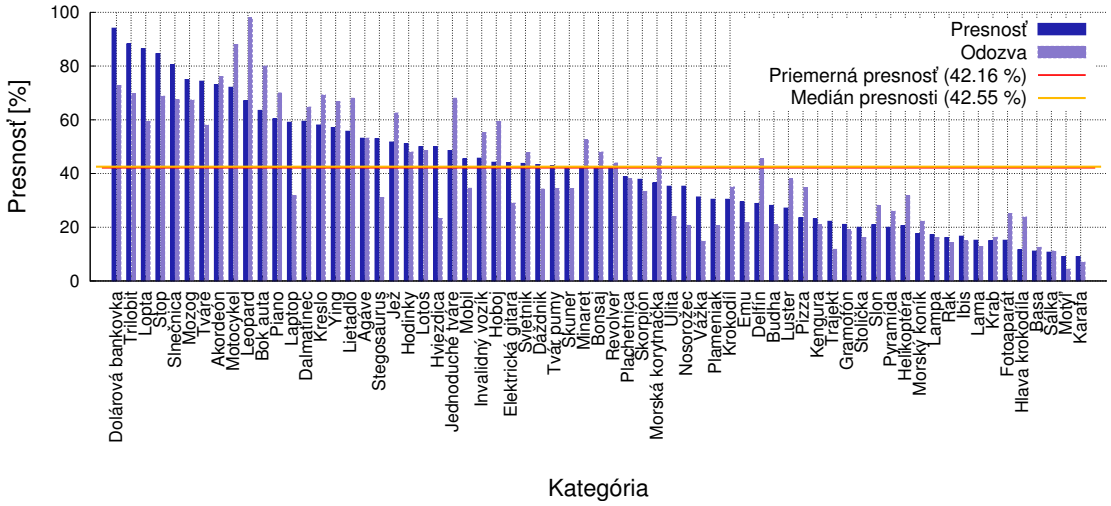
6.3 Mäkké priradenie – soft assigment

V rámci experimentov bolo vyhodnotené mäkké priradenie *viacerých najbližších susedov* – t.j. viacerých vizuálnych slov – ku každému extrahovanému príznakovému slovu. To spôsobilo „vyhladenie“ histogramu bag of words. Výsledky tejto metódy, ktorá mala na efektivitu kategorizácie pozitívny vplyv, sú v tabuľke 6.4.

Počet „susedov“	Priemerná presnosť [%]	Medián [%]	Doba kategorizácie [ms]
2	41,16	41,30	184
3	41,41	43,86	185
4	41,37	43,40	185
6	41,62	43,33	188
8	41,29	41,67	191
10	42,16	42.86	189
16	40.83	40,00	193

Tabuľka 6.4: Výsledky mäkkého priradzovania vizuálnych slov. Vybrané optimálne riešenie je vyznačené.

Podrobné výsledky vyznačeného riešenia z tabuľky 6.4 pre jednotlivé kategórie sú uvedené v grafe 6.2.



Obr. 6.2: Výsledky mäkkého priradenia pri použití 10-tich najbližších „susedov“

6.4 Delenie obrazu pre extrakciu lokálnych príznakov

Použitie tejto metódy spôsobí rozdelenie fotografie podľa rovnomernej mriežky. V *každej bunke je určený bag of words samostatne*. Spojenie týchto histogramov prebieha pred vstupom do klasifikátoru. Tým vzniká niekoľko-násobne dlhší histogram.

Na základe výsledkov uvedených v tabuľke 6.5 nemala táto metóda pozitívny vplyv na výsledky kategorizácie. Je možné sa domnievať, že problémom sú práve príliš dlhé histogramy na vstupe klasifikátoru. Podobnou problematikou sa zaoberal článok v magazíne *Pattern recognition* [14], kde pri výraznom znížení dimenzionality rôznych farebných histogramov boli dosiahnuté rovnaké, či v niektorých prípadoch i lepšie výsledky klasifikácie. Z tohoto dôvodu prebehli experimenty s menšími vizuálnymi slovníkmi, ktoré implikujú kratšie BOWs.

Výsledky (viď. 6.5 a 6.1) ukazujú, že pre *malé slovníky* (napr. 500 slov), dokáže delenie obrazu zlepšiť presnosť klasifikácie. Táto metóda by teda mohla byť vhodná pri požiadavkách na menšiu pamäťovú náročnosť či na rýchlejšie učenie systému. Pre ilustráciu je uvedené porovnanie časov učenia:

Základné riešenie	2000 vizuálnych slov	bez delenia obrazu	učenie trvá 2h36min
Alternatíva	500 vizuálnych slov	delenia obrazu 2x1	učenie trvá 1h7min

6.5 Farebné korelogramy

Použitie farebných korelogramov pridáva informáciu o farebnosti a rozložení farieb v obraze, keďže tradičné príznakové vektory sú extrahované len na základe intenzity v obraze, ktorý je v odtieňoch šedej farby.

Rovnako ako u lokálnych príznakov je i v tomto prípade *možné obraz deliť* na niekoľko častí, v každej časti určiť korelogram oddelene a spojiť ich pred vstupom do klasifikátoru.

Ako je uvedené v tabuľke 6.6, použitie farebných korelogramov značne vylepšilo výsledky klasifikácie, obzvlášť pri delení obrazu na viaceré časti. Ako optimálne riešenie je zvolené *delenie 8x8*, keďže jemnejšie delenie obrazu výsledky výrazne nezlepšilo a naviac je doba

Delenie obrazu		Počet	Priemerná	Medián	Priemerná rýchlosť
riadkov	stĺpcov	vizuálnych slov	presnosť [%]	presnosti [%]	kategorizácie [ms]
2	1	500	38,40	35,90	130
3	1	500	37,16	37,50	106
2	1	1000	38,65	35,48	136
3	1	1000	37,20	35,00	115
2	1	2000	37,78	36,43	150
2	2	2000	33,40	33,33	149
3	1	2000	36,09	38,81	131
3	2	2000	30,51	33.33	152

Tabuľka 6.5: Výsledky kategorizácie pri použití delenia obrazu pred detekciou význačných bodov.

učenia systému a inicializácie pred klasifikáciou (t.j. import slovníku a modelu klasifikátoru) značne zvýšená.

Delenie obrazu		Priemerná	Medián	Približná doba	Doba inicializácie
riadkov	stĺpcov	presnosť [%]	presnosti [%]	učenia [h:min]	pre kategorizáciu [s]
1	1	42,07	38,10	3:28	2,53
3	1	44,01	41,18	4:34	3,01
2	2	44,05	40,91	4:54	3,14
3	2	45,09	42,65	5:54	3,52
4	4	46,33	47,50	8:33	5,42
6	6	48,35	50,00	14:45	8,54
8	8	49,42	52,78	23:08	11,33
10	10	49,55	48,48	33:13	15,79
12	12	49,55	52,27	38:15	23,28

Tabuľka 6.6: Výsledky klasifikácie pri využití farebných korelogramov a pri ich výpočte v jednotlivých častiach obrazu oddelene.

Čo sa týka rýchlosti klasifikácie, výpočet farebného korelogramu je časovo náročná operácia. Pri experimentoch sa priemerná doba klasifikácie pohybovala *od 0,953s do 1,167s* bez ohľadu na spôsob delenia obrazu, čo je približne 5-krát až 6-krát viac, ako doba klasifikácie pri použití základného riešenia.

6.5.1 Optimalizácia rýchlosti

Ako bolo uvedené v predchádzajúcej sekcii, výpočet farebného korelogramu je časovo relatívne náročná operácia. Preto sú v tejto práci použité optimalizácie jeho výpočtu, ktoré sú uvedené v 5.2.

Ide jednak o *zníženie hustoty* (a teda počtu) skúmaných bodov obrazu. Pri priechode po obraze sú skúmané len body v každom n*-tom* ($n = hustota$) riadku a stĺpci. Efekt tejto optimalizácie je vlastne rovnaký, ako keby bol vstupný obraz pred výpočtom korelogramu zmenšený. Hodnota uvedenej hustoty analyzovaných bodov je teda závislá na veľkosti klasifikovaných fotografií. V tejto práci sa pri testovaní používali fotografie v rozlíšení približne 300 x 200 bodov.

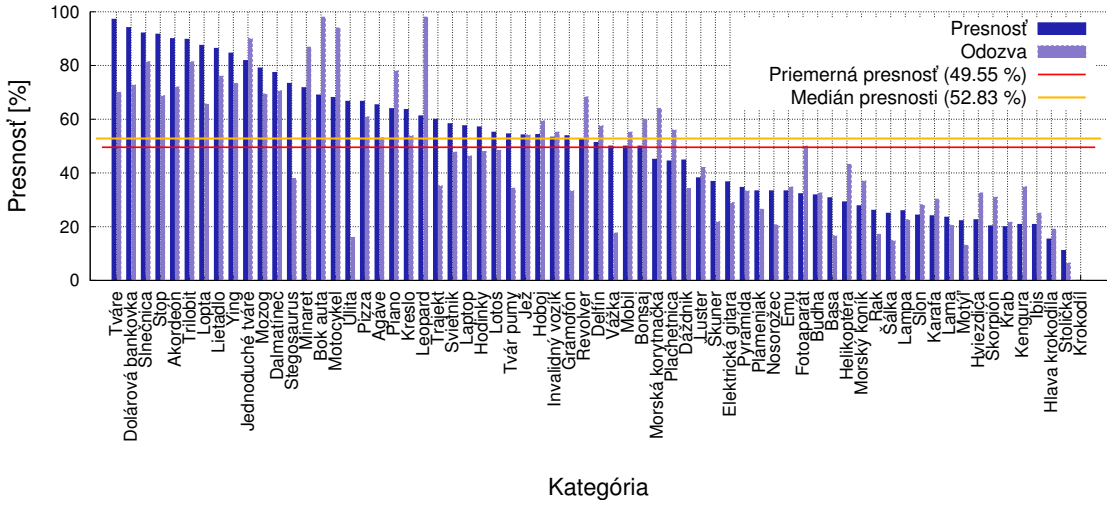
Ďalej je redukovaný počet „susedov“ v okolí každého analyzovaného bodu. Pri porovnávaní farby je *vynechaný každý druhý bod okolia*. Vzhľadom na to, že sú využité relatívne malé okolia, väčšie redukcie počtu porovnávaných „susedov“ neboli skúmané.

Výsledky uvedených optimalizácií sú uvedené v tabuľke 6.7. Je zjavné, že optimalizácie mali značne pozitívny vplyv na rýchlosť klasifikácie pri približnom zachovaní pôvodnej presnosti. Na základe uvedených výsledkov je taktiež možné konštatovať, že klasifikácia dosahuje vyššiu presnosť pri použití prvej z uvedených optimalizácií, ktorá de facto samotný algoritmus výpočtu nemení, ale ako bolo uvedené, len „zmenší“ obraz pred výpočtom korelogramu.

Hustota bodov	Redukcia okolia	Zníženie výpočtov [%]	Priemerná presnosť [%]	Medián presnosti [%]	Priemerná doba kategorizácie [ms]
1	nie	0,00	49,42	52,78	0,955
1	áno	50,00	48,51	50,00	0,641
2	nie	25,00	49,38	51,85	0,441
2	áno	12,50	48,64	50,00	0,376
3	nie	11,11	49,55	52,83	0,364
3	áno	5,56	48,55	50,00	0,325
4	nie	6,25	49,17	51,85	0,328
4	áno	3,13	48,55	50,00	0,307
5	nie	4,00	49,13	50,00	0,312
5	áno	2,00	48,51	50,00	0,298

Tabuľka 6.7: Výsledky klasifikácie pri optimalizácii výpočtu farebných korelogramov.

Podrobné výsledky klasifikácie pri použití farebných korelogramov, vypočítaných v mriežke 8 x 8 a obmedzení počtu skúmaných bodov obrazu na body v každom treťom riadku a a stĺpci sú uvedené v grafe 6.3.



Obr. 6.3: Výsledky klasifikácie cez všetky kategórie pri použití farebných korelogramov pri optimálnych nastaveniach.

6.6 Výsledné riešenie

na základe predchádzajúcich experimentov preukázali pozitívny vplyv na kategorizáciu obrazu techniky *mäkké priradenie* a *farebné korelogramy* – predovšetkým s využitím *delenia obrazu* pred ich samotným výpočtom.

Spojením týchto techník a nastavením príslušných parametrov podľa predchádzajúcich experimentov vzniklo riešenie, pri ktorom je základné riešenie (viď. 6.1) obohatené o mäkké priradenie 10 najbližších susedov (pri vyhľadávaní vo vizuálnom slovníku) a výpočet farebného korelogramu s delením obrazu podľa mriežky 8 x 8.

V tabuľke 6.8 sú uvedené výsledky tohto riešenia. Ako je zrejmé z uvedených experimentov, spojenie mäkkého priradenia a farebných korelogramov má v konečnom dôsledku na výsledky kategorizácie negatívny vplyv. Výsledné riešenie porovnávané s výsledkami VOC 2007 sa teda obmedzí na použitie farebných korelogramov bez mäkkého priradenie.

Mäkké priradenie sa teda osvedčilo ako technika, ktorá vylepšuje výsledky klasifikácie pri zachovaní jednoduchosti riešenia a rýchlosti klasifikácie. Nedosahuje sa pri ňom však takých výsledkov ako pri použití farebných korelogramov.

Farebné korelogramy		Mäkké priradenie	Priemerná	Medián
riadkov	stĺpcov	[počet susedov]	presnosť [%]	presnosti [%]
8	8	1	49,55	52,83
8	8	3	47,94	50,00
8	8	5	46,33	44,83
8	8	10	44,76	43,75

Tabuľka 6.8: Výsledky kategorizácie s využitím farebných korelogramov (počítaných v delenom obraze) a mäkkého priradenia vizuálnych slov k príznakovému vektoru.

6.6.1 Porovnanie s výsledkami VOC Challenge

Riešenie určené v predchádzajúcej sekcii a vyznačené v tab. 6.8 je aplikované na jednotlivé testovacie sady uvedené v 4.2.1.

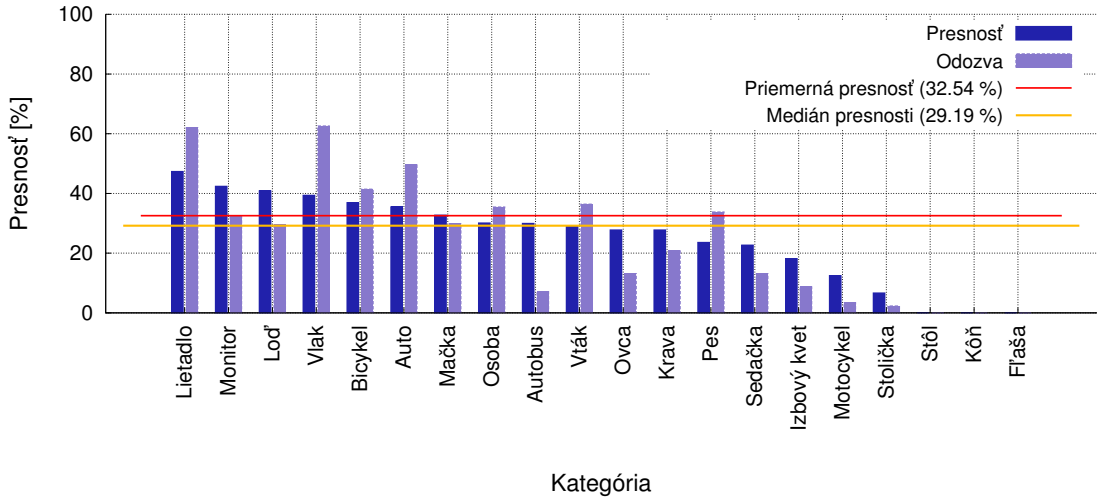
Na základe výsledkov v tab. 6.9 a výsledkov súťaže VOC 2007 (viď. 3.3) je možné konštatovať, že výsledky tejto práce sa blížia výsledkom, ktoré dosahovali špičkové tímy v tejto súťaži. Najpresnejšie porovnanie ponúkajú výsledky pri použití sady 2, ktorá je priamo čerpaná z testovacích dát tejto súťaže. Je však potrebné mať na pamäti, že z tejto sady boli odobrané fotografie (asi 45%), ktoré sú anotované viacerými kategóriami, keďže v tejto práci je použitá kategorizácia vždy do práve jednej kategórie. Tento fakt pochopiteľne úlohu kategorizácie zjednodušil.

Výsledky pri použití sady 3 sú tu uvedené hlavne pre demonštráciu faktu, že jednoduchosť scény a nízka variabilita zobrazení objektov na fotografiách databáze Caltech101 značne kategorizáciu uľahčujú. Výsledky oproti sade 2, ktorá obsahuje reálnejšie fotografie sú výrazne lepšie.

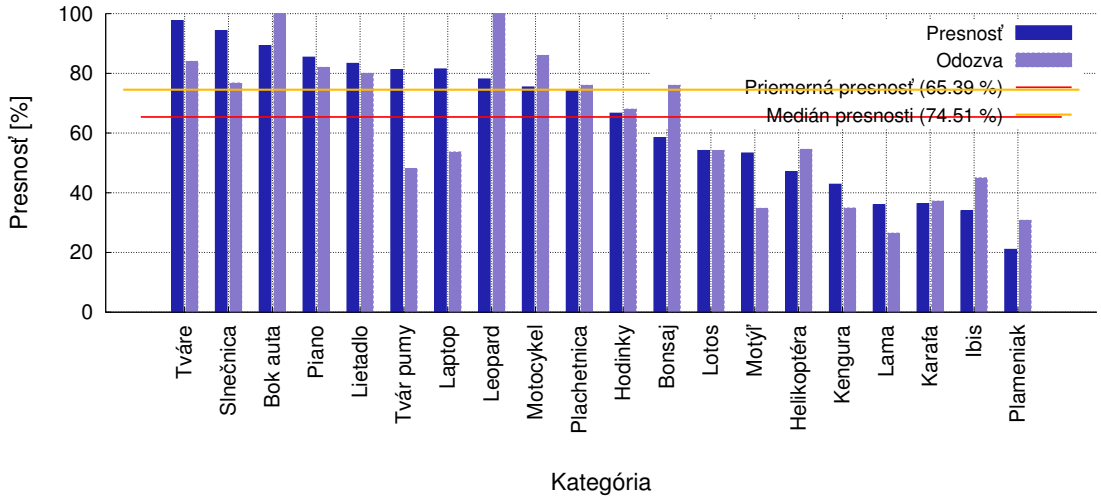
Porovnanie s výsledkami súťaže VOC 2007 na základe mediánu presnosti cez všetky kategórie testovacej sady sa nachádza na obr.6.6. Medián je zvolený z toho dôvodu, že jeho hodnoty pre riešenia jednotlivých tímov sú zverejnené v článku, ktorý popisuje výsledky z roku 2007 [9].

Testovacia sada	Zdroj fotografií	Počet kategórií	Priemerná presnosť [%]	Medián presnosti [%]	Podrobné výsledky
1	Caltech101	67	49,55	52,83	graf 6.3
2	VOC 2007	20	32,54	29,19	graf 6.4
3	Caltech101	20	65,39	74,51	graf 6.5

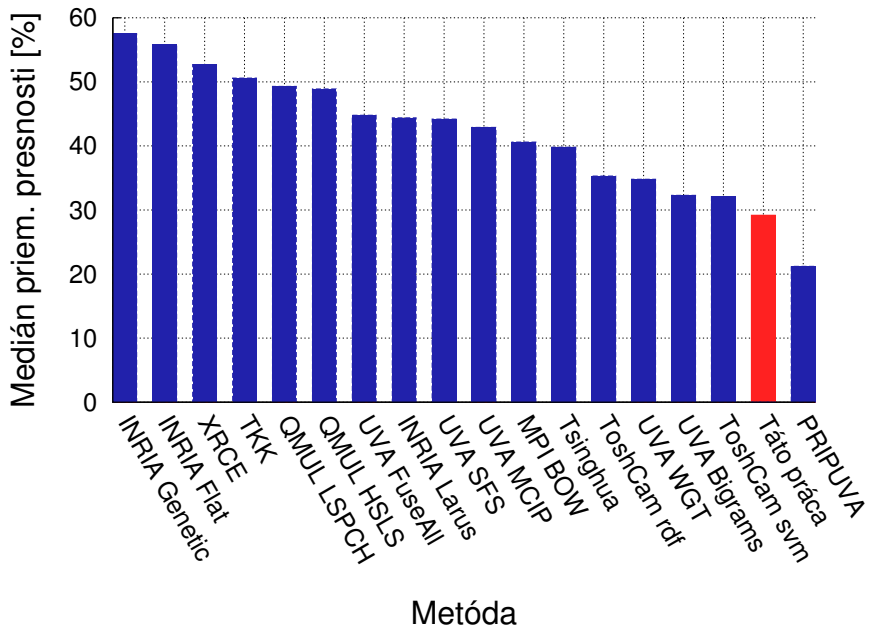
Tabuľka 6.9: Výsledky klasifikácie pre finálne riešenie a rôzne testovacie sady.



Obr. 6.4: Podrobné výsledky klasifikácie finálneho riešenia pri použití testovacej sady 2 (podmnožina testovacej sady dát súťaže VOC 2007).



Obr. 6.5: Podrobné výsledky klasifikácie finálneho riešenia pri použití testovacej sady 3 (podmnožina sady Caltech101 – 20 kategórií).



Obr. 6.6: Porovnanie výsledkov klasifikácie finálneho riešenia pri použití testovacej sady 2 (podmnožina testovacej sady dát súťaže VOC 2007) s výsledkami súťaže VOC 2007. Porovnávanou hodnotou je medián presnosti cez všetky kategórie.

Kapitola 7

Záver

Hlavným zámerom tejto práce bolo vytvorenie systému pre automatickú kategorizáciu fotografií, ktorá by sa opierala len o obrazový obsah. Vytýčenými cieľmi boli hlavne spoľahlivosť kategorizácie ale taktiež efektivita behu. Sekundárnym zámerom bolo naštudovať, použiť a na základe experimentov porovnať alternatívne či menej tradičné metódy a techniky pre kategorizáciu obrazu.

V rámci riešenia tejto práce boli vytvorené dve kooperujúce aplikácie. Prvá z nich na základe trénovacích anotovaných dát vytvorí vizuálny slovník a natrénuje klasifikátor. Následne druhá z dvojice aplikácií použije tieto výstupy pre vlastnú kategorizáciu fotografií neznámych kategórií.

Popri tradičných technikách ako použitie *lokálnych príznakov* (v tejto práci SURF), vyhľadávanie extrahovaných príznakových vektorov vo *vizuálnom slovníku*, vytváranie histogramu početnosti výskytu vizuálnych slov – *bag-of-words* – a na jeho základe klasifikácia s využitím *support vector machines*, boli v tejto práci použité aj mierne pokročilejšie techniky. Ide konkrétne o použitie mäkkého priradenia – *soft assignment* či delenia obrazu na jednotlivé časti. V týchto častiach boli nezávisle vytvárané histogramy bag-of-words, či vypočítané *farebné korelogramy*, ktoré dopĺňali popis obrazu na základe lokálnych príznakov. Ďalej prebehli experimenty s extrakciou lokálnych príznakov v alternatívnom farebnom priestore *opponent color space*.

Uvedené „vylepšenia“ tradičného postupu vo všeobecnosti vylepšili výsledky kategorizácie. Konkrétne hodnoty presnosti sú silne závislé na počte kategórií a na rozmanitosti umiestnení objektov v scéne. Pri kategorizovaní fotografií databázy Caltech101 a použití 67 kategórii je dosiahnutá priemerná presnosť 49,55%. Čo sa týka efektivity behu aplikácie, ktorá vykonáva vlastnú kategorizáciu, tak doba spracovania fotografie na bežnom osobnom počítači sa pohybovala do 200ms. Pri použití farebných korelogramov sa táto doba výrazne zvýšil, ale s využitím jednoduchých optimalizácií sa doba kategorizácie pohybovala okolo 300ms.

Na základe uvedených faktov je možne konštatovať, že zámery a ciele práce boli splnené. Budúca práca v tejto oblasti by mohla zahŕňať použitie efektívnejších a robustnejších náhodných štruktúr pre reprezentáciu vizuálne slovníka *extremly randomized trees*[4], či detekciu tzv. oblasti záujmu *(region of interest)*. To znamená hľadať v obraze oblasti „kde niečo je“. Prípadne by mohol byť použitý tzv. *dense sampling* – extrahovanie lokálnych príznakov v bodoch, ktoré sú rozmiestnené po fotografii v pravidelnej mriežke. Rozšírenie práce by bolo možné aj v oblasti delenia obrazu, keď by bola fotografia delená naraz podľa viacerých mriežok. Budúca práca by sa mohla taktiež odraziť v zmene vlastnej klasifikácie, ktorá by podporovala zatriedenie fotografie do viacerých kategórií.

Nesporne by bolo do budúcna vhodné vhodne vyriešiť rozhranie aplikácie podľa konkrétnej domény, kde bude využívaná (grafické rozhranie v prípade bežných užívateľov alebo sofistikovanejšie rozhranie na príkazovom riadku pri využití napr. vo webových fotoalbumoch).

Literatúra

[1] *OpenCV reference manual, v2.2*. December 2010, [PDF].

[2] Bay, H.; Ess, A.; Tuytelaars, T.; aj.: Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding (CVIU)*, ročník 110, 2008: s. 346–359.
URL
ftp://ftp.vision.ee.ethz.ch/publications/articles/eth_biwi_00517.pdf

[3] Bentley, J. L.: Multidimensional binary search trees used for associative searching. *Communications of the ACM*, ročník 18, č. 9, 1975: s. 509–517.
URL http://portal.acm.org/citation.cfm?doid=361002.361007

[4] Bosch, A.; Zisserman, A.; Munoz, X.: Image Classification using Random Forests and Ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Říjen 2007, s. 1–8.
URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4409066

[5] Burget, L.: Klasifikace a rozpoznávání – úvod do problematiky. 2011, [PDF].

[6] Chandran, S.: Introduction to kd-trees. 2002, [PDF].
URL http://www.cs.umd.edu/class/spring2002/cmsc420-0401/pbasic.pdf

[7] Csurka, G.; Dance, C. R.; Fan, L.; aj.: Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, s. 1–22.

[8] Everingham, M.; Van Gool, L.; Williams, C. K. I.; aj.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.
http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[9] Everingham, M.; Van Gool, L.; Williams, C. K. I.; aj.: The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, ročník 88, č. 2, Červen 2010: s. 303–338, [online].
URL
http://pascallin.ecs.soton.ac.uk/challenges/VOC/pubs/everingham10.pdf

[10] Fei-Fei, L.; Fergus, R.; Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *Workshop on Generative-Model Based Vision*, IEEE, CVPR 2004, 2004, [online].
URL http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[11] Hsu, C.-W.; Chang, C.-C.; Lin, C.-J.: A Practical Guide to Support Vector Classification. 2010.
URL http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[12] Kuang, Y.; Åström, K.; Kopp, L.; aj.: Optimizing Visual Vocabularies Using Soft Assignment Entropies. In *Asian Conference on Computer Vision*, 2010.

[13] Philbin, J.; Chum, O.; Isard, M.; aj.: Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
URL http://www.robots.ox.ac.uk/~vgg/publications/papers/philbin07.pdf

[14] Qiu, G.; Feng, X.; Fang, J.: Compressing Histogram Representations For Automatic Colour Photo Categorization. 2004.
URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.4961

[15] Rosa, Š.: Automatic photo tagging. *Sborník prací konference a soutěže Student EEICT 2010*, 2010.
URL http://www.feec.vutbr.cz/EEICT/2010/sbornik/02-Magisterske_projekty/02-Zpracovani_signalu_obrazu_a_dat/13-xrosas00.pdf

[16] van de Sande, K.; Tahir, M. A.; aj.: University of Amsterdam & Surrey @ PASCAL VOC 2008. 2008.
URL http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/workshop/tahir.pdf

[17] van de Sande, K. E. A.; Gevers, T.; Snoek, C. G. M.: Evaluating Color Descriptors for Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 32, č. 9, 2010: s. 1582–1596.
URL http://www.science.uva.nl/research/publications/2010/vandeSandeTPAMI2010

[18] Smeulders, A.; Worring, M.; Santini, S.; aj.: Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ročník 22, č. 12, Prosinec 2000: s. 1349 –1380, ISSN 0162-8828, doi:10.1109/34.895972.

[19] Wikipedia: tf-idf. 2011, [Online; citivané 20.3.3011].
URL http://en.wikipedia.org/wiki/Tf%E2%80%93idf

[20] Zbořil, F.: *Základy umělé inteligence – studijní opora*. FIT VUT v Brně, 2006, [PDF].

Dodatok A

Obsah CD

Priložené CD obsahuje:

- zdrojové kódy tejto práce
- zdrojové kódy použitej knižnice OpenCV umožňujúce preklad
- binárnu spustiteľnú verziu výsledných aplikácií spolu so zdieľanými knižnicami
- skripty demonštrujúce použitie a činnosť
- použité trénovacie a testovacie sady fotografií
- príklady vizuálnych slovníkov a modelov klasifikátoru z prebehnutých trénovaní
- *readme* súbor s popisom použitia
- PDF a LaTeX verziu tejto práce
- plagát prezentujúci použité metódy, postupy a výsledky