

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VYHLEDÁVÁNÍ OBRAZU NA ZÁKLADĚ PODOBNOSTI

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

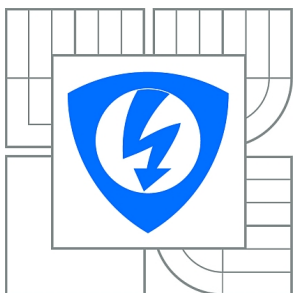
AUTOR PRÁCE
AUTHOR

Bc. MARTIN HARVÁNEK

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VYHLEDÁVÁNÍ OBRAZU NA ZÁKLAD PODOBNOSTI

IMAGE SEARCH USING SIMILARITY MEASURES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN HARVÁNEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. RADIM BURGET, Ph.D.

BRNO 2014



VYSOKÉ U ENÍ
TECHNICKÉ V BRN

Fakulta elektrotechniky
a komunika ních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunika ní a informa ní technika

Student: Bc. Martin Harvánek

ID: 72569

Ro ník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Vyhledávání obrazu na základ podobnosti

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodami m ení podobnosti obraz . Navrhn te a implementujte alespo t i metody, s pomocí kterých lze m ít podobnost obraz . Demonstrujte dosažené výsledky a na vhodném p íklad a výsledky diskutujte.

DOPORU ENÁ LITERATURA:

[1] Chechik, Gal, et al. "Large scale online learning of image similarity through ranking." The Journal of Machine Learning Research 11 (2010): 1109-1135.

[2] Batko, Michal, et al. "Building a web-scale image similarity search system." Multimedia Tools and Applications 47.3 (2010): 599-629.

Termín zadání: 10.2.2014

Termín odevzdání: 30.5.2014

Vedoucí práce: Ing. Radim Burget, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Ji í Mišurec, CSc.

P edseda oborové rady

UPOZORN NÍ:

Autor diplomové práce nesmí p i vytvá ení diplomové práce porušit autorská práva t etích osob, zejména nesmí zasahovat nedovoleným zp sobem do cizích autorských práv osobnostních a musí si být pln v dom následk porušení ustanovení § 11 a následujících autorského zákona . 121/2000 Sb., v etn možných trestn právních d sledk vyplývajících z ustanovení ásti druhé, hlavy VI. díl 4 Trestního zákoníku .40/2009 Sb.

ABSTRAKT

V práci sú implementované metódy: kruhových sektorov, momentov farieb, vektoru spojitých farieb a Gáborových filtrov, ktoré sú založené na obrazových charakteristikách nízkej úrovne. Tieto metódy boli vyhodnotené po nájdení optimálnych parametrov. Hľadanie optimálnych parametrov metód je realizované pomocou presnosti klasifikácie učiacich sa algoritmov a operátora krížová validácia v programe RapidMiner. Implementované metódy sú hodnotené na základe celkovej priemernej precíznosti nad množinou obrazov s desiatimi kategóriami - starodávne budovy, pláž, autobus, dinosaurus, slon, kvet, jedlo, kôň, hora, domorodci. Implementovanou modifikáciou (farebný priestor HSB + štatistická funkcia median) metódy kruhových sektorov je dosiahnutá o 8% vyššia presnosť klasifikácie ako pôvodná metóda uvedená v literatúre. Kombináciou metód momentov farieb, kruhových sektorov a Gáborových filtrov s pridelenými váhovými koeficientami, bol dosiahnutý najlepší výsledok celkovej priemernej precíznosti na úrovni 70,48% zo všetkých implementovaných metód.

KLÍČOVÁ SLOVA

Metrika podobnosti, charakteristiky obrazu, vektor spojitých farieb, kruhové sektory, momenty farieb, Gáborove filtre, presnosť klasifikácie, krížová validácia

ABSTRACT

There are these methods implemented: circular sectors, color moments, color coherence vector and Gabor filters, they are based on low-level image features. These methods were evaluated after their optimal parameters were found. The finding of optimal parameters of methods is done by measuring of classification accuracy of learning operators and usage of operator cross validation on images in program RapidMiner. Implemented methods are evaluated on these image categories - ancient, beach, bus, dinosaur, elephant, flower, food, horse, mountain and natives, based on total average precision. The classification accuracy result is increased by 8% by implemented modification (HSB color space + statistical function median) of original method circular sectors. The combination of methods color moments, circular sectors and Gabor filters with weighted ratio gives the best total average precision at 70,48% and is the best method among all implemented methods.

KEYWORDS

Similarity metrics, image features, color coherence vector, circular sectors, color moments, Gabor filters, classification accuracy, cross validation

HARVÁNEK, Martin *Vyhledávání obrazu na základě podobnosti*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 63 s. Vedoucí práce byl Ing. Radim Burget, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Vyhledávání obrazu na základě podobnosti“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Poděkování

Ďakujem vedúcemu diplomovej práce Ing. Radimovi Burgetovi, Ph.D. za veľmi užitočnú metodickú pomoc a cenné rady pri spracovaní práce.

OBSAH

| | |
|---------------------------------------------------|-----------|
| Úvod | 12 |
| 1 Podobnosť v obraze | 13 |
| 1.1 CBIR systémy | 13 |
| 1.2 Metódy využívajúce charakteristiky farieb | 15 |
| 1.2.1 Prienik histogramov | 15 |
| 1.2.2 Vektor spojitých farieb | 16 |
| 1.2.3 Metóda kruhových sektorov | 18 |
| 1.2.4 Metóda momentov farieb | 18 |
| 1.3 Metódy založené na charakteristikách textúry | 20 |
| 1.3.1 Gáborove filtre | 21 |
| 1.4 Funkcie vzdialenosti | 22 |
| 2 Riešenie | 25 |
| 2.1 Úvod | 25 |
| 2.2 Množina obrazov | 25 |
| 2.3 Vektor charakteristík | 26 |
| 2.4 Hľadanie optimálnych parametrov metód | 27 |
| 2.5 Implementácia metódy vektora spojitých farieb | 28 |
| 2.5.1 Navrhnutý algoritmus | 28 |
| 2.5.2 Určenie optimálnych parametrov | 30 |
| 2.6 Implementácia metódy kruhových sektorov | 31 |
| 2.6.1 Navrhnutý algoritmus | 32 |
| 2.6.2 Modifikácie metódy | 35 |
| 2.6.3 Určenie optimálnych parametrov | 36 |
| 2.7 Implementácia metódy momentov farieb | 37 |
| 2.7.1 Navrhnutý algoritmus | 37 |
| 2.7.2 Modifikácie metódy | 38 |
| 2.7.3 Určenie optimálnych parametrov | 39 |
| 2.8 Implementácia metódy Gáborových filtrov | 40 |
| 2.8.1 Navrhnutý algoritmus | 40 |
| 2.8.2 Určenie optimálnych parametrov | 42 |
| 3 Výsledky práce | 44 |
| 3.1 Spôsob merania metód | 44 |
| 3.2 Vyhodnotenie metódy momentov farieb | 45 |
| 3.3 Vyhodnotenie metódy kruhových sektorov | 46 |

| | | |
|----------|--------------------------------------------------|-----------|
| 3.4 | Vyhodnotenie metódy Gáborových filtrov | 46 |
| 3.5 | Vyhodnotenie kombinácie metód | 47 |
| 3.6 | Vyhodnotenie kategórii obrazov | 48 |
| 3.7 | Výpočtová náročnosť | 48 |
| 3.7.1 | Extrahovanie charakteristík | 48 |
| 3.7.2 | Vyhľadávanie podobných obrazov | 49 |
| 3.8 | Ukážky vyhľadávania podobných obrazov | 49 |
| 3.9 | Zhodnotenie | 51 |
| 4 | Záver | 57 |
| | Literatúra | 58 |
| | Zoznam symbolov, veličín a skratiek | 61 |
| | Zoznam príloh | 62 |
| A | Obsah CD | 63 |

ZOZNAM OBRÁZKOV

| | | |
|------|---------------------------------------------------------------------------|----|
| 1.1 | Architektúra CBIR systému | 14 |
| 1.2 | Metóda kruhových sektorov [16] | 18 |
| 1.3 | Metóda kruhových sektorov - vplyv radenia [16] | 19 |
| 1.4 | Momenty farieb - delenie obrazu na regióny | 20 |
| 2.1 | Príklad vzorových obrazov z rôznych kategórii | 26 |
| 2.2 | RapidMiner: schéma operátora X validácia | 27 |
| 2.3 | CCV - 4-okolie, 8-okolie pixelu | 28 |
| 2.4 | CCV - príklad regiónov a farebných buniek | 30 |
| 2.5 | Vzorový obraz | 30 |
| 2.6 | Kruhové sektory - plocha kruhov | 32 |
| 2.7 | Kruhové sektory - pohyb po sektoroch | 33 |
| 2.8 | Kruhové sektory - maska a sektory prvého kruhu | 34 |
| 2.9 | Gáborove filtre - konvolúcia 8 filtrov | 42 |
| 3.1 | Momenty farieb - celková priemerná precíznosť | 45 |
| 3.2 | Kruhové sektory - celková priemerná precíznosť | 47 |
| 3.3 | Gáborove filtre - celková priemerná precíznosť | 49 |
| 3.4 | Kombinácia metód - celková priemerná precíznosť | 50 |
| 3.5 | Vzorový obraz pre vyhľadávanie - kôň | 53 |
| 3.6 | Kombinácia metód - nájdené obrazy k vzoru kôň | 53 |
| 3.7 | Vzorový obraz pre vyhľadávanie - kôň otočený o 90° | 53 |
| 3.8 | Kombinácia metód - nájdené obrazy k o 90° otoč. vzoru kôň | 54 |
| 3.9 | Kruhové sektory - nájdené obrazy k o 90° otoč. vzoru kôň | 54 |
| 3.10 | Vzorový obraz pre vyhľadávanie - domorodci | 55 |
| 3.11 | Kombinácia metód - nájdené obrazy k vzoru domorodci | 55 |
| 3.12 | Vzorový obraz pre vyhľadávanie - domorodci otočený o 90° | 55 |
| 3.13 | Kombinácia metód - nájdené obrazy k o 90° otoč. vzoru domorodci | 56 |
| 3.14 | Kruhové sektory - nájdené obrazy k o 90° otoč. vzoru domorodci | 56 |

ZOZNAM TABULIEK

| | | |
|------|-----------------------------------------------------------------------|----|
| 1.1 | Mapovanie komponenty na bunky farby | 17 |
| 1.2 | Počet spojitých a nespojitých pixelov | 17 |
| 2.1 | CCV - vplyv parametrov na presnosť klasifikácie | 31 |
| 2.2 | CCV - optimálne parametre | 31 |
| 2.3 | Kruhové sektory - vplyv parametrov na presnosť klasifikácie | 37 |
| 2.4 | Kruhové sektory - optimálne parametre | 37 |
| 2.5 | Momenty farieb - vplyv parametrov na presnosť klasifikácie | 39 |
| 2.6 | Momenty farieb - optimálne parametre | 40 |
| 2.7 | Gáborove filtre - vplyv parametrov na presnosť klasifikácie | 42 |
| 2.8 | Gáborove filtre - optimálne parametre | 43 |
| 3.1 | Momenty farieb - celková priemerná presnosť | 45 |
| 3.2 | Momenty farieb - presnosť kat. - Euklid | 46 |
| 3.3 | Momenty farieb - presnosť kat. - Manhattan | 46 |
| 3.4 | Kruhové sektory - celková priemerná presnosť | 46 |
| 3.5 | Kruhové sektory - presnosť kat. - Euklid | 47 |
| 3.6 | Kruhové sektory - presnosť kat. - Manhattan | 48 |
| 3.7 | Gáborove filtre - celková priemerná presnosť | 48 |
| 3.8 | Gáborove filtre - presnosť kat. - Euklid | 49 |
| 3.9 | Gáborove filtre - presnosť kat. - Manhattan | 50 |
| 3.10 | Kombinácia metód - celková priemerná presnosť | 50 |
| 3.11 | Kombinácia metód - presnosť kat. - Euklid | 51 |
| 3.12 | Kombinácia metód - presnosť kat. - Manhattan | 51 |
| 3.13 | Priemerná presnosť metód v kategóriach | 52 |
| 3.14 | Výpočtová náročnosť - extrakcia charakteristík | 52 |

ÚVOD

Vyhľadávanie obrazu na základe podobnosti má široké uplatnenie v rôznych oblastiach. Pri posudzovaní podobnosti obrazov u človeka sa jedná o subjektívny postoj, pretože každý človek vníma okolitý svet inak. V prípade posudzovania podobnosti obrazov strojom ide o postoj objektívny, pretože pri opakovanom vyhodnocovaní podobnosti medzi dvoma obrazmi dostaneme vždy rovnaký výsledok. Tento postoj však nemusí byť vždy správny a pre určenie podobnosti medzi obrazmi môže byť požadovaný vstup od človeka, ktorý stroj využije pri ďalšom vyhľadávaní podobných obrazov. Stroj posudzuje podobnosť obrazov na základe rozdielu obrazových charakteristík, medzi ktoré patria: farba, textúra a tvary.

V práci sú implementované metódy využívajúce farebné charakteristiky obrazu - vektor spojitých farieb, momenty farieb, kruhové sektory a textúrové charakteristiky obrazu - Gáborove filtre. Pre určenie optimálnych parametrov metód sa využíva presnosť klasifikácie učiacich sa algoritmov v programe RapidMiner nad vektormi charakteristík množiny obrazov. Za minimálnu hranicu presnosti klasifikácie je stanovená hodnota 60%. Zvolenú hranicu nedosahuje metóda vektoru spojitých farieb. Po nájdení optimálnych parametrov metód sú všetky metódy vyhodnocované na základe celkovej priemernej presnosti pri nájdení zvoleného počtu najpodobnejších obrazov. Miera podobnosti obrazu je posudzovaná pomocou metrík Euklid a Manhattan.

Náhradou farebného priestoru RGB priestorom HSB a štatistickej funkcie priemer funkciou medián, je u metódy kruhových sektorov dosiahnutá o 8% vyššia presnosť klasifikácie (75,6%) v programe RapidMiner v porovnaní s pôvodnou metódou (67,6%) z literatúry [16]. Pri desiatich najpodobnejších obrazoch dosahuje najlepšiu celkovú priemernú presnosť metóda momentov farieb - 68,7% pri metrikách Euklid a Manhattan zároveň. Nasleduje metóda kruhov s 67,23% pri metrike Manhattan a metóda Gáborových filtrov s 47,77% pri metrike Euklid. Kombináciou metód momentov farieb, kruhových sektorov a Gáborových filtrov s váhovým rozložením vplyvu jednotlivých metód na metriku podobnosti, je dosiahnutá o 1,78% vyššia celková priemerná presnosť - 70,48% pri metrike Manhattan ako najlepší výsledok samostatnej metódy momentov farieb.

Zbytok práce je organizovaný nasledovne. V kapitole 1 sú pojednávané metódy založené na farebných a textúrových charakteristikách. Jedná sa o metódy: prienik histogramov, vektor spojitých farieb, kruhové sektory, momenty farieb a Gáborove filtre. V kapitole 2 sa preberá implementácia a určenie optimálnych parametrov metód: vektor spojitých farieb, kruhové sektory, momenty farieb a Gáborove filtre. V kapitole 3 sa popisuje spôsob vyhodnocovania metód, sú tu zhrnuté dosiahnuté výsledky a praktické ukážky hľadania podobných obrazov k vzoru.

1 PODOBNOSŤ V OBRAZE

Vyhľadávanie obrazu na základe podobnosti obsahu má široké uplatnenie v rôznych oblastiach ako napríklad medicína, meteorológia, polícia, strojársky, automobilový a elektrotechnický priemysel atď. [18]. Konkrétne príklady aplikácie využitia vyhľadávania obrazu na základe podobnosti sú [5]:

- vyhľadávanie obrazu z množiny obrazov na základe vzoru
- detekcia autorsky chránených obrazov
- mapovanie satelitných dát na mapu
- biometria - detekcia tváre, rozpoznanie tváre, rozpoznanie rukopisu, rozpoznanie odtlačku prstov

V mojej práci sa budem venovať aplikácii - vyhľadávanie obrazu na základe obsahu. Systémy, ktoré pracujú v tejto aplikácii sa nazývajú CBIR (Content-Based Image Retrieval). Podmožinou CBIR systémov sú systémy RBIR (Region-Based Image Retrieval), ktoré využívajú segmentáciu obrazu na regióny a tie následne pomocou rôznych metrík porovnávajú s regiónmi v ostatných obrazoch.

1.1 CBIR systémy

Procesory sú v dnešnej dobe oveľa výkonnejšie, operačné pamäte lacnejšie a zároveň s vyššou kapacitou ako v minulosti. Práve z tohto dôvodu vzrástol záujem o databázy, ktoré uchovávajú digitálne obrazy. Najväčším problémom je vyhľadávanie obrazu na základe obsahu nad týmito databázami. Pre to, aby sme mohli obrazy vyhľadať, je nevyhnutné obrazy označiť - indexovať. Jednou z možností je do obrazu zaniest informáciu v textovej podobe, ktorá slovne popisuje obsah obrazu. Tento prístup je uplatnený v systémoch TBIR (Text-Based Image Retrieval). Samotný index obrazu v tomto prípade tvorí textový reťazec (napr. auto, dom, pes atď.), ktorý je spojený s obrazom. Systém tak na základe textového dotazu vráti obrazy, ktoré sú indexované dotazovaným textom. Príkladom TBIR je images.google.com alebo www.flickr.com.

Indexovať každý obraz textom v databázach s rozsiahlym počtom obrazov, je neefektívne a časovo náročné. Navyše každý obraz môže byť posudzovaný inak podľa toho, ktorá osoba tento popis vykonáva. Jedná sa teda o príliš subjektívny prístup v prípade, že by sme chceli použiť metódu TBIR na všetky obrazy, čo vedie k nejednoznačnosti pri vytváraní takéhoto indexu. Jeden obraz by tak mohol mať viac indexov (textových popisov) podľa toho, kto obraz posudzuje. Ďalšou z možností ako indexovať obraz, je použiť k indexácii jeho vlastnosti - farbu, textúru a tvar.

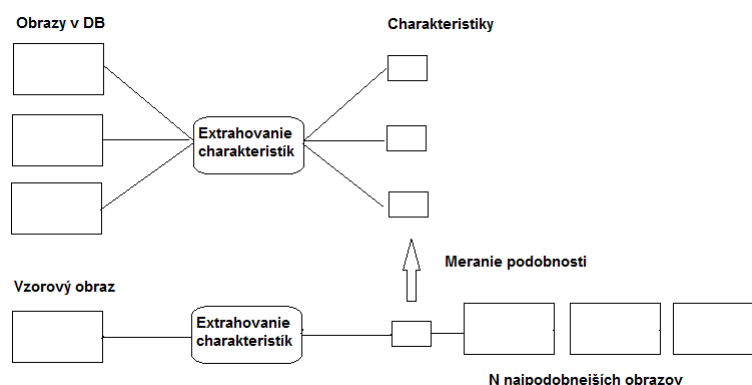
Výhodou tohto prístupu je fakt, že ide o konzistentný a objektívny prístup, nakoľko vlastnosti digitálneho obrazu sú v čase nemenné a jednoznačné. Obrazové vlastnosti sú extrahované do viacrozmerného vektora vlastností a uložené v podobe indexu obrazu v databáze. [14] Aplikáciou metriky podobnosti (viď. Funkcie vzdialenosti - 1.4) na vektory obrazových vlastností jednotlivých obrazov získame mieru podobnosti týchto obrazov.

Úroveň kvality CBIR systémov tkvie práve v metódach a spôsoboch extrahovania obrazových vlastností [21].

Fungovanie CBIR systémov možme popísať v týchto bodoch [22]:

1. Dotaz: užívateľ zadá vzorový obraz
2. Extrahovanie obrazových vlastností vzorového obrazu a vytvorenie vektora obrazových vlastností
3. Výpočet podobnosti: systém na základe zvolenej metriky porovnáva vektory obrazových vlastností vzorového obrazu s indexovanými obrazovými vlastnosťami obrazov z databázy
4. Výsledok: systém získa obrázky najpodobnejšie vzorovému obrazu a zobrazí ich užívateľovi
5. Spätná väzba: užívateľ ohodnotí jednotlivé nájdené obrázky podľa skutočnej podobnosti k vzoru. Systém sa na základe ohodnotenia adaptuje a pokračuje od kroku 2.
6. Koniec: proces končí, keď je užívateľ spokojný s výsledkom vyhľadávania

Schému CBIR systému môžeme vidieť na obrázku 1.1, ktorý uvádza [3].



Obr. 1.1: Architektúra CBIR systému

1.2 Metódy využívajúce charakteristiky farieb

Jedná sa o najviac využívané metódy používané v CBIR. Najprimitívnejšie metódy porovnávajú farbu na úrovni pixelu. Tieto metódy sú veľmi nepresné. Ostatné metódy, ktoré využívajú farbu k tvorbe indexu obrazu, sú väčšinou postavené na využití histogramu farieb v obraze [14]. Farebný histogram je graf, ktorý zobrazuje kvantitatívne zastúpenie jednotlivých farieb spektra v obraze. Samotný histogram je potom uložený v databáze a predstavuje index obrazu.

Výhodou tejto metódy je fakt, že metóda je invariantná voči geometrickým operáciám nad obrazmi - rotácia, skosenie, zmena uhlu pohľadu a pod., pretože kvantitatívne zastúpenie jednotlivých farebných zložiek v histograme je podobné aj po týchto operáciach.

Pomocou farebného histogramu môžeme nájsť dva obrazy, ktoré budú mať podobné histogramy farieb, ale scéna v obraze môže byť odlišná - červené jablko a červené auto.

1.2.1 Prienik histogramov

V literatúre [24][23] sa popisuje algoritmus prieniku histogramov. Tento algoritmus je extrémne efektívny a jednoduchý na implementáciu. Porovnáva sa histogram vzorového obrazu so všetkými histogramami obrazov v databáze. Nech je I reprezentant obrazu v databáze, M vzorový obraz a j konkrétna farba spektra. Potom môžeme definovať prienik histogramov takto:

$$\sum_{j=1}^n \min(I_j, M_j)$$

Výsledkom je počet pixelov vzorového obrazu, ktoré majú rovnakú farbu v obraze z databázy. Pokiaľ vzťah upravíme na:

$$H(I, M) = \frac{\sum_{j=1}^n \min(I_j, M_j)}{\sum_{j=1}^n M_j}$$

Máme funkciu, ktorá nám dáva normalizované hodnoty z intervalu $\langle 0, 1 \rangle$.

Metóda je odolná voči rozptylu pozadia, zmene uhlu pohľadu na objekt, zmene v rozlíšení obrazu. Metóda nie je odolná proti zmene jasu v obraze. Túto nevýhodu možno odstrániť pomocou algoritmu konštantnej farby popísanom v [4]. Senzitivita prieniku histogramov na zmenu jasu môže byť vážnym nedostatkom v real-time aplikáciách zachytávajújúcich obraz z kamery.

V literatúre [24] sa uvádza, že maximálny priestor, ktorý zaberá histogram obrazu v databáze je $\frac{(2\delta)^{n-1}}{\sqrt{n}}$, kde δ je minimálna hodnota prieniku histogramu nutná k určeniu podobnosti a n je počet buniek histogramu. Experimentálne bolo overené, že vhodná hodnota prieniku histogramov je $\delta = 0, 4$.

1.2.2 Vektor spojitých farieb

Ďalšou metódou založenou na farbe je vektor spojitých farieb - CCV (Color Coherence Vectors), ktorú môžeme nájsť v literatúre [17]. Je založená na využití histogramu farieb a orientáciou v priestore. Farebná spojitosť je určená rovnakou úrovňou farby pixelov v určitom regióne obrazu. Takéto regióny sa nazývajú komponenty - C a tvoria ho susedné pixely $p_1, p_2 \dots p_n$ s rovnakou úrovňou farby. Pre určenie spojitosti sa využíva 8-okolie alebo 4-okolie pixelu. Pokiaľ je počet pixelov v komponente väčší ako hodnota τ , sú tieto pixely vyhodnotené ako spojité. Inak sa jedná o nespojité pixely. Pred samotným vyhodnocovaním okolia pixelu je vhodné použiť priemerovací alebo medián filter, ktorý potlačí mierne odchýlky farby v okolí pixelu.

Algoritmus vytvorenia CCV je nasledovný:

1. stanovanie hodnoty τ
2. filtrácia obrazu pomocou priemerovania alebo medián filtru
3. vypočítanie histogramu farieb obrazu
4. zoskupenie hodnôt histogramu do sektorov podľa úrovne farieb
5. prechádzanie okolia pixelov a vytváranie komponent s počtom pixelov, ktoré v nej ležia
6. mapovanie farby na komponenty s počtom pixelov
7. vytvorenie CCV - určenie počtu spojitých (α) a nespojitých (β) pixelov podľa mapovania farby na komponenty

CCV sa určí pre vzorový I a porovnávaný obraz I' . Na rozdiel od klasického porovnávania histogramov sa porovnávajú zvlášť spojité a zvlášť nespojité pixely v danej farebnej bunke. Pri porovnaní CCV za použitia L1 metriky (1.9) dostaneme tento vzťah:

$$\Delta_G = \sum_{j=1}^n |\alpha_j - \alpha'_j| + |\beta_j - \beta'_j|$$

Túto metódu si môžeme uviesť na príklade z [17]. Uvažujeme hodnotu $\tau = 4$ a obraz, ktorého všetky kanály RGB (Red Green Blue) majú rovnakú hodnotu (jedná sa o obraz v odtieni šedej), čo nám umožní reprezentovať farbu pixelu jedným číslom. Máme daný príklad obrazu 6x6 po aplikácii filtru priemerovania:

| | | | | | |
|----|----|----|----|----|----|
| 22 | 10 | 21 | 22 | 15 | 16 |
| 24 | 21 | 13 | 20 | 14 | 17 |
| 23 | 17 | 38 | 23 | 17 | 16 |
| 25 | 25 | 22 | 14 | 15 | 21 |
| 27 | 22 | 12 | 11 | 21 | 20 |
| 24 | 21 | 10 | 12 | 22 | 23 |

Rozdelíme úrovne farieb do 3 buniek. 1. úroveň obsahuje hodnoty 10 - 19, 2.úroveň 20-29 atď. Dostaneme túto maticu buniek farieb:

```

2 1 2 2 1 1
2 2 1 2 1 1
2 1 3 2 1 1
2 2 2 1 1 2
2 2 1 1 2 2
2 2 1 1 2 2

```

Uurčíme komponentny na základe susedných buniek farieb pomocou 8-okolia a označíme ich písmenami A,B,...

```

B C B B A A
B B C B A A
B C D B A A
B B B A A E
B B A A E E
B B A A E E

```

Vytvoríme mapovaciú tabuľku komponenty a bunky farby. Na základe mapovacej

| | | | | | |
|---------------|----|----|---|---|---|
| Komponenta | A | B | C | D | E |
| Bunka | 1 | 2 | 1 | 3 | 1 |
| Počet pixelov | 12 | 15 | 3 | 1 | 5 |

Tab. 1.1: Mapovanie komponenty na bunky farby

tabuľky 1.1 vytvoríme tabuľku s počtom spojených (α) a nespojitých (β) pixelov v danej bunke na základe stanovenej hodnoty $\tau = 4$: Z tabuľky 1.2 vytvoríme CCV:

| | | | |
|----------|----|----|---|
| Bunka | 1 | 2 | 3 |
| α | 17 | 15 | 0 |
| β | 3 | 0 | 1 |

Tab. 1.2: Počet spojených a nespojitých pixelov

$\langle(17, 3), (15, 0), (0, 1)\rangle$.

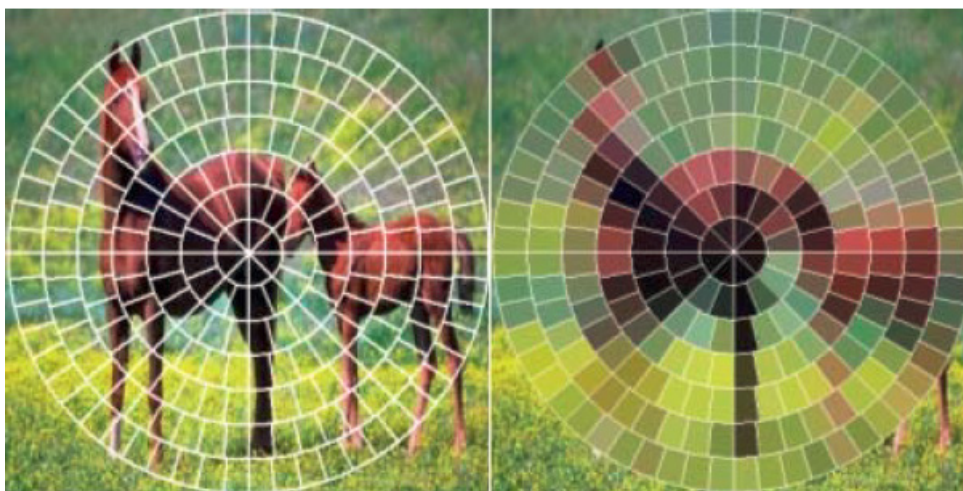
Táto metóda vykazuje lepšie výsledky ako metóda prieniku histogramu farieb[13].

1.2.3 Metóda kruhových sektorov

V literatúre [16] sa popisuje metóda založená na slabej segmentácii. Pri pozorovaní obrazu je pre ľudské oko prirodzené, že sa zameria na stred obrazu a postupuje od stredu ku krajom obrazu. Stred obrazu je tak veľmi dôležitý pri zachytávaní fotky fotoaparátom. Táto metóda využíva túto skutočnosť a jej princíp spočíva v rozdelení obrazu na sústredné kruhy C_i . Každý kruh sa ďalej rozdelí na sektory S_i tak, aby $S_i = 8 \cdot C_i$, ako môžeme vidieť na obrázku 1.2. V tomto zdroji sa uvádza ako vhodný počet kruhov 7. To znamená, že siedmy kruh bude mať $7 \cdot 8 = 56$ sektorov a celkový počet sektorov v obraze bude 252. V každom sektore sa vypočíta priemerná hodnota každého farebného kanálu farebného modelu RGB zvlášť a ten predstavuje jeden element vo vektore obrazových vlastností. Pre zvýšenie odolnosti metódy voči rotácii, sa v [16] odporúča zoradiť sektory podľa priemernej hodnoty farby v jednotlivých kanáloch vzostupne proti smeru hodinových ručičiek tak, ako môžeme vidieť na obrázku 1.3.

Pri počte 252 sektorov a 3 farebných kanáloch tak dostaneme 756 hodnôt vo vektore obrazových charakteristík v tvare:

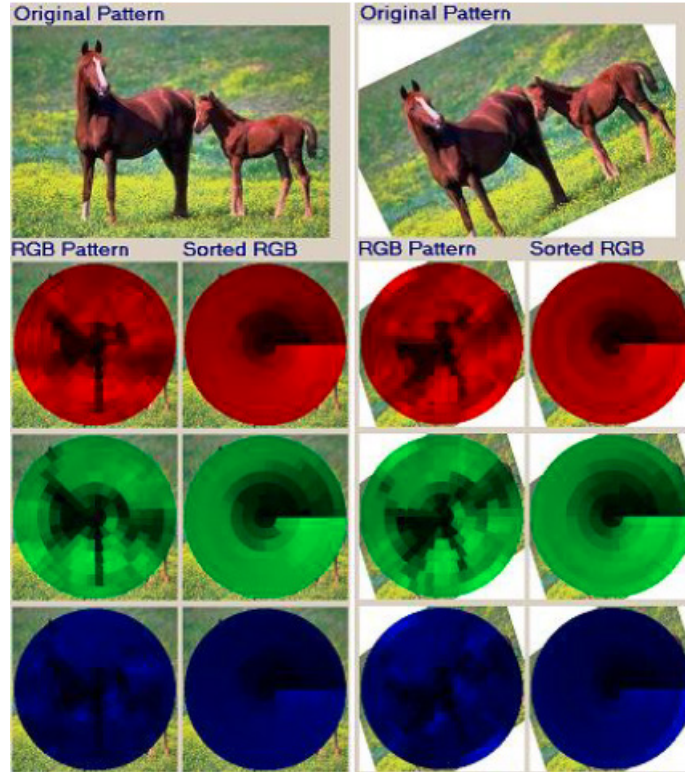
$$v = \{E_{1,1,R}, E_{1,1,G}, E_{1,1,B}, E_{1,2,R}, E_{1,2,G}, E_{1,2,B}, \dots, E_{c,s,R}, E_{c,s,G}, E_{c,s,B}\}$$



Obr. 1.2: Metóda kruhových sektorov [16]

1.2.4 Metóda momentov farieb

V [21] sa diskutuje metóda výpočtu prvých troch momentov farieb v regiónoch obrazov. Princíp metódy spočíva v rozdelení obrazu na samostatné neprekrývajúce sa



Obr. 1.3: Metóda kruhových sektorov - vplyv radenia [16]

regióny. V týchto regiónoch sa vypočítavajú momenty (prvý, druhý a tretí) farebných kanálov, ktorými sú priemer 1.1, smerodajná odchýlka 1.2 a šikmosť 1.3.

$$E_{r,i} = \frac{1}{N} \sum_{j=1}^N I_{ij} \quad (1.1)$$

$$\sigma_{r,i} = \sqrt{\left(\frac{1}{N} \sum_{j=1}^N (I_{ij} - E_{r,i})^2 \right)} \quad (1.2)$$

$$s_{r,i} = \sqrt[3]{\frac{1}{N} \sum_{j=1}^N (I_{ij} - E_{r,i})^3} \quad (1.3)$$

Parameter r predstavuje konkrétny región v obraze, i reprezentuje konkrétny farebný kanál použitého modelu pre reprezentáciu farby a j predstavuje hodnotu konkrétneho pixelu v danom kanáli i .

V tomto zdroji [21] sa odporúča použiť farebný model HSV (niekedy označovaný ako HSB), ktorý je pre vnímanie farby prirodzenejší ako farebný model RGB. Hodnoty H, S, V predstavujú farebnú hĺbku, saturáciu a hodnotu. Pokiaľ dôjde k zmene intenzity farby v modeli RGB v konkrétnom farebnom kanáli, môžeme dostať ako

výsledok úplne inú farbu. Naopak pri zmene hodnoty jedného kanálu v modeli HSV, dostávame zmenenú len danú farebnú charakteristiku - hĺbku, saturáciu alebo hodnotu. Farebný model HSV dostaneme podľa [2] nelineárnou transformáciou farebného modelu RGB pomocou rovníc 1.4, 1.5 1.6.

$$H = \cos^{-1} \frac{\frac{(R-G)+(R-B)}{2}}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \quad (1.4)$$

$$S = 1 - \frac{3[\min(R, G, B)]}{R + G + B} \quad (1.5)$$

$$V = \left(\frac{R + G + B}{3}\right) \quad (1.6)$$

Ďalej sa v [21] odporúča deliť obraz na tri horizontálne neprekrývajúce sa regióny pokrývajúce celú šírku obrazu ako možno vidieť na obrázku 1.4.

Touto metódou tak dostaneme z každého obrazu vektor farebných charakteristík o veľkosti 27, pretože v každom farebnom kanáli H,S,B počítame 3 momenty intenzity farby a toto aplikujeme pre 3 regióny. Pri použití rovnakých vyššie spomenutých premenných, môžeme vyjadriť vektor charakteristík v tvare :

$$v = \{E_{1,1}, \sigma_{1,1}, s_{1,1}, E_{1,2}, \sigma_{1,2}, s_{1,2}, \dots, E_{r,i}, \sigma_{r,i}, s_{r,i}\}$$

| |
|-----------|
| R1 |
| R2 |
| R3 |

Obr. 1.4: Momenty farieb - delenie obrazu na regióny

1.3 Metódy založené na charakteristikách textúry

Textúra v obraze predstavuje vzor, ktorý sa pravidelne opakuje. Obsahuje podstatnú informáciu o štruktúre povrchu, veľkosti a orientácii vzoru - napríklad oblaky, listy, tehly atď [21]. Analýza textúr má dlhú históriu a využívajú sa pri nej metódy založené na vlnkovej transformácii [15]. Do určitej miery môže byť táto informácia užitočná, vždy však záleží od konkrétneho obrazu.

1.3.1 Gáborove filtre

Gáborove filtre sú tvorené skupinou vlniek, ktoré zachytávajú energiu v určitej frekvencii a orientácii. Gáborove filtre sú vhodné pre textúrovú analýzu práve pre nastaviteľné parametre, ktorými sú mierka a orientácia filtra. [21]

Nech $I(x, y)$ je obraz o veľkosti PXQ . Potom podľa [7][15][21] môžeme definovať Gáborovu vlnkovú transformáciu pomocou konvolúcie :

$$G_{mn}(x, y) = \sum_s \sum_t I(x - s, y - t) \psi_{mn}^*(st)$$

kde parametre s, t predstavujú masku filtra a ψ_{mn}^* je komplexne združená funkcia k funkcii ψ_{mn} , ktorú tvorí množina funkcií generovaných z nasledujúcej funkcie pomocou zmeny mierky a orientácie:

$$\psi(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \cdot \exp(j2\pi Wx)$$

kde parameter W predstavuje modulačnú frekvenciu. $\psi(x, y)$ je Gausova krivka modulovaná komplexnou sínusoidou. Gáborove vlnky dostaneme aplikáciou funkcie:

$$\psi_{mn}(x, y) = a^{-m} \psi(\tilde{x}, \tilde{y})$$

kde m predstavuje mierku a n orientáciu vlnky. Funkcie \tilde{x} a \tilde{y} dostaneme pomocou vzťahu:

$$\begin{aligned} \tilde{x} &= a^{-m}(x \cos \theta + y \sin \theta) \\ \tilde{y} &= a^{-m}(-x \sin \theta + y \cos \theta) \end{aligned}$$

kde $a > 1$ a $\theta = n \frac{\pi}{N}$ Vyššie uvedené neznáme sú definované nasledovne:

$$\begin{aligned} a &= \left(\frac{U_h}{U_l}\right)^{\frac{1}{M-1}} \\ W_{m,n} &= a^m U_l \\ \sigma_{x,m,n} &= \frac{(a+1)\sqrt{2ln2}}{2\pi a^m (a-1)U_l} \\ \sigma_{y,m,n} &= \frac{1}{2\pi \tan\left(\frac{\pi}{2N}\right) \sqrt{\frac{U_h^2}{2ln2} - \left(\frac{1}{2\pi\sigma_{x,m,n}}\right)^2}} \end{aligned}$$

Parameter a predstavuje veľkosť mierky, U_h najvyššiu stredovú frekvenciu, U_l najnižšiu stredovú frekvenciu. Premenné σ_x a σ_y predstavujú priestorovú a pásmovú šírku v súradniciach x a y . Podľa [7] [21] je vhodné nastaviť parametre $U_h = 0,4$ a $U_l = 0,05$.

Aplikáciou Gáborových filtrov rôznych mierok a orientácii na obraz získame po konvolúcii pole magnitúd :

$$E(m, n) = \sum_x \sum_y |G_{mn}(x, y)|$$

Jednotlivé magnitúdy predstavujú energiu obsahu obrazu v rôznych mierkach a orientáciách. Obrazové vlastnosti získame po vypočítaní priemeru a smerodajnej odchýlky: [21]

$$\mu_{mn} = \frac{E(m, n)}{PXQ}$$

$$\sigma_{mn} = \frac{\sqrt{\sum_x \sum_y (|G_{mn}(x, y)| - \mu_{mn})^2}}{PXQ}$$

Hodnoty μ_{mn} a σ_{mn} tak tvoria vektor textúrových vlastností daného obrazu v databáze a môžeme ho zapísať v tvare:

$$v = \{\mu_{1,1}, \sigma_{1,1}, \mu_{1,2}, \sigma_{1,2}, \dots, \mu_{m,n}, \sigma_{m,n}\}$$

1.4 Funkcie vzdialenosti

Ako bolo spomenuté vyššie, podobnosť obrazu môžeme vyhodnocovať na základe charakteristík obrazu - farba, tvar a textúra. Tieto charakteristiky sú obsiahnuté vo vektore charakteristík obrazu [11]. Mieru podobnosti obrazu získame aplikáciou funkcie vzdialenosti na vektory charakteristík obrazov podľa zvolenej metriky [25]. Podľa [25] musia byť splnené tieto štyri axiómy metrick:

1. axióm ekvivalencie: $p(a, a) = p(b, b)$
2. axióm minima: $p(a, b) \geq p(a, a)$
3. axióm symetrie: $p(a, b) = p(b, a)$
4. axióm trojuholníkovej nerovnosti: $p(a, b) + p(b, c) \geq p(a, c)$

Kde p je funkcia vzdialenosti a a , b a c sú konkrétne body predstavujúce charakteristiky obrazov A , B a C .

Metriky vzdialenosti sa používajú v systémoch CBIR na meranie podobnosti obrazových charakteristík. Existujú rôzne funkcie merania podobností charakteristík obrazu. Nech x a y sú 2 d-rozmerné vektory obrazových charakteristík, potom môžeme definovať tieto funkcie vzdialenosti [11]:

$$d_E(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} \quad (1.7)$$

Funkciu 1.7 nazývame Euklidovská alebo L2 metrika.

$$d_{Mah}(x, y) = \sqrt{(x - y)'Cov(x)^{-1}(x - y)} \quad (1.8)$$

Funkcia 1.8 sa nazýva metrika Mahalanobis. Vhodnejšie je však použiť sumu rozdielov absolútnych hodnôt jednotlivých vlastností obrazu ako sumu rozdielov absolútnych hodnôt umocnených na druhú. Táto funkcia sa niekedy tiež nazýva mestský blok alebo Manhattanovská metrika, prípadne L1 metrika 1.9. Definujeme ju nasledovne:

$$d_M(x, y) = \sum_{i=1}^d |x_i - y_i| \quad (1.9)$$

Ďalšou funkciou pre meranie vzdialenosti obrazových vlastností, je metrika variácií váženého priemeru vyjadrená rovnicou 1.10

$$d_{Wmv}(x, y) = \sum_m \sum_n d_m(x, y). \quad (1.10)$$

kde

$$d_m(x, y) = \left| \frac{\mu_{mn}^x - \mu_{mn}^y}{\sigma(\mu_{mn})} \right| + \left| \frac{\sigma_{mn}^x - \sigma_{mn}^y}{\sigma(\mu_{mn})} \right|$$

$\sigma(\mu_{mn})$ a $\sigma(\sigma_{mn})$ sú smerodajné odchýlky obrazových vlastností a sú určené k normalizácii jednotlivých komponent obrazových vlastností, čo zväčšuje robustnosť a úspešnosť systému CBIR. Medzi najlepšie metriky patria Canberra 1.11 a Bray-Curtis 1.12 [9][11][25].

$$d_c(x, y) = \sum_{i=1}^d \frac{|x_i - y_i|}{|x_i| + |y_i|} \quad (1.11)$$

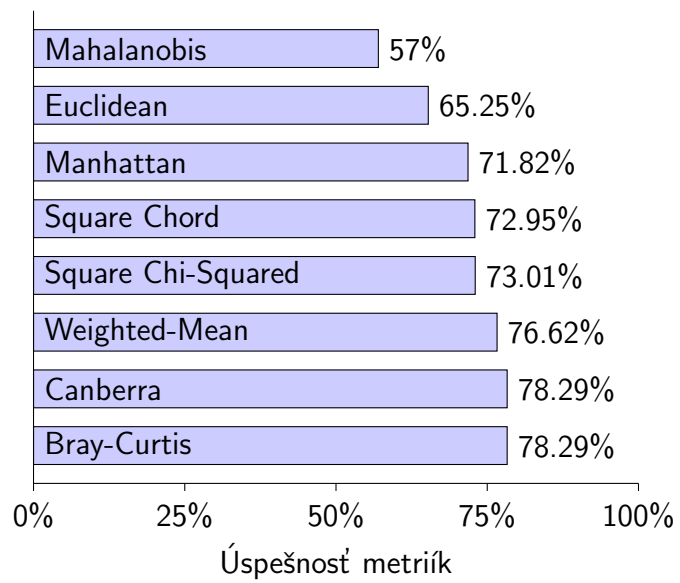
$$d_{BC}(x, y) = \sum_{i=1}^d \frac{|x_i - y_i|}{x_i + y_i} \quad (1.12)$$

V týchto metrikách čitateľ vyjadruje vzdialenosť a menovateľ ju normalizuje, takže hodnoty funkcií $d_{BC}, d_c \leq 1$. Poslednými z metrik sú metrika Squared Chord 1.13 a Square Chi-Squared 1.14.

$$d_{SC}(x, y) = \sum_{i=1}^d (\sqrt{x_i} - \sqrt{y_i})^2 \quad (1.13)$$

$$d_{chi} = \sum_{i=1}^d \frac{(x_i - y_i)^2}{x_i + y_i} \quad (1.14)$$

V literatúre [11] porovnávajú tieto metódy pre meranie podobnosti na základe textúr obrazov. Výsledkom je úspešnosť jednotlivých metód pri rovnakom vzorovom obraze. Ich výsledok môžeme vidieť v nasledujúcom grafe:



2 RIEŠENIE

2.1 Úvod

Ako bolo spomenuté v 1.1, kľúčom k úspechu pri CBIR systémoch je kvalitná extrakcia obrazových charakteristík. Každá zo spomenutých metód obsahuje modifikovateľné parametre, ktoré majú vplyv na kvalitu extrahovaných charakteristík. Je veľmi dôležité nastaviť parametre metód tak, aby systém pri vyhľadávaní podobných obrazov našiel relevantných kandidátov. Pokiaľ uvažujeme rôzne kategórie obrazov (kategória = obrazy s podobnou zachytenou scénou - napr. auto, obloha, krajina atď.) a chceme, aby metóda našla obrazy z rovnakých kategórii, je dôležité nájsť také metódy a ich parametre, aby boli zoptimalizované pre všetky kategórie obrazov. Je veľmi pravdepodobné, že aj pri nájdení metódy s najvyššou celkovou priemernou precíznosťou, budú existovať také obrazové kategórie, v ktorých môže táto metóda dosahovať nižšiu priemernú precíznosť ako pri iných metódach.

Zo spomínaných metód som sa rozhodol implementovať tieto - vektor spojených farieb 1.2.2, kruhové sektory 1.2.3, momenty farieb 1.2.4, a Gáborove filtre 1.3.1.

Pre implementáciu metód som zvolil programovací jazyk Java a využil som knižnicu ImageJ [8], ktorá poskytuje množstvo implementovaných tried a metód pre spracovanie obrazu a uľahčuje prácu pri implementovaní metód. Výstupom každej metódy je vektor charakteristík vzorového obrazu a ten je porovnávaný so všetkými vektormi charakteristík obrazov podľa schémy 1.1. Výsledkom porovnania podľa zvolenej metriky podobnosti 1.4 je premenná `diff`, ktorá je typu desiatinné číslo a určuje mieru podobnosti prípadne odlišnosti vzorového a cieľového obrazu. Nižšia hodnota `diff` znamená vyššiu podobnosť, naopak vyššia hodnota `diff` znamená nižšiu podobnosť. Pokiaľ sa `diff = 0.0`, jedná sa o identické obrazy.

2.2 Množina obrazov

Prvou úlohou pri implementovaní metód vyhľadávajúcich podobné obrazy je určiť si obrazy, s ktorými budeme pracovať. Pri vhodne zvolenej množine obrazov môžeme dosiahnuť kvalitné výsledky.

V mojej práci som použil obrazy zo zdroja [26]. Tieto obrazy sú rozdelené do desiatich kategórií a ich počet je 1000. V každej kategórii je 100 obrazov. Každý obraz má rozmery 354x256 pri orientácii na šírku, resp. 256x354 pri orientácii na výšku. Kategórie obrazov sú: starodávne budovy (ancient), pláž (beach), autobus (bus), dinosaurus (dinousaur), slon (elephant), kvet (flower), jedlo (food), kôň (horse), hora

(mountain), domorodci (natives). Na obrázku 2.1 môžeme vidieť príklady vzorových obrazov z niektorých kategórii.



Obr. 2.1: Príklad vzorových obrazov z rôznych kategórii

2.3 Vektor charakteristík

Výsledkom metód sú desatinné čísla, ktoré predstavujú obrazové charakteristiky podľa zvolenej metódy. Je dôležité tieto charakteristiky na začiatku extrahovať zo všetkých obrazov a uchovať ich. Pre ich uchovanie som sa rozhodol využiť súbor typu `.csv`. Jeden riadok v tomto súbore predstavuje konkrétny obraz. V stĺpcoch sú uložené jednotlivé charakteristiky metód s výnimkou prvých dvoch, kde sa nachádza názov kategórie obrazu (`Category`) a samostatný identifikátor obrazu (`ImageID`). Štruktúru súboru môžeme zapísať v nasledujúcom vektorovom tvare :

$$M = \begin{matrix} & \textit{Category} & \textit{ImageID}_1 & v_1 & v_2 & \cdots & v_n \\ \textit{Category} & & \textit{ImageID}_2 & v_1 & v_2 & \cdots & v_n \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ & \textit{Category} & \textit{ImageID}_z & v_1 & v_2 & \cdots & v_n \end{matrix}$$

kde v_x označuje extrahovaný vektor charakteristík konkrétnej metódy ($n = 4$), $z = 1000$.

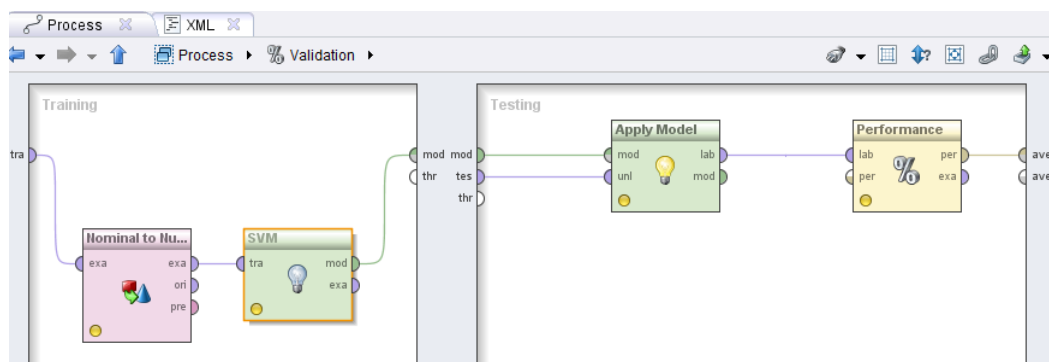
Pre väčšiu prehľadnosť a vyhodnocovanie charakteristík každej metódy zvlášť je vhodné extrahované charakteristiky uložiť do samostatných súborov. Dostaneme tak niekoľko samostatných matic v tvare M_f , kde f označuje metódu.

2.4 Hľadanie optimálnych parametrov metód

Vektor extrahovaných obrazových charakteristík obsahuje desatinné čísla. Každú kategóriu tak charakterizuje určitá množina desatinných čísel. Pomocou algoritmov umelej inteligencie môžeme natrénovať klasifikátory, ktoré budú schopné určiť podľa hodnôt vo vektore charakteristík o akú kategóriu obrazu sa jedná. Kategorizáciu obrazov použijeme tak, ako je uvedená v 2.2.

RapidMiner [19] je program, ktorý umožňuje dolovanie znalostí z bázy dát pomocou rôznych operátorov. V mojej práci som použil verziu č. 5.0. Pre vyhodnotenie kvality nastavenia parametrov jednotlivých metód som použil zložený operátor krížovej validácie [1]. Operátor pozostáva z trénovacej a testovacej časti. Vstupom trénovacej množiny sú extrahované charakteristiky z obrazov. V trénovacej časti sa určí model, pomocou ktorého sa v testovacej časti posudzuje kvalita modelu. V mojom prípade som v trénovacej časti použil operátory klasifikácie, pretože obrazy patria do známych kategórii. Výstupom testovacej časti sú výsledky z merania. V rámci merania som zvolil presnosť klasifikácie jednotlivých obrazových kategórií.

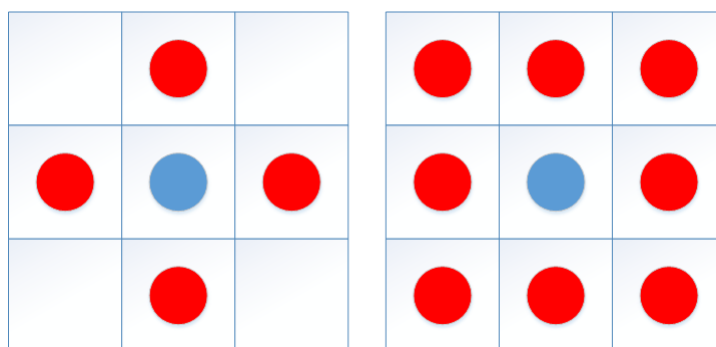
Schému operátora krížovej validácie možno vidieť na obrázku 2.2. Blok SVM(Support Vector Machine) predstavuje učiaci sa operátor z implementácie LibSVM [6]. Parametre operátora SVM a ich význam možno nájsť v [1]. U každej metódy som musel najskôr nastaviť parametre učiaceho sa operátora tak, aby dosahovala najlepšie výsledky presnosti klasifikácie. Blok Apply Model aplikuje model po natrénovaní na testovacie dáta, ktoré sú vstupom do bloku Performance. V tomto bloku sa vyhodnocuje úspešnosť klasifikácie. RapidMiner umožňuje sledovať niekoľko kritérií. Ako hlavné kritérium som zvolil presnosť klasifikácie, teda percentuálnu mieru úspešnosti predikcie voči skutočnosti. Za kvalitné nastavenie parametrov som zvolil hranicu presnosti 60%.



Obr. 2.2: RapidMiner: schéma operátora X validácia

2.5 Implementácia metódy vektora spojitéch farieb

U tejto metódy je dôležité navrhnuť algoritmus, ktorý bude prechádzať obraz a ukladať informáciu o počte spojitéch (α) a nespojitéch (β) pixeloch v rámci jednotlivých farebných buniek. Pre vyhodnotenie počtu spojitéch a nespojitéch pixelov sa využíva parameter τ . Parameter `bin` určuje počet farebných buniek histogramu. Parameter `neighb` určuje počet okolia pixelu a môže nadobúdať hodnotu 4 alebo 8. Na obrázku 2.3 vidíme 4-okolie a 8-okolie pixelu, ktorý je označený modrou farbou.



Obr. 2.3: CCV - 4-okolie, 8-okolie pixelu

2.5.1 Navrhnutý algoritmus

Na začiatku sa vytvoria mapovacie tabuľky: `regionPixel`(klúč: súradnice pixelu, hodnota: č. regiónu), `regionBin` (klúč: farebná bunka, hodnota: č. regiónu), `regionPixNr`(klúč: č. regionu, hodnota: počet pixelov v regióne), `CCV`(klúč: farebná bunka, hodnota: pole[0] - α , pole[1] - β) a nastaví sa premenná `region = 0`. Potom algoritmus pracuje podľa tejto schémy:

1. posuň sa na ďalší pixel a ulož súradnice do premennej `myPix`. Pokiaľ neexistuje ďalší pixel, pokračuj bodom 12.
2. získaj súradnice ďalšieho susedného pixelu a ulož do premennej `tmpPix`. Ak neexistuje ďalší sused, pokračuj bodom 8.
3. porovnaj intenzitu farby pixelov `myPix` a `tmpPix`
4. pokiaľ sú farby rôzne, pokračuj bodom 2
5. ak sú farby rovnaké, zisti, či existuje v tabuľke `regionPixel` záznam s kľúčom `tmpPix`
6. ak áno, ulož jeho hodnotu do `tmRregion` a vlož do tabuľky `regionPixel` záznam s kľúčom `myPix` a hodnotou `tmpRegion`. Ak nie, pokračuj bodom 2.

7. inkrementuj záznam v tabuľke `regionPixNr` s kľúčom `tmpRegion` o hodnotu 1 a pokračuj bodom 1.
8. vlož do tabuľky `regionPixel` záznam s kľúčom `myPixel` a hodnotou `region`
9. vlož do tabuľky `regionBin` záznam s kľúčom intenzity pixelu `myPix` a prirad' na koniec poľa hodnotu `region`
10. vlož do tabuľky `regionPixNr` záznam s kľúčom `region` a hodnotou 1
11. inkrementuj premennú `region` o hodnotu 1
12. ukonči, pokiaľ si na poslednom pixeli, inak pokračuj bodom 1.

Týmto algoritmom získame všetky potrebné tabuľky pre vytvorenie vektora spojitých farieb. Nasledujúcim algoritmom vytvoríme CCV, ktorý je zároveň výstupným vektorom charakteristík tejto metódy:

1. získaj ďalšiu hodnotu farebnej bunky a ulož do premennej `i`. Ak neexistuje, pokračuj bodom 10.
2. vytvor pole `alphaBeta[2]` a inicializuj hodnotou 0
3. získaj pole regiónov `regions` z tabuľky `regionBin` s kľúčom `i`. Pokiaľ záznam neexistuje, vlož na koniec poľa CCV pole `alphaBeta` a pokračuj bodom 1.
4. pokračuj na ďalší prvok v poli `regions` a ulož aktuálnu hodnotu do `region`.
5. získaj záznam z tabuľky `regionPixNr` s kľúčom `region` a ulož hodnotu do `pixnr`
6. ak je `pixnr > τ`, inkrementuj `alphaBeta[0]` hodnotou `pixnr`
7. inak inkrementuj `alphaBeta[1]` hodnotou `pixnr`
8. ak existuje ďalší záznam v poli `regions`, pokračuj bodom 4.
9. prirad' na koniec poľa CCV pole `alphaBeta`
10. ak existuje ďalšia hodnota farebnej bunky, pokračuj bodom 1, inak skonči

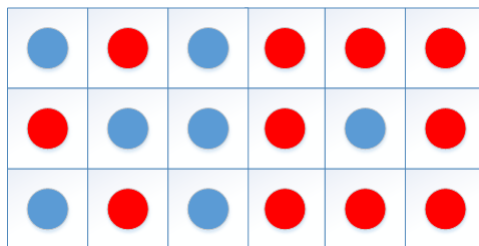
Takto získame počty spojitých (α) a nespojitých (β) pixelov v každej farebnej bunke. Výstupný vektor charakteristík je v tvare:

$$v = \{CCV_{1,\alpha}, CCV_{1,\beta}, CCV_{2,\alpha}, CCV_{2,\beta}, \dots, CCV_{i,\alpha}, CCV_{i,\beta}\}$$

Na obrázku 2.4 vidíme 2 farebné bunky a 8 regiónov pri použití 4-okolia pixelu alebo 4 regióny pri použití 8-okolia pixelu.

Vektor charakteristík vzorového obrazu 2.5 má použitím parametrov tejto metódy $\tau = 1\%$, `bin = 16` a 4-okolie pixelu nasledujúce hodnoty v prvých ôsmich farebných bunkách vo vektore charakteristík:

$$v = \{0; 10277; 0; 10187; 0; 10291; 0; 10064; 0; 10089; 0; 9942; 0; 9909; 0; 10113\}$$



Obr. 2.4: CCV - príklad regiónov a farebných buniek



Obr. 2.5: Vzorový obraz

2.5.2 Určenie optimálnych parametrov

U metódy vektora spojitých farieb som vyhodnocoval vplyv týchto parametrov na úspešnosť klasifikácie: počet okolia pixelu (4 a 8), počet farebných buniek (16, 32, 64) a parameter tau ($\tau = \{1\%, 2\%, 5\%\}$). V tabuľke 2.1 uvádzam vplyv parametrov tejto metódy na presnosť klasifikácie. V programe RapidMiner som použil učiaci sa operátor - k-NN [10] s týmito nastaveniami:

- **k:** 8
- **measure types:** MixedMeasure
- **mixed measure:** MixedEuclideanDistance

Z tabuľky 2.1 je zrejmé, že zmena parametrov metódy - počet buniek, počet okolia pixelu a tau, nemá príliš veľký vplyv na presnosť klasifikácie. Najvyššia dosiahnutá hodnota 38,4% nedosahuje požadovanej úrovne 60% stanovenej v 2.4. Táto metóda je pre moju stanovenú množinu obrazov nepoužiteľná.

V tabuľke 2.2 môžeme vidieť presnosť klasifikácie tejto metódy s nastavenými parametrami okolie pixelu - 8, počet buniek - 32 a $\tau = 2\%$ v rámci jednotlivých kategórií

| Okolie pixelu | Počet buniek | $\tau = 1\%$ | $\tau = 2\%$ | $\tau = 5\%$ |
|---------------|--------------|--------------|---------------|--------------|
| 4 | 16 | 37,6 % | 36,4 % | 36,3 % |
| 4 | 32 | 37,8 % | 37,9 % | 35,4 % |
| 4 | 64 | 35,4 % | 36,3 % | 36,8 % |
| 8 | 16 | 36,4 % | 36,7 % | 36,0 % |
| 8 | 32 | 37,8 % | 38,4 % | 35,2 |
| 8 | 64 | 34,9 % | 36,7 % | 36,8 % |

Tab. 2.1: CCV - vplyv parametrov na presnosť klasifikácie

obrazov. Najvyššiu presnosť klasifikácie dosiahla kategória dinosaur (98,82%). Najnižšiu presnosť klasifikácie dosiahla kategória ancient (22,67%).

| Predikcia | Skutočnosť | | | | | | | | | |
|-----------|------------|--------|--------|---------------|----------|--------|--------|--------|----------|---------|
| | ancient | beach | bus | dinosaur | elephant | flower | food | horse | mountain | natives |
| ancient | 17 | 12 | 7 | 6 | 2 | 4 | 4 | 1 | 18 | 4 |
| beach | 16 | 23 | 7 | 4 | 7 | 2 | 2 | 4 | 11 | 1 |
| bus | 12 | 13 | 33 | 1 | 13 | 13 | 7 | 5 | 10 | 11 |
| dinosaur | 0 | 0 | 0 | 84 | 0 | 0 | 1 | 0 | 0 | 0 |
| elephant | 8 | 15 | 14 | 0 | 47 | 6 | 16 | 19 | 18 | 10 |
| flower | 4 | 2 | 2 | 0 | 1 | 52 | 4 | 4 | 3 | 1 |
| food | 8 | 8 | 3 | 2 | 16 | 5 | 36 | 7 | 7 | 30 |
| horse | 14 | 17 | 12 | 0 | 7 | 8 | 8 | 51 | 15 | 9 |
| mountain | 4 | 7 | 1 | 3 | 1 | 2 | 3 | 0 | 7 | 0 |
| natives | 17 | 3 | 21 | 0 | 6 | 8 | 19 | 9 | 11 | 34 |
| Precíz. | 22.67% | 29.87% | 27.97% | 98.82% | 30.72% | 71.23% | 29.51% | 36.17% | 25.00% | 26.56% |

Tab. 2.2: CCV - optimálne parametre

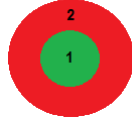
2.6 Implementácia metódy kruhových sektorov

U metódy kruhových sektorov je hlavný problém nájsť algoritmus, ktorý rozdelí obraz na sústredné kruhy a tie následne na rovnako veľké sektory tak, ako bolo spomenuté v 1.2.3. Každý sústredný kruh začína v súradniciach stredu obrazu. Nech je šírka obrazu $width$ a výška obrazu $height$. Potom má stred kruhov súradnice v bode $S = \{\frac{width}{2} - 1, \frac{height}{2} - 1\}$. Hodnota -1 je uvedená vo vzorci pre výpočet stredu, pretože ľavý horný roh má v ImageJ súradnice $0, 0$. Priemer d_i najväčšieho kruhu je rovný menšiemu z rozmerov obrazu, teda ak $height \leq width \rightarrow d_i = height$, inak $d_i = width$.

Jedna možnosť, ako rozdeliť obraz na kruhové sektory, je najskôr rozdeliť obraz na samostatné sústredné kruhy a prechádzať ich priestor tak, aby reprezentoval oddelené sektory. Vnútorň kruh pokrýva celú vnútornú plochu. Každý ďalší kruh vyplní plochu rovnú rozdielu obsahu vlastného kruhu s obsahom kruhu predchádzajúceho $S_i = S_i - S_{i-1}$, ako možno vidieť na obrázku 2.6. Pre výrez kruhov

v obraze použijeme oblasť záujmu - ROI(Region of Interest) z knižnice ImageJ - trieda `EllipseRoi`.

Vieme, že v každom kruhu C_i sa nachádza počet sektorov $s = 8 \cdot i$. Z toho vyplýva, že jednotlivé sektory sú delené pod uhlom $\frac{2\pi}{8 \cdot i}$. Napríklad pri kruhu $i = 3$ budú sektory delené pod uhlom $\frac{2\pi}{24}$. Vnútrotný priestor sektorov môžeme prechádzať pomocou algoritmu rasterizácie kružnice [12]. Existuje však efektívnejší algoritmus, ktorý popisujem v 2.6.1.



Obr. 2.6: Kruhové sektory - plocha kruhov

2.6.1 Navrhnutý algoritmus

Kruh si môžeme predstaviť ako jednotkovú kružnicu, kde os x reprezentuje funkcia \cos a os y funkcia \sin . Hodnoty týchto funkcií sú v intervale $\langle -1, 1 \rangle$. Polomer najväčšieho kruhu rozdelíme počtom kruhov, ktoré delia obraz a dostaneme vzdialenosť r medzi kruhmi. Pokiaľ premietneme výsledok goniometrických funkcií do vzdialenosti medzi kruhmi, dostaneme súradnice bodov na konkrétnej kružnici. Stačí získať hraničné body sektoru P_1, P_2, P_3, P_4 a pomocou triedy `PolygonROI` z knižnice ImageJ označiť ROI v obraze. Obsah tohto polygónu reprezentuje konkrétny sektor v kruhu.

Princíp výpočtu krajných bodov polygónu konkrétneho sektoru s_i možno vidieť na obrázku 2.7. S predstavuje stred obrazu, α je uhol vyznačeného sektoru, r je vzdialenosť medzi kruhmi. Nech s_i je konkrétny sektor a C_i konkrétny kruh. Potom súradnice bodov P_1, P_2, P_3, P_4 získame podľa nasledujúcich vzťahov :

$$\alpha_1 = s_i \cdot \left(\frac{2\pi}{8C_i}\right)$$

$$\alpha_2 = s_{i-1} \cdot \left(\frac{2\pi}{8C_i}\right)$$

$$P1_x = S_x + \cos \alpha_1 C_{i-1} r$$

$$P1_y = S_y - \sin \alpha_1 C_{i-1} r$$

$$P2_x = S_x + \cos \alpha_1 C_i r$$

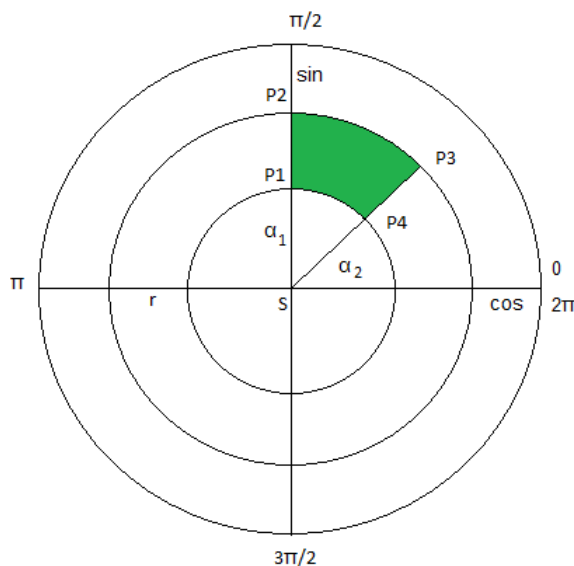
$$P2_y = S_y - \sin \alpha_1 C_i r$$

$$P3_x = S_x + \cos \alpha_2 C_i r$$

$$P3_y = S_y - \sin \alpha_2 C_i r$$

$$P4_x = S_x + \cos \alpha_2 C_{i-1} r$$

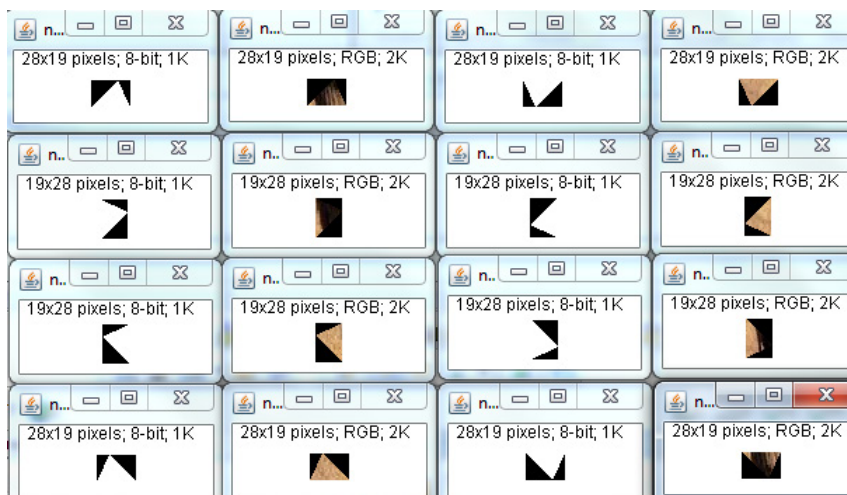
$$P4_y = S_y - \sin \alpha_2 C_{i-1} r$$



Obr. 2.7: Kruhové sektory - pohyb po sektoroch

Po aplikácii označenia ROI polygónom nám metóda `getRoi` nevráti obraz, ktorý reprezentuje tento polygón, ale najmenšiu možnú obálku, ktorá tento polygón obaľuje. Zároveň nám táto metóda vráti súradnice bodov v obraze, kde sa tento ROI nachádza. Pre získanie konkrétnych bodov, ktoré ležia v oblasti záujmu ohraničeného polygónom, musíme použiť metódu `getMask`. Táto metóda vráti binárny obraz, kde najvyššia farebná intenzita reprezentuje bod nachádzajúci sa v oblasti polygónu. Analogicky najnižšia intenzita farby predstavuje miesto, ktoré nepokrýva polygón. Aplikáciu masky a výsek sektoru prvého kruhu pomocou polygónu na originálny obraz 2.5 možno vidieť na obrázku 2.8. V prvých dvoch stĺpcoch je maska a obsah sektoru č. 1-4, v druhých dvoch stĺpcoch je maska a obsah sektoru č. 5-8.

Pomocou vrátenej masky a konkrétnych súradnic polygónu prechádzame každý pixel v sektore a ukladáme hodnoty intenzity farieb pixelov v každom farebnom kanáli RGB zvlášť do polí `R[]`, `G[]` a `B[]`. Po získaní všetkých intenzít farieb v danom sektore vypočítame priemernú hodnotu intenzity farby v každom kanáli RGB zvlášť a uložíme do polí `sR[i]`, `sG[i]` a `sB[i]`, kde `i` predstavuje konkrétny sektor. Tento postup opakujeme pre ostatné sektory v kruhu. Po získaní všetkých priemerných intenzít farieb v sektoroch daného kruhu aplikujeme radiaci algoritmus na polia `sR[]`,



Obr. 2.8: Kruhové sektory - maska a sektory prvého kruhu

$s_G[]$ a $s_B[]$, ktorý zoradí hodnoty intenzít priemerných farieb vzostupne v každom kanáli zvlášť tak, ako je uvedené na obrázku 1.3. Na radenie polí $s_R[]$, $s_G[]$ a $s_B[]$ som použil algoritmus Quicksort - varianta Dual pivot, ktorý poskytuje štandardná knižnica jazyka Java [27]. Hodnoty polí prvého kruhu obrázku 2.5 sú nasledujúce:

$$s_{1,R} = [84, 043; 73, 960; 207, 891; 202, 116; 202, 436; 208, 254; 152, 043; 78, 377]$$

$$s_{1,G} = [61, 230; 52, 399; 162, 887; 157, 138; 157, 138; 163, 633; 114, 221; 57, 556]$$

$$s_{1,B} = [44, 883; 37, 514; 113, 258; 113, 036; 115, 164; 116, 363; 83, 413; 42, 027]$$

Po aplikácii radiaceho algoritmu dostaneme tieto zoradené polia:

$$s_{1,R} = [73, 960; 78, 377; 84, 043; 152, 043; 202, 116; 202, 436; 207, 891; 208, 254]$$

$$s_{1,G} = [52, 399; 57, 556; 61, 230; 114, 221; 157, 138; 157, 138; 162, 887; 163, 633]$$

$$s_{1,B} = [37, 514; 42, 027; 44, 883; 83, 413; 113, 036; 113, 258; 115, 164; 116, 363]$$

Hodnoty zo zoradených polí tvoria výstupnú postupnosť vo vektore charakteristík v tomto tvare:

$$v = \{s_R[1], s_G[1], s_B[1], s_R[2], s_G[2], s_B[2], \dots, s_R[i], s_G[i], s_B[i]\}$$

kde i je celkový počet sektorov v obraze.

2.6.2 Modifikácie metódy

Medián

Priemerná hodnota intenzity farby nemusí presne vystihovať rozloženie prvkov v poliach. Pokiaľ máme vzostupne zoradené pole hodnôt, môžeme použiť štatistickú funkciu medián, teda strednú hodnotu štatistického súboru, a dosiahnuť možné zlepšenie metódy. Na získanie hodnôt intenzít farieb použijeme rovnaký algoritmus 2.6.1. Po aplikácii štatistickej funkcie medián na obraz 2.5 dostaneme tieto hodnoty polí v prvom kruhu obrazu:

$$s_{1,R} = [84, 0; 34, 5; 208, 0; 202, 0; 202, 0; 208, 5; 195, 0; 50, 0]$$

$$s_{1,G} = [60, 0; 21, 0; 163, 0; 157, 0; 156, 0; 163, 0; 146, 0; 36, 0]$$

$$s_{1,B} = [42, 0; 16, 0; 113, 0; 114, 0; 116, 0; 116, 0; 108, 0; 25, 0]$$

Na tieto polia aplikujeme rovnaký radiaci algoritmus a dostaneme tieto hodnoty

$$s_{1,R} = [34, 5; 50, 0; 84, 0; 195, 0; 202, 0; 202, 0; 208, 0; 208, 5]$$

$$s_{1,G} = [21, 0; 36, 0; 60, 0; 146, 0; 156, 0; 157, 0; 163, 0; 163, 0]$$

$$s_{1,B} = [16, 0; 25, 0; 42, 0; 108, 0; 113, 0; 114, 0; 116, 0; 116, 0]$$

HSB

Táto metóda využíva farebný priestor RGB, ktorý nie je príliš intuitívny pre ľudské vnímanie farby, ako bolo uvedené v 1.2.4. Existujú aj iné farebné priestory, medzi ktoré patri napríklad model HSB[20]. Opäť použijeme rovnaký algoritmus pre získavanie sektorov v obraze. Knižnica ImageJ ponúka metódu RGBtoHSB, ktorá skonvertuje intezitu farby pixelu z modelu RGB na model HSB. Označíme výstupné polia intenzít $s_H[]$, $s_S[]$ a $s_B[]$. V prvom kruhu obrazu 2.5 dostaneme tieto priemerné hodnoty intenzít farby v sektoroch:

$$s_{1,H} = [0, 066; 0, 203; 0, 087; 0, 082; 0, 080; 0, 086; 0, 118; 0, 079]$$

$$s_{1,S} = [0, 330; 0, 290; 0, 815; 0, 793; 0, 794; 0, 817; 0, 596; 0, 307]$$

$$s_{1,B} = [0, 071; 0, 077; 0, 088; 0, 081; 0, 079; 0, 085; 0, 076; 0, 071]$$

Zoradením predchádzajúcich polí získame hodnoty:

$$s_{1,H} = [0, 066; 0, 079; 0, 080; 0, 082; 0, 086; 0, 087; 0, 118; 0, 203]$$

$$s_{1,S} = [0, 431; 0, 442; 0, 442; 0, 456; 0, 500; 0, 527; 0, 534; 0, 565]$$

$$s_{1,B} = [0, 290; 0, 307; 0, 330; 0, 596; 0, 793; 0, 794; 0, 815; 0, 817]$$

HSB + medián

Pri využití farebného priestoru HSB môžeme takisto počítať medián intenzity farby v jednotlivých sektoroch. Pri aplikácii na rovnaký obraz 2.5 získame z prvého kruhu obrazu tieto hodnoty:

$$s_{1,H} = [0, 071; 0, 077; 0, 088; 0, 081; 0, 079; 0, 085; 0, 076; 0, 071]$$

$$s_{1,S} = [0, 486; 0, 515; 0, 454; 0, 437; 0, 430; 0, 441; 0, 461; 0, 465]$$

$$s_{1,B} = [0, 329; 0, 135; 0, 816; 0, 792; 0, 792; 0, 818; 0, 765; 0, 196]$$

Zoradením predchádzajúcich polí získame hodnoty:

$$s_{1,H} = [0, 071; 0, 071; 0, 076; 0, 077; 0, 079; 0, 081; 0, 085; 0, 088]$$

$$s_{1,S} = [0, 430; 0, 437; 0, 441; 0, 454; 0, 461; 0, 465; 0, 486; 0, 515]$$

$$s_{1,B} = [0, 135; 0, 196; 0, 329; 0, 765; 0, 792; 0, 792; 0, 816; 0, 818]$$

2.6.3 Určenie optimálnych parametrov

U metódy kruhových sektorov som vyhodnocoval vplyv týchto parametrov na úspešnosť klasifikácie: veľkosť obrazu (200x200, 300x300, 400x400), počet kruhov (3, 5 a 7), farebný priestor (RGB a HSB) a druh štatistickej funkcie (medián, priemer). V programe RapidMiner som použil učiaci sa operátor - SVM(LibSVM) s týmito nastaveniami:

- **SVM type:** C-SVC
- **kernel type:** linear
- **C:** 1,1
- **Epsilon:** 0,001

Z tabuľky 2.3 vyplýva niekoľko faktov :

1. farebný model HSB dosahuje vyššiu presnosť ako model RGB
2. štatistická funkcia medián dosahuje vyššiu presnosť ako priemer v oboch farebných modeloch
3. navrhnuté parametre z [16] dosahujú o 8% nižšiu úspešnosť ako najlepší výsledok zmodifikovanej metódy (67,6% < 75,6%)

V tabuľke 2.4 môžeme vidieť presnosť klasifikácie v rámci jednotlivých obrazových kategórii. Najpresnejšiu klasifikáciu dosiahla kategória dinosaur (97,09%). Najmenej presnú klasifikáciu dosiahla kategória ancient (53,45%).

| Rozmery | Kruhy | RGB priemer | RGB medián | HSB priemer | HSB medián |
|---------|-------|-------------|------------|-------------|---------------|
| 200x200 | 3 | 64,3 % | 66,0 % | 71,0 % | 71,5 % |
| 200x200 | 5 | 65,6 % | 67,3 % | 72,3 % | 72,9 % |
| 200x200 | 7 | 68,1 % | 70,5 % | 72,6 % | 74,8 % |
| 300x300 | 3 | 65,0 % | 65,6 % | 70,6 % | 72,4 % |
| 300x300 | 5 | 68,6 % | 68,9 % | 72,8 % | 72,0 % |
| 300x300 | 7 | 67,2 % | 71,3 % | 72,9 % | 74,1 % |
| 400x400 | 3 | 64,4 % | 65,5 % | 71,1 % | 72,2 % |
| 400x400 | 5 | 68,6 % | 69,0 % | 72,7 % | 72,4 % |
| 400x400 | 7 | 67,6 % | 70,8 % | 73,2 % | 75,6 % |

Tab. 2.3: Kruhové sektory - vplyv parametrov na presnosť klasifikácie

| Predikcia | Skutočnosť | | | | | | | | | |
|-----------|------------|--------|--------|---------------|----------|--------|--------|--------|----------|---------|
| | ancient | beach | bus | dinosaur | elephant | flower | food | horse | mountain | natives |
| ancient | 62 | 16 | 0 | 0 | 7 | 0 | 3 | 2 | 12 | 14 |
| beach | 11 | 62 | 3 | 0 | 1 | 0 | 5 | 2 | 20 | 2 |
| bus | 2 | 2 | 83 | 0 | 0 | 3 | 3 | 0 | 5 | 1 |
| dinosaur | 0 | 1 | 0 | 100 | 0 | 0 | 1 | 0 | 0 | 1 |
| elephant | 8 | 2 | 0 | 0 | 79 | 0 | 1 | 0 | 3 | 6 |
| flower | 0 | 0 | 4 | 0 | 0 | 88 | 4 | 0 | 0 | 2 |
| food | 0 | 3 | 4 | 0 | 0 | 7 | 72 | 1 | 2 | 11 |
| horse | 3 | 1 | 0 | 0 | 1 | 0 | 2 | 93 | 0 | 0 |
| mountain | 6 | 12 | 4 | 0 | 6 | 2 | 1 | 0 | 57 | 3 |
| natives | 8 | 1 | 2 | 0 | 6 | 0 | 8 | 2 | 1 | 60 |
| Precíz. | 53.45% | 58.49% | 83.84% | 97.09% | 79.80% | 89.80% | 72.00% | 93.00% | 62.64% | 68.18% |

Tab. 2.4: Kruhové sektory - optimálne parametre

2.7 Implementácia metódy momentov farieb

Dôležitou časťou pri tejto metóde je určiť algoritmus, ktorý obraz rozdelí na horizontálne neprekrývajúce sa regióny. Zo získaných oblastí - regiónov extrahuje intenzitu farby v každom kanáli H, S, B zvlášť, a vypočíta momenty prvého, druhého a tretieho rádu z intenzity farby tak, ako je uvedené v 1.2.4.

2.7.1 Navrhnutý algoritmus

Šírka obrazu (w) bude v jednotlivých regiónoch zachovaná, pretože delíme obraz horizontálne. Nech je parameter r počet regiónov v obraze a výška obrazu h . Pomocou nasledujúceho vzťahu získame výšku jednotlivých regiónov h_R .

$$h_R = \frac{h}{r}$$

Ľavý horný (P) a pravý dolný (Q) bod jednotlivých regiónov (i) získame podľa vzťahu:

$$P_{x,i} = 0$$

$$P_{y,i} = P_{y,i-1} + 1$$

$$Q_{x,i} = w - 1$$

$$Q_{y,i} = P_{y,i} + h_R$$

Pomocou triedy `setRoi` označíme obdĺžnikovú oblasť záujmu v obraze a predáme jej súradnice bodov P a Q .

Pri volaní metódy `getRoi` sa nám vráti oblasť záujmu v tvare obdĺžnika triedy `RectangleROI`, pretože obálka, ktorá pokrýva ROI je identická s obdĺžnikom označujúcim daný región. Pre každý kanál H,S,B obrazu potom počítame pomocou triedy `ImageStatistics` momenty prvého (E), druhého (σ) a tretieho rádu (s) volaním metód `mean`, `stdDev` a `skewness`. Vypočítané momenty jednotlivých kanálov uložíme za seba do výstupnej postupnosti vektora charakteristik v v tvare:

$$v = \{E_{H,i}; \sigma_{H,i}; s_{H,i}; E_{S,i}; \sigma_{S,i}; s_{S,i}; E_{B,i}; \sigma_{B,i}; s_{B,i}\}$$

Pri rozmeroch obrazu 400×400 a použití 3 regiónov, má obraz 2.5 vo výstupnom vektore v v prvom regióne tieto hodnoty:

$$v_1 = \{84, 311; 49, 519; 0, 574; 97, 051; 65, 858; 0, 226; 171, 552; 50, 481; -0, 546\}$$

2.7.2 Modifikácie metódy

Túto metódu môžeme opäť zmodifikovať. Jednou z modifikácií je zmena rozmerov obrazu. Ďalšou modifikáciou je použitie farebného modelu RGB.

Rozmery obrazu

Rozmery vstupného obrazu môžeme upraviť pomocou metódy `resize`, ktorej predáme parametre nových rozmerov obrazu. Zmenené rozmery obrazu môžu mať do určitej miery vplyv na kvalitu metódy.

RGB

Farebný model HSB môžeme nahradiť farebným modelom RGB a dosiahnuť určitej zmeny pri testovaní klasifikácie metód. V neprospech modelu RGB však hovorí aj výsledok testovania presnosti klasifikácie u metódy kruhových sektorov 2.6.3. Pre konverziu farby z modelu HSB do modelu RGB použijeme metódu `HSBtoRGB`, ktorá je súčasťou triedy `Color`. Vektor charakteristík má pri použití modelu RGB tento tvar:

$$v = \{E_{R,i}; \sigma_{R,i}; s_{R,i}; E_{G,i}; \sigma_{G,i}; s_{G,i}; E_{B,i}; \sigma_{B,i}; s_{B,i}\}$$

Pri rozmeroch obrazu 400×400 , použití 3 regiónov a modelu RGB, má obraz 2.5 vo výstupnom vektore v v prvom regione tieto hodnoty:

$$v_1 = \{138, 019; 63, 977; -0, 431; 144, 375; 60, 971; -0, 582; 99, 273; 60, 901; 0, 425\}$$

2.7.3 Určenie optimálnych parametrov

U metódy momentov farieb som vyhodnocoval vplyv parametrov metódy: rozmery obrazu (200×200 , 300×300 , 400×400), počet horizontálnych regiónov (1, 3, 5) a použitý farebný model (RGB, HSB) na presnosť klasifikácie. V tabuľke 2.5 uvádzam výsledky merania. V programe RapidMiner som použil učiaci sa operátor SVM(LibSVM) s týmito parametrami:

- **SVM type:** C-SVC
- **kernel type:** linear
- **C:** 0,9
- **Epsilon:** 0,001

| Rozmery | Počet regiónov | RGB | HSB |
|---------|----------------|--------|---------------|
| 200x200 | 1 | 72,4 % | 73,4 % |
| 200x200 | 3 | 72,9 % | 76,8 % |
| 200x200 | 5 | 72,1 % | 62,6 % |
| 300x300 | 1 | 72,9 % | 72,9 % |
| 300x300 | 3 | 72,3 % | 76,3 % |
| 300x300 | 5 | 71,7 % | 62,9 % |
| 400x400 | 1 | 72,4 % | 73,2 % |
| 400x400 | 3 | 72,5 % | 76,5 % |
| 400x400 | 5 | 72,4 % | 63,2 % |

Tab. 2.5: Momenty farieb - vplyv parametrov na presnosť klasifikácie

V tabuľke 2.5 môžeme vidieť, že rozmery obrazu a počet horizontálnych regiónov obrazu majú nepatrný vplyv na presnosť klasifikácie pri použití farebného modelu RGB. Naopak pri použití modelu HSB má počet regiónov väčší vplyv na presnosť klasifikácie. Navrhované nastavenie tejto metódy z [21] - rozmery 400×400 , 3 regióny a model HSB, dosiahlo presnosť klasifikácie 76,5%. Pri použití parametrov - rozmery 200×200 , 3 regióny a modelu HSB, dosiahneme o 0,3% presnejšiu klasifikáciu.

V tabuľke 2.6 je výsledok precíznosti klasifikácie tejto metódy v rámci jednotlivých kategórií obrazov pri nastavených parametroch metódy: rozmery obrazu - 200×200 , počet regiónov - 3 a použitý farebný model HSB. Táto metóda dosiahla

ako jediná precíznosť > 60% u všetkých kategórií. Najpresnejšie klasifikované boli obrázky z kategórie dinosaur (99,01%). Najmenej presne boli klasifikované obrázky z kategórie beach (62,14%).

| Predikcia | Skutočnosť | | | | | | | | | |
|-----------|------------|--------|--------|---------------|----------|--------|--------|--------|----------|---------|
| | ancient | beach | bus | dinosaur | elephant | flower | food | horse | mountain | natives |
| ancient | 71 | 12 | 7 | 0 | 6 | 4 | 1 | 0 | 7 | 6 |
| beach | 8 | 64 | 2 | 0 | 3 | 3 | 0 | 3 | 18 | 2 |
| bus | 0 | 1 | 80 | 0 | 0 | 5 | 5 | 0 | 4 | 4 |
| dinosaur | 0 | 0 | 0 | 100 | 0 | 0 | 1 | 0 | 0 | 0 |
| elephant | 8 | 5 | 0 | 0 | 76 | 0 | 2 | 2 | 6 | 9 |
| flower | 1 | 0 | 5 | 0 | 0 | 83 | 10 | 0 | 0 | 1 |
| food | 3 | 1 | 4 | 0 | 0 | 2 | 73 | 1 | 0 | 8 |
| horse | 0 | 1 | 0 | 0 | 2 | 2 | 3 | 91 | 0 | 0 |
| mountain | 3 | 14 | 1 | 0 | 5 | 1 | 1 | 0 | 61 | 1 |
| natives | 6 | 2 | 1 | 0 | 8 | 0 | 4 | 3 | 4 | 69 |
| Precíz. | 62.28% | 62.14% | 80.81% | 99.01% | 70.37% | 83.00% | 79.35% | 91.92% | 70.11% | 71.13% |

Tab. 2.6: Momenty farieb - optimálne parametre

2.8 Implementácia metódy Gáborových filtrov

Gáborove filtre nám umožňujú získať z obrazu textúrové charakteristiky. Metóda najskôr vypočíta matice filtrov a následne pomocou konvolúcie aplikuje filtre na vzorový obraz. Po filtrácii vypočíta v obraze priemernú hodnotu a smerodajnú odchýlku z intenzity farby a uloží do vektora charakteristík obrazu.

2.8.1 Navrhnutý algoritmus

Počet filtračných matíc je určený parametrami `scales`(počet mierok) a `orientations`(počet orientácií). Algoritmus prechádza každý bod x, y filtračnej matice rozmerov `width` \times `height` a vypočítava hodnotu na týchto súradniciach pomocou metódy `gaborFunction`, ktorá je implementovaná podľa rovníc uvedených v 1.3.1. Týmto spôsobom dostaneme filtračnú maticu v tvare $G_{m,n}$, kde m predstavuje konkrétnu mierku a n predstavuje konkrétnu orientáciu.

Predchádzajúci algoritmus je aplikovaný mn krát a výsledná matica $G_{m,n}$ je uložená do pola matíc $P_G[]$. Vzorový obraz je premenený do 255 odtieňov šedej pomocou metódy `convertToGray8`. Následne je obraz konvolvovaný každou maticou $P_G[i]$, kde i predstavuje konkrétny Gáborov filter. Z obrazu je po aplikácii Gáborovho filtra získaná priemerná hodnota (μ) pomocou metódy `mean` a smerodajná odchýlka intenzity šedej (σ) pomocou metódy `stdDev`, ktorá sa vloží na koniec vektora charakteristík:

$$v = \{\mu_{1,1}, \sigma_{1,1}, \mu_{1,2}, \sigma_{1,2}, \dots, \mu_{m,n}, \sigma_{m,n}\}$$

Pri počte mierok $m = 2$, orientácii $n = 4$ a veľkosti matíc 13×13 budú v poli P_G na prvých štyroch pozíciách nasledujúce matice:

$$G_{1,1} = \begin{bmatrix} -0,628 & -0,628 & -0,628 & -0,628 & -0,628 & -0,628 & -0,628 & -0,628 & -0,628 \\ -0,268 & -0,268 & -0,268 & -0,268 & -0,268 & -0,268 & -0,268 & -0,268 & -0,268 \\ 0,290 & 0,290 & 0,290 & 0,290 & 0,290 & 0,290 & 0,290 & 0,290 & 0,290 \\ 0,796 & 0,796 & 0,796 & 0,796 & 0,796 & 0,796 & 0,796 & 0,796 & 0,796 \\ 1,000 & 1,000 & 1,000 & 1,000 & 1,000 & 1,000 & 1,000 & 1,000 & 1,0 \\ 0,796 & 0,796 & 0,796 & 0,796 & 0,796 & 0,796 & 0,796 & 0,796 & 0,796 \\ 0,290 & 0,290 & 0,290 & 0,290 & 0,290 & 0,290 & 0,290 & 0,290 & 0,290 \\ -0,268 & -0,268 & -0,268 & -0,268 & -0,268 & -0,268 & -0,268 & -0,268 & -0,268 \\ -0,628 & -0,628 & -0,628 & -0,628 & -0,628 & -0,628 & -0,628 & -0,628 & -0,628 \end{bmatrix}$$

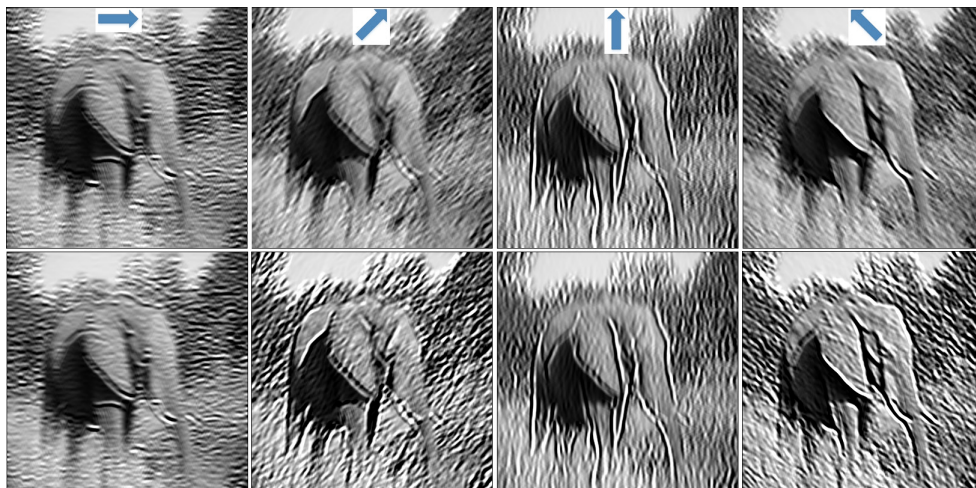
$$G_{1,2} = \begin{bmatrix} -0,552 & -0,678 & -0,669 & -0,497 & -0,181 & 0,219 & 0,611 & 0,896 & 1,0 \\ -0,678 & -0,669 & -0,497 & -0,181 & 0,219 & 0,611 & 0,896 & 1,000 & 0,896 \\ -0,669 & -0,497 & -0,181 & 0,219 & 0,611 & 0,896 & 1,000 & 0,896 & 0,611 \\ -0,497 & -0,181 & 0,219 & 0,611 & 0,896 & 1,000 & 0,896 & 0,611 & 0,219 \\ -0,181 & 0,219 & 0,611 & 0,896 & 1,000 & 0,896 & 0,611 & 0,219 & -0,181 \\ 0,219 & 0,611 & 0,896 & 1,000 & 0,896 & 0,611 & 0,219 & -0,181 & -0,497 \\ 0,611 & 0,896 & 1,000 & 0,896 & 0,611 & 0,219 & -0,181 & -0,497 & -0,669 \\ 0,896 & 1,000 & 0,896 & 0,611 & 0,219 & -0,181 & -0,497 & -0,669 & -0,678 \\ 1,000 & 0,896 & 0,611 & 0,219 & -0,181 & -0,497 & -0,669 & -0,678 & -0,552 \end{bmatrix}$$

$$G_{1,3} = \begin{bmatrix} -0,628 & -0,268 & 0,290 & 0,796 & 1,000 & 0,796 & 0,290 & -0,268 & -0,628 \\ -0,628 & -0,268 & 0,290 & 0,796 & 1,000 & 0,796 & 0,290 & -0,268 & -0,628 \\ -0,628 & -0,268 & 0,290 & 0,796 & 1,000 & 0,796 & 0,290 & -0,268 & -0,628 \\ -0,628 & -0,268 & 0,290 & 0,796 & 1,000 & 0,796 & 0,290 & -0,268 & -0,628 \\ -0,628 & -0,268 & 0,290 & 0,796 & 1,000 & 0,796 & 0,290 & -0,268 & -0,628 \\ -0,628 & -0,268 & 0,290 & 0,796 & 1,000 & 0,796 & 0,290 & -0,268 & -0,628 \\ -0,628 & -0,268 & 0,290 & 0,796 & 1,000 & 0,796 & 0,290 & -0,268 & -0,628 \\ -0,628 & -0,268 & 0,290 & 0,796 & 1,000 & 0,796 & 0,290 & -0,268 & -0,628 \end{bmatrix}$$

$$G_{1,4} = \begin{bmatrix} 1,000 & 0,896 & 0,611 & 0,219 & -0,181 & -0,497 & -0,669 & -0,678 & -0,552 \\ 0,896 & 1,000 & 0,896 & 0,611 & 0,219 & -0,181 & -0,497 & -0,669 & -0,678 \\ 0,611 & 0,896 & 1,000 & 0,896 & 0,611 & 0,219 & -0,181 & -0,497 & -0,669 \\ 0,219 & 0,611 & 0,896 & 1,000 & 0,896 & 0,611 & 0,219 & -0,181 & -0,497 \\ -0,181 & 0,219 & 0,611 & 0,896 & 1,000 & 0,896 & 0,611 & 0,219 & -0,181 \\ -0,497 & -0,181 & 0,219 & 0,611 & 0,896 & 1,000 & 0,896 & 0,611 & 0,219 \\ -0,669 & -0,497 & -0,181 & 0,219 & 0,611 & 0,896 & 1,000 & 0,896 & 0,611 \\ -0,678 & -0,669 & -0,497 & -0,181 & 0,219 & 0,611 & 0,896 & 1,000 & 0,896 \\ -0,552 & -0,678 & -0,669 & -0,497 & -0,181 & 0,219 & 0,611 & 0,896 & 1,000 \end{bmatrix}$$

Pri aplikácii 8 Gáborových filtrov pri $m = 2$, $n = 4$ a veľkosti matice 9×9 na vzorový obraz 2.5 dostaneme vyfiltrované obrazy, ktoré môžeme vidieť na obraze 2.9. Šípky naznačujú uhol, pod ktorým sa získava textúrová informácia. V prvom riadku sú obrazy filtrované mierkou $m = 1$ a v druhom riadku sú obrazy filtrované mierkou $m = 2$. Výstupný vektor charakteristík má pri týchto parametroch vstupného obrazu 2.5 nasledujúce hodnoty v prvej mierke $m = 1$:

$$v = \{127, 28; 57, 82; 127, 24; 55, 60; 127, 65; 61, 77; 127, 28; 56, 36\}$$



Obr. 2.9: Gáborove filtre - konvolúcia 8 filtrov

2.8.2 Určenie optimálnych parametrov

U metódy Gáborových filtrov som vyhodnocoval vplyv parametrov: počet mierok ($s = \{2, 4, 6\}$), počet orientácií ($o = \{2, 4, 6\}$) a rozmery filtračnej matice (9×9 , 13×13 a 33×33). Výsledky sú uvedené v tabuľke 2.7. V programe RapidMiner som použil učiaci sa operátor SVM(LibSVM) s týmito nastavenými parametrami:

- **SVM type:** C-SVC
- **kernel type:** linear
- **C:** 1,1
- **Epsilon:** 0,001

| Rozmery matice | s | o=2 | o=4 | o=6 |
|----------------|---|--------|---------------|--------|
| 9x9 | 2 | 53,3 % | 67,8 % | 68,2 % |
| 9x9 | 4 | 54,4 % | 69,1 % | 69,8 % |
| 9x9 | 6 | 53,6 % | 69,9 % | 69,1 % |
| 13x13 | 2 | 58,4 % | 72,6 % | 71,8 % |
| 13x13 | 4 | 58,8 % | 74,5 % | 74,1 % |
| 13x13 | 6 | 58,9 % | 75,2 % | 74 % |
| 33x33 | 2 | 60,9 % | 67,6 % | 68,6 % |
| 33x33 | 4 | 64,6 % | 71,8 % | 69,6 % |
| 33x33 | 6 | 64,6 % | 71,9 % | 71,3 % |

Tab. 2.7: Gáborove filtre - vplyv parametrov na presnosť klasifikácie

Zistil som, že optimálne nastavenie metódy je pri použití veľkosti filtračnej matice 13×13 , počte mierok filtrov $s = 6$ a počte orientácií filtrov $o = 4$. Rovnaká kombinácia parametrov je uvádzaná v [7] ako optimálna. Táto kombinácia nám dá celkovo 24 filtrov v podobe filtračných matíc, ktoré sa aplikujú pomocou konvolúcie na množinu obrazov a po ich aplikácii sa vyhodnocuje moment prvého a druhého rádu v obraze.

V tabuľke 2.8 môžeme vidieť presnosť klasifikácie metódy Gáborových filtrov v rámci jednotlivých obrazových kategórií pri použití optimálnych parametrov metódy. Najpresnejšiu klasifikáciu dosiahla kategória dinosaur (98%). Najmenej presnú klasifikáciu dosiahla kategória ancient (53, 54%).

| Predikcia | Skutočnosť | | | | | | | | | |
|-----------|------------|--------|--------|---------------|----------|--------|--------|--------|----------|---------|
| | ancient | beach | bus | dinosaur | elephant | flower | food | horse | mountain | natives |
| ancient | 53 | 12 | 14 | 0 | 5 | 0 | 5 | 4 | 0 | 6 |
| beach | 8 | 64 | 1 | 0 | 4 | 0 | 4 | 4 | 10 | 2 |
| bus | 19 | 1 | 85 | 0 | 0 | 0 | 2 | 0 | 1 | 0 |
| dinosaur | 0 | 0 | 0 | 98 | 1 | 0 | 0 | 0 | 0 | 1 |
| elephant | 4 | 6 | 0 | 2 | 83 | 0 | 2 | 7 | 9 | 7 |
| flower | 0 | 0 | 0 | 0 | 0 | 95 | 4 | 0 | 3 | 3 |
| food | 5 | 5 | 0 | 0 | 2 | 2 | 69 | 1 | 8 | 9 |
| horse | 6 | 3 | 0 | 0 | 2 | 2 | 1 | 84 | 1 | 0 |
| mountain | 3 | 8 | 0 | 0 | 1 | 1 | 8 | 0 | 60 | 11 |
| natives | 2 | 1 | 0 | 0 | 2 | 0 | 5 | 0 | 8 | 61 |
| Precíz. | 53.54% | 65.98% | 78.70% | 98.00% | 69.17% | 90.48% | 68.32% | 84.85% | 65.22% | 77.22% |

Tab. 2.8: Gáborove filtre - optimálne parametre

3 VÝSLEDKY PRÁCE

Metódy momentov farieb, kruhových sektorov a Gáborových filtrov dosiahli tieto hodnoty pri meraní presnosti klasifikácie v programe RapidMiner: 76,8%, 75,6% a 75,2%. Týmito meraniami som našiel optimálne nastavenie parametrov metód. Je však dôležité zistiť, aké presnosti dostávajú jednotlivé metódy pomocou navrhnutého riešenia.

Metóda vektora spojitéch farieb nedosiahla požadovanú presnosť klasifikácie 60%, preto s ňou v ďalšom vyhodnocovaní nepočítam.

3.1 Spôsob merania metód

Vo vybranej množine obrazov sa nachádza 10 kategórií obrazov. Pokiaľ systém pracuje na vstupe so vzorovým obrazom, ku ktorému hľadá podobné obrazy, môžeme kvalitu danej metódy posudzovať podľa počtu nájdených obrazov z rovnakej kategórie. Pri vyhodnocovaní som použil matematický výraz precíznosť (P). Precíznosť určuje schopnosť metódy nájsť len relevantné obrazy [21]:

$$P = \frac{A}{A + B}$$

kde A je počet nájdených relevantných obrazov a B predstavuje počet nerelevantných obrazov. Výsledok je pomerové číslo, ktoré môžeme previesť na %.

Metódy som vyhodnocoval tak, že vektor charakteristík každého obrazu v_s bol porovnávaný s každým vektorom charakteristík v_d . Pri veľkosti množiny 1000 sa jednalo o 1000000 porovnaní. V každej z desiatich kategórií je 100 obrazov. Pri porovnávaní vektorov charakteristík som použil Euklidovú1.7 a Manhattan1.9 metriku. V každej kategórii som počítal priemernú hodnotu precíznosti kategórie P_i z vrátených precízností jednotlivých obrazov P :

$$P_i = \frac{\sum_{j=1}^{100} P_j}{100}$$

Zo získaných priemerných precízností P_i som vypočítal celkový priemer precízností metódy P_c nasledovne:

$$P_c = \frac{\sum_{i=1}^{10} P_i}{10}$$

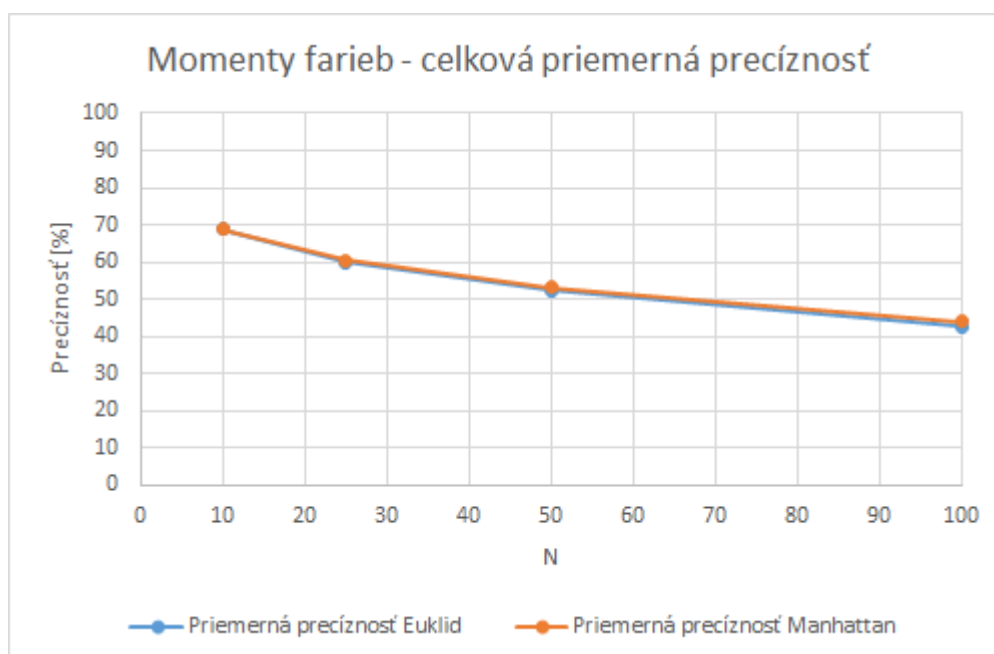
Pri každej metóde som ďalej vyhodnotil závislosť precíznosti od počtu prvých N najpodobnejších obrazov, kde $N = \{10, 25, 50, 100\}$.

3.2 Vyhodnotenie metódy momentov farieb

V tabuľke 3.1 uvádzam výsledky celkových priemerných precízností metódy momentov farieb v závislosti od počtu nájdených najpodobnejších obrazov. Tieto výsledky sú vynesené v grafe na obrázku 3.1. V tabuľkách 3.2 a 3.3 sú uvedené namerané priemerné precízności v rámci jednotlivých kategórií. Metóda mala nastavené optimálne parametre zistené v 2.7.3: rozmiery obrazu 200×200 , počet regiónov 3 a farebný model HSB.

| N | Euklid | Manhattan |
|-----|---------------|---------------|
| 10 | 68,7 % | 68,7 % |
| 25 | 59,78 % | 60,43 % |
| 50 | 52,38 % | 53,16 % |
| 100 | 42,77% | 44,07 % |

Tab. 3.1: Momenty farieb - celková priemerná precíznosť



Obr. 3.1: Momenty farieb - celková priemerná precíznosť

Táto metóda dosiahla najlepší výsledok v rámci vyhodnotenia samostatných metód. Metriky merania podobnosti Euklid a Manhattan nemajú zásadný vplyv na precíznosť metódy. U oboch metrík dosiahla metóda výsledok 68,7%

| N | Ancient | Mountain | Bus | Dinosaur | Elephant | Food | Horse | Beach | Flowers | Natives |
|-----|---------|----------|---------|---------------|----------|---------|---------|---------|---------|---------|
| 10 | 52,6 % | 51,9 % | 77 % | 99,9 % | 63,8 % | 64,9 % | 95 % | 50,6 % | 72,2 % | 59,1 % |
| 25 | 41,92 % | 43,76 % | 68,28 % | 99,16 % | 50,24 % | 52,2 % | 90,48 % | 42,04 % | 60,72 % | 49 % |
| 50 | 33,78 % | 37,66 % | 60,22 % | 97,16 % | 42,42 % | 43,06 % | 83,04 % | 36,68 % | 47,74 % | 42,04 % |
| 100 | 26,75 % | 31,39 % | 47,59 % | 86,93 % | 34,15 % | 34,42 % | 66,41 % | 30,71 % | 33,91 % | 35,46 % |

Tab. 3.2: Momenty farieb - precíznosť kat. - Euklid

| N | Ancient | Mountain | Bus | Dinosaur | Elephant | Food | Horse | Beach | Flowers | Natives |
|-----|---------|----------|---------|---------------|----------|---------|---------|---------|---------|---------|
| 10 | 50,7 % | 51,3 % | 78,1 % | 99,9 % | 65,6 % | 63,3 % | 95,1 % | 51,2 % | 71,3 % | 60,5 % |
| 25 | 41,96 % | 44,32 % | 70,32 % | 99,36 % | 52,12 % | 52,68 % | 90,44 % | 42,72 % | 58 % | 52,36 % |
| 50 | 33,68 % | 38,48 % | 62,32 % | 98,1 % | 43,76 % | 44,5 % | 84,22 % | 36,8 % | 45,66 % | 44,1 % |
| 100 | 27,44 % | 31,86 % | 50,47 % | 90,69 % | 35,82 % | 35,97 % | 67,9 % | 30,31 % | 33,03 % | 37,19 % |

Tab. 3.3: Momenty farieb - precíznosť kat. - Manhattan

3.3 Vyhodnotenie metódy kruhových sektorov

V tabuľke 3.4 uvádzam výsledky celkových priemerných precízností metódy kruhových sektorov v závislosti od počtu nájdených najpodobnejších obrazov. Tieto výsledky sú vynesené v grafe na obrázku 3.2. V tabuľkách 3.5 a 3.6 sú uvedené namerané priemerné precíznosti v rámci jednotlivých kategórii. Metóda mala nastavené optimálne parametre zistené v 2.6.3: rozmery obrazu 400x400, počet sektorov 7, farebný model HSB a štatistická funkcia medián.

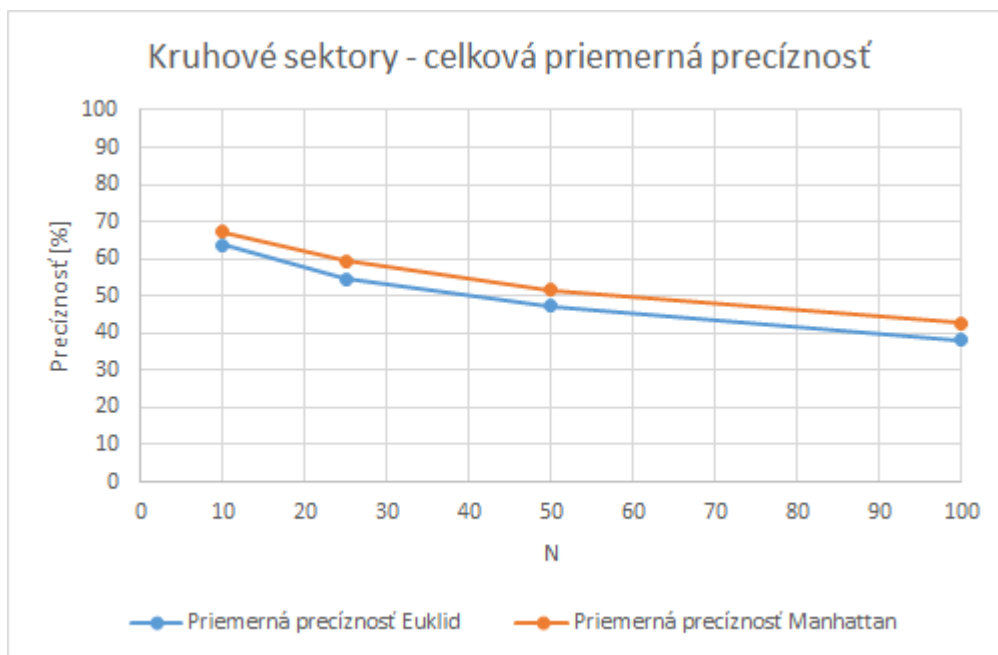
| N | Euklid | Manhattan |
|-----|---------|----------------|
| 10 | 63,75 % | 67,23 % |
| 25 | 54,57 % | 59,36 % |
| 50 | 47,22 % | 51,66 % |
| 100 | 38,25 % | 42,74 % |

Tab. 3.4: Kruhové sektory - celková priemerná precíznosť

Metóda kruhových sektorov dosiahla druhý najlepší výsledok v rámci vyhodnotenia jednotlivých metód. U tejto metódy mal vplyv metriky podobnosti výraznejší vplyv. Najlepší výsledok 67,23% dosiahla metóda pri metrike Manhattan.

3.4 Vyhodnotenie metódy Gáborových filtrov

V tabuľke 3.7 uvádzam výsledky celkových priemerných precízností metódy Gáborových filtrov v závislosti od počtu nájdených najpodobnejších obrazov. Tieto výsledky sú vynesené v grafe na obrázku 3.3. V tabuľkách 3.8 a 3.9 sú uvedené namerané priemerné precíznosti v rámci jednotlivých kategórii. Metóda mala nastavené optimálne parametre zistené v 2.8.2: rozmery filtračnej matice 13x13, počet mierok 6, počet orientácií 4.



Obr. 3.2: Kruhové sektory - celková priemerná precíznosť

| N | Ancient | Mountain | Bus | Dinosaur | Elephant | Food | Horse | Beach | Flowers | Natives |
|-----|---------|----------|--------|---------------|----------|---------|---------|---------|---------|---------|
| 10 | 44,3 % | 51,7 % | 62,5 % | 97,7 % | 65,2 % | 56,9 % | 89,8 % | 53,9 % | 72,2 % | 43,3 % |
| 25 | 35,48 % | 44,44 % | 52 % | 97,28 % | 52,48 % | 44,68 % | 82,24 % | 44,04 % | 59,16 % | 33,92 % |
| 50 | 30,04 % | 40,58 % | 42,4 % | 94,8 % | 44,72 % | 36,54 % | 73,04 % | 37,56 % | 43,96 % | 28,52 % |
| 100 | 25,73 % | 34,26 % | 34 % | 80,44 % | 36,76 % | 28,79 % | 56,24 % | 31,51 % | 29,95 % | 24,81 % |

Tab. 3.5: Kruhové sektory - precíznosť kat. - Euklid

Metóda Gáborových filtrov dosiahla najhorší výsledok v rámci vyhodnotenia jednotlivých metód. U tejto metódy mala metrika podobnosti len mierny vplyv. Najlepší výsledok 47,77% dosiahla metóda pri metrike Euklid.

3.5 Vyhodnotenie kombinácie metód

Výsledky vzdialeností metód momentov farieb d_m , kruhových sektorov d_c a Gáborových filtrov d_g som skombinoval do výslednej vzdialenosti d . Každéj metóde som pridelil váhu vplyvu metódy na mieru podobnosti v poradí w_1 , w_2 a w_3 . Váhy vychádzajú z nameraných hodnôt precíznosti jednotlivých metód. Výsledná vzdialenosť má tvar $d = 0,5d_m + 0,4d_c + 0,1d_g$.

V tabuľke 3.10 sa nachádza celková priemerná precíznosť v závislosti od počtu prvých najpodobnejších obrazov N . Graf k tejto tabuľke je na obrázku 3.4. V tabuľkách 3.11 3.12 sa nachádza priemerná precíznosť v rámci jednotlivých obrazových kategórii pri použití funkcií vzdialenosti Euklid a Manhattan.

| N | Ancient | Mountain | Bus | Dinosaur | Elephant | Food | Horse | Beach | Flowers | Natives |
|-----|---------|----------|---------|---------------|----------|---------|---------|---------|---------|---------|
| 10 | 51 % | 52 % | 63,9 % | 99,7 % | 66 % | 66 % | 92,1 % | 50,3 % | 79,2 % | 52,1 % |
| 25 | 41,76 % | 44,72 % | 54,24 % | 99,52 % | 53,64 % | 58 % | 86,96 % | 43,04 % | 67,8 % | 43,88 % |
| 50 | 33,76 % | 39,66 % | 45,22 % | 98,84 % | 44,74 % | 48,54 % | 79,18 % | 37,04 % | 51,82 % | 37,8 % |
| 100 | 28,59 % | 34,04 % | 36,23 % | 91,81 % | 36,98 % | 37,01 % | 62,42 % | 31,28 % | 36,55 % | 32,46 % |

Tab. 3.6: Kruhové sektory - precíznosť kat. - Manhattan

| N | Euklid | Manhattan |
|-----|----------------|-----------|
| 10 | 47,77 % | 46,09 % |
| 25 | 39,64 % | 38,38 % |
| 50 | 35,11 % | 34,41 % |
| 100 | 31,38 % | 30,98 % |

Tab. 3.7: Gáborove filtre - celková priemerná precíznosť

Kombináciou metód s váhovými faktormi $w_1 = 0,5$ pre metódu momentov farieb, $w_2 = 0,4$ pre metódu kruhových sektorov a $w_3 = 0,1$ pre metódu Gáborových filtrov, som dosiahol o 1,78% vyššiu celkovú priemernú precíznosť ako najvyššia dosiahnutá celková priemerná precíznosť jednotlivých metód - 68,7%.

3.6 Vyhodnotenie kategórii obrazov

Metóda s najlepšou celkovou priemernou precíznosťou nemusí dosahovať najlepšie výsledky vo všetkých kategóriách obrazov. V tabuľke 3.13 sú uvedené najlepšie metódy pre konkrétne kategórie obrazov a ich priemerné precíznosti v danej kategórii.

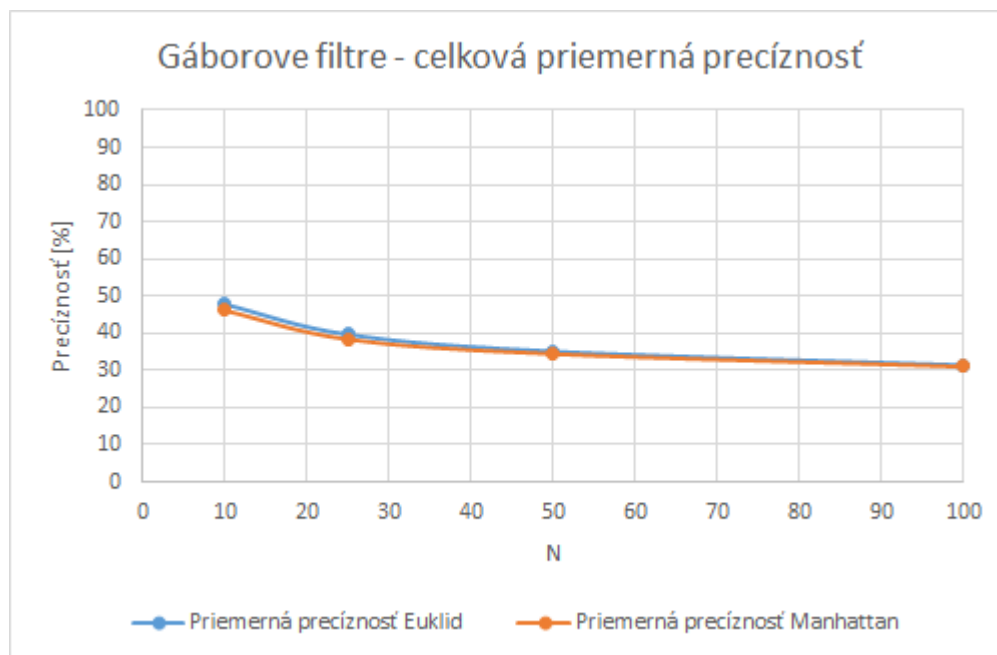
3.7 Výpočtová náročnosť

Nasledujúce výpočty som vykonal na notebooku Lenovo model T420 s procesorom Intel i5 2,5 GHz a 4GB RAM v režime optimalizovanom pre výkon.

3.7.1 Extrahovanie charakteristík

V tabuľke 3.14 uvádzam dobu nutnú k extrahovaniu obrazových charakteristík v rámci jednotlivých metód s optimálnym nastavením parametrov.

Značne dlhý čas extrahovania textúrových charakteristík je pri Gáborových filtroch zapríčinený 24000 násobnou operáciou konvolúcie s konvolučnou maticou o rozmeroch 13×13 . Túto operáciu môžeme urýchliť napríklad znížením rozmerov vstupných obrazov.



Obr. 3.3: Gáborove filtre - celková priemerná precíznosť

| N | Ancient | Mountain | Bus | Dinosaur | Elephant | Food | Horse | Beach | Flowers | Natives |
|-----|---------|----------|---------|---------------|----------|---------|---------|---------|---------|---------|
| 10 | 26,4 % | 26,8 % | 55,7 % | 98,7 % | 46,6 % | 30,8 % | 50,2 % | 44,3 % | 65,1 % | 33,1 % |
| 25 | 19,08 % | 20,16 % | 42,48 % | 98,28 % | 40,72 % | 20,72 % | 39,76 % | 35,04 % | 56,16 % | 24,04 % |
| 50 | 16,26 % | 17,28 % | 34,4 % | 98,12 % | 36,54 % | 17,82 % | 33,06 % | 28,72 % | 48,44 % | 20,46 % |
| 100 | 14,83 % | 15,53 % | 28,93 % | 95,79 % | 31,58 % | 15,87 % | 27,09 % | 22,87 % | 42,94 % | 18,35 % |

Tab. 3.8: Gáborove filtre - precíznosť kat. - Euklid

3.7.2 Vyhládavanie podobných obrazov

Všetky metódy dosiahli približne rovnakej vyhľadávaciu dobu 2s v rozpätí $\pm 0,5s$, pretože sa pre vzorový obraz vždy porovnáva všetkých 1000 obrazov.

3.8 Ukážky vyhľadávania podobných obrazov

Uvádzam 6 príkladov, na ktorých demonštrujem funkčnosť metód. Vzorové obrazy som zvolil podľa dosiahnutých výsledkov kombinácie metód uvedených v tabuľkách 3.11 a 3.12. V kategórii kôň dosahovala kombinácia metód druhý najlepší výsledok. Naopak v kategórii domorodci dosahovala kombinácia metód najhoršie výsledky.

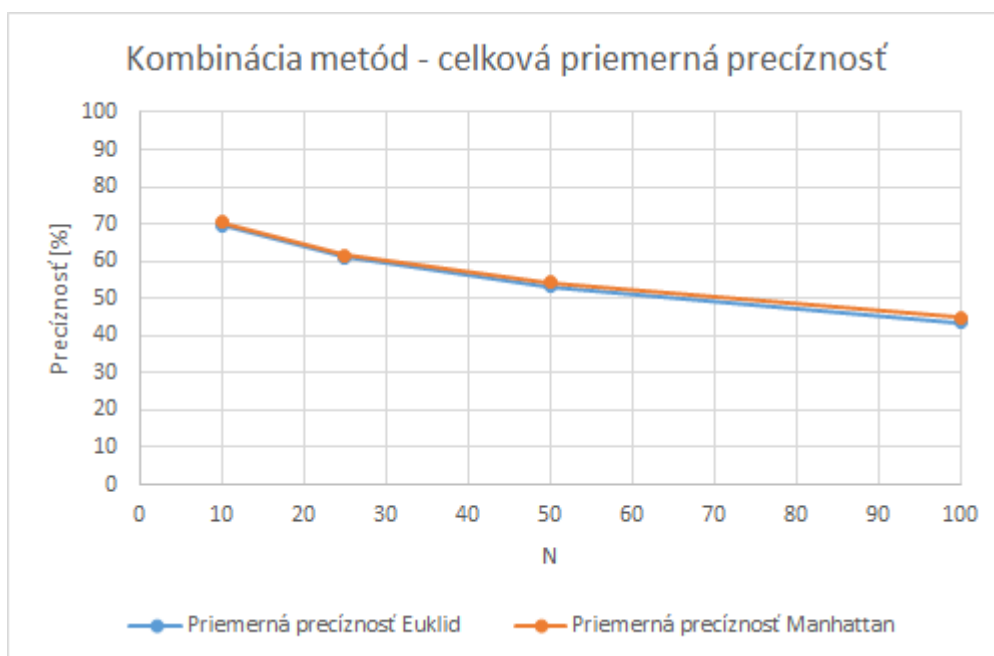
Pomocou kombinovanej metódy a metódy kruhových sektorov som hľadal 10 najpodobnejších obrazov k vzorom 3.5, 3.10 a k ich o 90° otočeným variantám 3.7 a 3.12. Nájdené obrazy k vzoru z kategórie kôň možno vidieť na obrázkoch 3.6, 3.8 a 3.9. Nájdené obrazy k vzoru z kategórie domorodci je možné vidieť na obrázkoch 3.11, 3.13 a 3.14.

| N | Ancient | Mountain | Bus | Dinosaur | Elephant | Food | Horse | Beach | Flowers | Natives |
|-----|---------|----------|---------|---------------|----------|---------|---------|---------|---------|---------|
| 10 | 25,6 % | 25,8 % | 51,1 % | 98,7 % | 46,1 % | 29,6 % | 48,3 % | 43 % | 63,1 % | 29,6 % |
| 25 | 18,68 % | 19,68 % | 38,6 % | 97,96 % | 38,96 % | 20,12 % | 38,52 % | 33,32 % | 54,88 % | 22,96 % |
| 50 | 16 % | 16,82 % | 32,02 % | 97,56 % | 36,06 % | 17,02 % | 32,34 % | 27,42 % | 48,24 % | 20,64 % |
| 100 | 14,79 % | 15,07 % | 27,81 % | 94,46 % | 31,45 % | 15,63 % | 26,84 % | 22,52 % | 42,7 % | 18,52 % |

Tab. 3.9: Gáborove filtre - precíznosť kat. - Manhattan

| N | Euklid | Manhattan |
|-----|---------|----------------|
| 10 | 69,52 % | 70,48 % |
| 25 | 61,01 % | 61,48 % |
| 50 | 53,29 % | 54,16 % |
| 100 | 43,55 % | 44,76 % |

Tab. 3.10: Kombinácia metód - celková priemerná precíznosť



Obr. 3.4: Kombinácia metód - celková priemerná precíznosť

Pod obrazmi je číslovanie, ktoré značí poradie podobnosti obrazu k vzoru a názov kategórie. Názov kategórie je vyznačený červenou, pokiaľ je nájdený obraz z inej kategórie ako vzorový obraz.

Na predchádzajúcich ukázkach je vidieť, že metóda s dosiahnutou najvyššou celkovou priemernou precíznosťou - kombinácia metód, nachádza k vzoru z kategórie kôň 10/10 obrazov z rovnakej kategórie. Vzorový obraz našla táto metóda na prvom mieste. Pomocou kombinácie metód došlo pri hľadaní podobných obrazov k vzoru

| N | Ancient | Mountain | Bus | Dinosaur | Elephant | Food | Horse | Beach | Flowers | Natives |
|-----|---------|----------|---------|---------------|----------|---------|---------|---------|---------|---------|
| 10 | 52,3 % | 51,9 % | 79,7 % | 99,7 % | 65,1 % | 65,2 % | 93,9 % | 51,9 % | 78,89 % | 56,6 % |
| 25 | 42,04 % | 43,64 % | 70,16 % | 99,2 % | 53,28 % | 53,48 % | 88,96 % | 43,52 % | 67,36 % | 48,44 % |
| 50 | 33,02 % | 37,2 % | 62,54 % | 98,32 % | 44,8 % | 43,34 % | 80,22 % | 36,7 % | 54,58 % | 42,18 % |
| 100 | 26,13 % | 30,48 % | 50,22 % | 93,11 % | 36,66 % | 33,15 % | 61,49 % | 29,94 % | 39,62 % | 34,73 % |

Tab. 3.11: Kombinácia metód - precíznosť kat. - Euklid

| N | Ancient | Mountain | Bus | Dinosaur | Elephant | Food | Horse | Beach | Flowers | Natives |
|-----|---------|----------|--------|--------------|----------|--------|--------|--------|---------|---------|
| 10 | 50,9% | 53,3% | 78,7% | 99,6% | 67,6% | 66,7% | 94% | 53,6% | 82% | 58,4% |
| 25 | 41,28% | 42,32% | 71,68% | 98,92% | 54,2% | 54,44% | 88,44% | 44,48% | 69,72% | 49,32% |
| 50 | 33,42% | 36,92% | 64,1% | 98,7% | 47,5% | 44,54% | 78,96% | 37,36% | 57,24% | 42,9% |
| 100 | 27,16% | 30,91% | 52,58% | 95,18% | 39,16% | 33,88% | 60,39% | 29,84% | 42,69% | 35,81% |

Tab. 3.12: Kombinácia metód - precíznosť kat. - Manhattan

z kategórie domorodci k poklesu precíznosti a metóda našla 5/10 obrazov rovnakej kategórie, čo potvrdzuje dosiahnuté výsledky uvedené v tabuľkách 3.11 a 3.12.

Pri hľadaní podobných obrazov k o 90° otočenému vzoru kôň, našla kombinácia metód len 2/10 obrazov z rovnakej kategórie a vzorový obraz našla až na šiestom mieste. U metódy kruhových sektorov sa potvrdila invariantnosť voči rotácii a táto metóda našla vzorový obraz na prvom mieste. Metóda zároveň našla 6/10 obrazov z rovnakej kategórie. Pri vzorovom obraze z kategórie domorodci, ktorý je otočený o 90°, nenašla kombinácia metód vzorový obraz medzi desiatimi najpodobnejšími obrazmi a našla len 3/10 obrazov rovnakej kategórie. Metóda kruhových sektorov síce našla vzorový obraz na prvom mieste, došlo tu však k poklesu precíznosti a našla len 4/10 obrazov rovnakej kategórie

3.9 Zhodnotenie

Pri vyhľadávaní obrazu z množiny, kde je niekoľko kategórii, je nutné vybrať takú metódu, ktorá dosahuje najvyššiu celkovú priemernú precíznosť, pretože je prehľadávaný celý obrazový priestor. Implementované metódy dosahovali najnižšie priemerné precíznosti v kategóriách starodávne budov (ancient), hory (mountain), pláž (beach) a domorodci (natives). Tento fakt je zapríčinený výskytom podobných hodnôt obrazových charakteristík naprieč kategóriami. V týchto kategóriách je obtiažne špecifikovať ich identitu na základe extrahovaných charakteristík pomocou implementovaných metód. Túto nevýhodu je možné eliminovať vylúčením nevhodných kandidátov pri hľadaní podobných obrazov. Vyhľadávaný priestor sa tak zníži o problematiku obrazy a môže tak byť dosiahnutá vyššia precíznosť.

Tejto práci sa naskytuje príležitosť pokračovať v problematike. Proces hľadania optimálnych parametrov metód môže byť zautomatizovaný využitím učiacich sa algoritmov a ich úspešnosti pri klasifikácii na danej množine obrazov. Učiacie sa algoritmy by mohli byť využité takisto pri vyhľadávaní podobných obrazov a mohli by

| Kategória | Precíznosť | Metóda | Metrika |
|-----------|---------------|------------------|--------------------|
| ancient | 52,6 % | momenty farieb | Euklid |
| beach | 53,9 % | kruhové sektory | Euklid |
| bus | 79,7 % | kombinácia metód | Euklid |
| dinosaur | 99,9 % | momenty farieb | Euklid + Manhattan |
| elephant | 67,6 % | kombinácia metód | Manhattan |
| flower | 82 % | kombinácia metód | Manhattan |
| food | 66,7 % | kombinácia metód | Manhattan |
| horse | 95,1 % | momenty farieb | Manhattan |
| mountain | 53,3 % | kombinácia metód | Manhattan |
| natives | 60,5 % | momenty farieb | Manhattan |

Tab. 3.13: Priemerná precíznosť metód v kategóriach

| Metóda | Čas [m:s] |
|-----------------|-----------|
| CCV | 2:22 |
| Gáborove filtre | 22:24 |
| Kruhové sektory | 1:09 |
| Momenty farieb | 0:49 |

Tab. 3.14: Výpočtová náročnosť - extrakcia charakteristík

tak dynamicky meniť parametre metód, ktoré majú vplyv na extrahované charakteristiky. Takisto je možné implementovať ďalšie metódy, ktoré extrahujú obrazové charakteristiky, a využiť napríklad charakteristiky tvaru objektov v obraze.



Obr. 3.5: Vzorový obraz pre vyhľadávanie - kôň



Obr. 3.6: Kombinácia metód - nájdené obrázky k vzoru kôň



Obr. 3.7: Vzorový obraz pre vyhľadávanie - kôň otočený o 90°



Obr. 3.8: Kombinácia metód - nájdené obrázky k o 90° otoč. vzoru kôň



Obr. 3.9: Kruhové sektory - nájdené obrázky k o 90° otoč. vzoru kôň



Obr. 3.10: Vzorový obraz pre vyhľadávanie - domorodci



Obr. 3.11: Kombinácia metód - nájdené obrázky k vzoru domorodci



Obr. 3.12: Vzorový obraz pre vyhľadávanie - domorodci otočený o 90°



Obr. 3.13: Kombinácia metód - nájdené obrázky k o 90° otoč. vzoru domorodci



Obr. 3.14: Kruhové sektory - nájdené obrázky k o 90° otoč. vzoru domorodci

4 ZÁVER

V jazyku Java som implementoval štyri metódy, ktoré umožňujú vyhodnocovať podobnosť obrazov na stanovenej množine 1000 obrazov, ktoré sú zaradené do 10 kategórií - starodávne budovy, pláž, autobus, dinosaur, slon, kvet, jedlo, kôň, hora, domorodci po 100 obrazov. Pomocou učiacich sa algoritmov som v programe Rapid-Miner 5.0 vyhodnocoval vyplv parametrov metód na presnosť klasifikácie pomocou testovacej a trérovacej množiny, ktoré vznikli z extrahovaných obrazových charakteristík jednotlivých metód. Za minimálnu hranicu klasifikácie som určil hodnotu 60%. Tejto hranice nedosiahla metóda vektora spojitých farieb a v ďalšom vyhodnocovaní som s ňou nepočítal. Metódu kruhových sektorov som zmodifikoval použitím farebného modelu HSB a štatistickej funkcie medián a dosiahol som tak o 8% vyššiu presnosť klasifikácie v porovnaní s pôvodnou metódou.

Po nájdení optimálnych parametrov metód som vyhodnocoval celkovú priemernú precíznosť metód momentov farieb, kruhových sektorov a Gáborových filtrov v závislosti od počtu najpodobnejších obrazov. Najlepší výsledok dosiahla metóda momentov farieb - 68,7%. Druhý najlepší výsledok dosiahla zmodifikovaná metóda kruhových sektorov - 67,23%. Najhorší výsledok dosiahla metóda Gáborových filtrov - 47,77%. Skombinovaním troch metód s váhovými koeficientami 0,5 pre momenty farieb, 0,4 pre kruhové sektory a 0,1 pre Gáborove filtre som dosiahol najvyššiu priemernú precíznosť - 70,48%.

Nie je možné určiť optimálnu metódu, ktorá bude fungovať pre každý obraz rovnako. Vždy je nutné vychádzať z konkrétnej množiny obrazov. V práci som preto najskôr hľadal optimálne nastavenie parametrov metód na zvolenej množine obrazov a následne vyhodnocoval metódy s týmito parametrami. Dokonca aj pri použití metódy s dosiahnutou najvyššou celkovou priemernou precíznosťou - kombinácia metód s váhovými faktormi, existujú kategórie obrazov, kde sa pohybuje priemerná precíznosť kategórie na hranici 50%. To znamená, že pri hľadaní podobných obrazov k obrazom z týchto kategórií, budú nájdené obrázky často z inej kategórie. Tento fakt je možné eliminovať vylúčením obrazov z nerelevantných kategórií z vyhodnocoacieho priestoru a môže sa tak dosiahnuť vyššia precíznosť. Takisto je možné implementovať učiace algoritmy tak, aby sami určili optimálne parametre jednotlivých metód na zvolenej množine obrazov. Toto však ponechávam na ďalší rozvoj v tejto oblasti.

LITERATÚRA

- [1] AKTHAR, Fareed, HAHNE, Caroline. *RapidMiner 5: Operator Reference*. Dortmund: Rapid-I GmbH, 2012. 990 s.
- [2] BURGER, Wilhelm, BURGE, Mark J. *Digital Image Processing: An Algorithmic Introduction using Java*. First Edition. New York: Springer Science + Business Media, 2008. 564 s. ISBN 978-1-84628-379-6.
- [3] DO, Minh N., VETTERLI, Martin. Wavelet-Based Texture Retrieval Using Generalized Gaussian Density and Kullback-Leibler Distance. *Image Processing, IEEE Transactions on*, 2002, vol. 11, no. 2, s. 146-158.
- [4] FUNT, B. V., FINLAYSON, G. D. Color Constant Color Indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1995, vol. 17, no. 5, s. 522-529.
- [5] CHALOM, Edmond, ASA, Eran, BITON, Elior. Measuring image similarity: an overview of some useful applications. *IEEE Instrumentation & Measurement Magazine, IEEE*, 2013, vol. 16, no. 1, s. 24-28.
- [6] CHANG, Chih-Chung, LIN, Chih-Jen. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2011, vol. 2, no. 3, s. 27.
- [7] CHEN, Lianping, LU, Guojun, ZHANG, Dengsheng. Effects of Different Gabor Filter Parameters on Image Retrieval by Texture In: *MMM*. 2004. s. 273-278.
- [8] *ImageJ - Image Processing and Analysis in Java*. [online] <http://rsb.info.nih.gov/ij/index.html> [cit. 2014-01-02].
- [9] JURMAN, Giuseppe, et al. Canberra distance on ranked lists. In: *Proceedings, Advances in Ranking-NIPS 09 Workshop*. 2009. s. 22-27.
- [10] KELLER, James M., GRAY, Michael R., GIVENS, James A. A fuzzy k-nearest neighbor algorithm. *Systems, Man and Cybernetics, IEEE Transactions on*, 1985, vol. 4, s. 580-585.
- [11] KOKARE, Manesh, CHATTERJI, B. N., BISWAS, P. K. Comparison of Similarity Metrics for Texture Image Retrieval. In: *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*. IEEE, 2003. s. 571-575.
- [12] KRŠEK, P. *Základy počítačové grafiky: Studijní opora*. 2003. 89 s.

- [13] MA, Wei-Ying, ZHANG, HongJiang. Benchmarking of Image Features for Content-based Retrieval. In: *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*. IEEE, 1998. s. 253-257.
- [14] MADUGUNKI, Meenakshi, et al. Comparison of different CBIR Techniques. In: *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*. IEEE, 2011. s. 372-375.
- [15] MANJUNATH, Bangalore S, MA, Wei-Ying. Texture features for browsing and retrieval of image data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1996, vol. 18, no. 8: s. 837-842.
- [16] OMAR, Samia G., ISMAIL, Mohamed A.; GHANEM, Sahar M. WAY-LOOK4: A CBIR system based on class signature of the images' color and texture features. In: *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*. IEEE, 2009. s. 464-471.
- [17] PASS, Greg, ZABIH, Ramin, MILLER, Justin. Comparing images using color coherence vectors. In: *Proceedings of the fourth ACM international conference on Multimedia*. ACM, 1997. s. 65-73.
- [18] RANGKUTI, A. Haris, et al. Analysis of Image Similarity with CBIR Concept Using Wavelet Transform and Threshold Algorithm. In: *Computers & Informatics (ISCI), 2013 IEEE Symposium on*. IEEE, 2013. s. 122-127.
- [19] *RapidMiner 5.0: Manual*. Dortmund: Rapid-I GmbH, 2010. 118 s.
- [20] SCHWARZ, Michael W., COWAN, William B., BEATTY, John C. An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models. *ACM Transactions on Graphics (TOG)*, 1987, vol. 6, no. 2, s. 123-158.
- [21] SINGH, S. Mangijao, HEMACHANDRAN, K Content-Based Image Retrieval using Color Moment and Gabor Texture Feature *International Journal of Computer Science Issues (IJCSI)*, 2012, vol.9, no.5, s. 299-309.
- [22] STEJIC, Zoran, et al. Image similarity computation using local similarity patterns generated by genetic algorithm. In: *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*. IEEE. IEEE, 2002. s. 771-776.
- [23] SWAIN, Michael J., BALLARD, Dana H. Color Indexing. *International Journal of Computer Vision*, 1991, vol. 7, no. 1, s. 11-32.

- [24] SWAIN, Michael J., BALLARD, Dana H. Indexing Via Color Histograms. In: *Active Perception and Robot Vision*. Springer Berlin Heidelberg, 1992. s. 261-273.
- [25] TAN, Huashan, et al. Digital Image Similarity Metrics and Their Performances. In: *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*. IEEE, 2011. s. 3922-3925.
- [26] *James Z. Wang Research group*. [online] <http://wang.ist.psu.edu/docs/related.shtml> [cit. 2014-05-20].
- [27] YAROSLAVSKIY, Vladimir. Dual-Pivot Quicksort. [online] *Research Disclosure RD539015 (Sept., 2009)*. <http://iaroslavski.narod.ru/quicksort/DualPivotQuicksort.pdf> [cit. 2014-05-24].

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

CBIR Content-Based Image Retrieval

RBIR Region-Based Image Retrieval

TBIR Text-Based Image Retrieval

CCV Color Coherence Vectors

RGB Red Green Blue

HSV Hue Saturation Value

HSB Hue Saturation Brightness

ROI Region of Interest

SVM Support Vector Machine

RAM Random Access Memory

ZOZNAM PRÍLOH

A Obsah CD

63

A OBSAH CD

`thesis/` - elektronická podoba tejto práce

`code/images2/` - množina obrazov

`code/ccv.csv` - extrahované charakteristiky metódou CCV

`code/circular.csv` - extrahované charakteristiky metódou kruhových sektorov

`code/cmoments.csv` - extrahované charakteristiky metódou momentov farieb

`code/gabor.csv` - extrahované charakteristiky metódou Gáborových filtrov

`code/src/.../` - zdrojový kód v jazyku Java pre prostredie Eclipse

`code/examples/.../ClassPrecision` - vyhodnotenie celk. priemernej precíznosti

`code/examples/.../ExctractFeatures` - extrahovanie obrazových charakteristík

`code/examples/.../SimilarImagesFinder` - hľadanie podobných obrazov