

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

BIOLOGICKY INSPIROVANÉ METODY ROZPOZNÁ- VÁNÍ OBJEKTŮ

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ VAŮKO

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

BIOLOGICKY INSPIROVANÉ METODY ROZPOZNÁVÁNÍ OBJEKTŮ

BIOLOGICALLY INSPIRED METHODS OF OBJECT RECOGNITION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ VAŮKO

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ROMAN JURÁNEK

BRNO 2011

Abstrakt

Rozpoznávání objektů je jedna z mnoha úloh, ve kterých počítač stále zaostává za člověkem. Proto vývoj v této oblasti využívá poznatky z přírody a především z funkce lidského mozku. Tato práce se věnuje rozpoznávání objektů pomocí extrakce významných informací z obrazu, příznaků. Příznaky se získávají podobným způsobem, jako lidský mozek zpracovává vizuální vjemy. Následně se tyto příznaky používají pro natrénování klasifikátoru (například SVM, k-NN, ANN), pomocí kterého dochází k rozpoznávání objektů. Tato práce zkoumá fázi získávání příznaků z obrazu. Jejím cílem je zdokonalit extrakci příznaků a tím zvýšit úspěšnost rozpoznávání objektů počítačem.

Abstract

Object recognition is one of many tasks in which the computer is still behind the human. Therefore, development in this area takes inspiration from nature and especially from the function of the human brain. This work focuses on object recognition based on extracting relevant information from images, features. Features are obtained in a similar way as the human brain processes visual stimuli. Subsequently, these features are used to train classifiers for object recognition (e.g. SVM, k-NN, ANN). This work examines the feature extraction stage. Its aim is to improve the feature extraction and thereby increase performance of object recognition by computer.

Klíčová slova

Support vector machine, rozpoznávání objektů, vizuální kůra, extrakce příznaků, Gáborův filtr.

Keywords

Support vector machine, object recognition, visual cortex, feature extraction, Gabor filter.

Citace

Tomáš Vaško: Biologicky inspirované metody rozpoznávání objektů, diplomová práce, Brno, FIT VUT v Brně, 2011

Biologicky inspirované metody rozpoznávání objektů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Romana Juránka.

.....
Tomáš Valko
18. května 2011

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Romanovi Juránkovi za celoroční přístup, ochotu a pomoc. Dále bych rád poděkoval své přítelkyni, své rodině a svým přátelům za podporu.

© Tomáš Valko, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Zraková sústava	4
1.1	Terminológia a skratky	5
1.2	Popis vizuálnych ciest	5
2	Príznamy	7
2.1	Gáborové filtre	8
2.2	Haarove príznaky	10
2.3	Local Binary Patterns	11
3	Metóda extrakcie príznakov	12
3.1	Dopredná architektúra	12
3.2	Popis modelu	12
4	Klasifikácia	16
4.1	Support Vector Machines	17
5	Návrh aplikácie	20
5.1	Databáze obrázkov	20
5.2	Modul 1: Výpočet príznakov	22
5.3	Modul 2: Trénovanie SVM	24
5.4	Modul 3: Kategorizácia objektu	25
6	Implementácia	26
6.1	Moduly aplikácie	27
7	Experimentálne výsledky	29
7.1	Hľadanie optimálnych parametrov SVM	29
7.2	Závislosť úspešnosti na počte orientácií	32
7.3	Závislosť úspešnosti na počte šablón	34
7.4	Dosiahnuté výsledky a porovnanie s inými metódami	37
8	Záver	39
A	Obsah DVD	42
B	Manuál	43
B.1	Ukážka konfiguračného súboru	43
B.2	Ukážka súboru so zoznamom obrázkov	44

Úvod

Rozpoznávanie objektov v obraze je široký pojem, zahŕňa detekciu, lokalizáciu či kategorizáciu objektov. Práve posledná spomenutá možnosť, kategorizácia objektov, je cieľom tejto práce. Detekcia objektov pracuje s jedinou kategóriou objektov (napr. detekcia ľudskej tváre v obraze), lokalizácia objektov pridáva k detekcii presné umiestnenie hľadaného objektu v konkrétnom obraze. Nakoniec kategorizácia objektov je proces, pri ktorom máme danú množinu rôznych kategórií a našou úlohou je priradiť obraz, ktorý obsahuje objekt, do správnej kategórie.

Kategórie reprezentujú objekty reálneho sveta (napr. autá, chodci, ľudské tváre atd.). Skúmané obrazy môžu obsahovať pozadie, iné objekty (napr. stromy) alebo je hlavný objekt čiastočne zakrytý inými objektmi, distraktormi. A k tomu ešte pracujeme s viacerými objektmi záujmu. To všetko dohromady robí z kategorizácie objektov náročnú úlohu. Ideálnou situáciou pre kategorizáciu objektov je vysoká podobnosť objektov v rámci jednej kategórie a zároveň vysoká odlišnosť objektov medzi rôznymi kategóriami.

Rozpoznávanie objektov väčšinou obsahuje výpočet množiny relevantných informácií z obrazu, tzv. príznakov. Výhody príznakov sú v redukcii objemu informácií získaných z obrazu a možnosť mapovať získané príznaky do vhodných priestorov za účelom dobrej oddeliteľnosti príznakov z rôznych tried. Používajú sa v mnohých prácach pod rôznymi označeniami ako napríklad komponenty [6], časti [26] alebo fragmenty [24]. Metódy založené na príznakoch dosahujú vynikajúce výsledky pri detekcii objektov z jednej kategórie, príkladom sú práce na detekciu tvári [20], [25], áut [20] alebo chodcov [13].

Cieľom práce je zorientovať sa v metódach rozpoznávania objektov v obraze so zameraním na metódy inšpirované biológiou. Zvoliť si jednu z nich, použiť ju na kategorizáciu objektov a následne navrhnúť rozšírenie tejto metódy. Cieľom je zvýšiť úspešnosť danej metódy pri kategorizácii, porovnať ju so súčasnými prácami a diskutovať možný vývoj do budúcnosti.

Model získavania príznakov z obrazu použitý v tejto práci vychádza z práce [19], ktorý na základe skúmania činnosti mozgu primátov pri spracovávaní vizuálnej informácie, navrhli a zostavili model, ktorý túto činnosť aproximuje. Základom modelu sú jednoduché bunky (simple cell) a zložité bunky (complex cell) popísané v práci Hubel a Wiesel [7]. Tie sú usporiadané do hierarchickej štruktúry. Táto architektúra je s najväčšou pravdepodobnosťou dopredná, lebo všetky doposiaľ známe fakty nasvedčujú, že presne tak pracuje mozog v prvých stovkách milisekúnd vizuálneho vnímania.

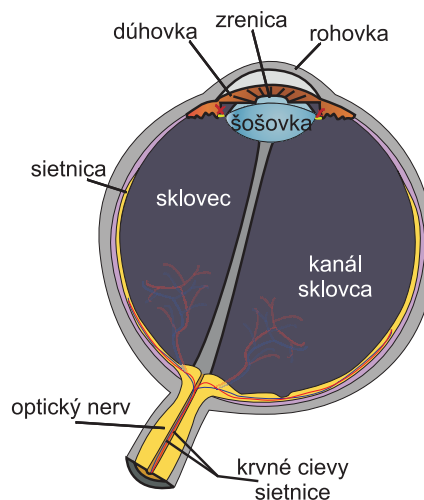
V prvej kapitole sa zoznámime s ľudským vizuálnym systémom a anatómiou ľudského zrakového ústrojenstva ako aj s šírením vizuálnej informácie v ľudskom mozgu. Druhá kapitola sa zaoberá získavaním relevantných informácií z obrazu, tzv. príznakov. Zoznámime sa v nej s Gáborovými funkciami. Ich matematickým základom a praktickým využitím pri získavaní príznakov z obrazu. V tretej kapitole sa dostávame k už spomínanému modelu inšpirovaného ľudským mozgom. Tento model bude predmetom skúmania a miestom,

kde sa dá úspešnosť výslednej kategorizácie zlepšiť. Štvrtá kapitola sa zaoberá klasifikáciou. Klasifikátor ako výsledný nástroj použitý k rozpoznávaniu a kategorizácii objektov. V tejto práci je primárne použitý Support Vector Machine (SVM). Piata kapitola obsahuje návrh aplikácie a nasledujúca šiesta kapitola jej implementáciu. Najdlhšia je siedma kapitola, ktorá prezentuje experimentálne výsledky. Záverečná ôsma kapitola sumarizuje prácu ako celok. Rozoberá splnenie zadania, dosiahnuté výsledky a diskutuje o ďalších možnostiach vývoja.

Kapitola 1

Zraková sústava

Na začiatku cesty vnímaného obrazu človekom je oko. Ľudské oko je síce komplikovaný, ale nedokonalý orgán. V tejto kapitole som čerpal informácie z [3] a [1]. Hlavné časti oka sú znázornené na obrázku 1.1. **Rohovka** je priehľadný (neobsahuje žiadne cievy), ľahko konvexný povrch v prednej časti oka. **Zrenica** je otvor pred šošovkou, ktorý ovláda množstvo svetla dopadajúce na sietnicu. **Dúhovka** je sval, ktorý ovláda priemer otvorenia zrenice a jeho pigment určuje farbu oka. **Šošovka** sústreďuje svetelné lúče na sietnicu. **Sklovec** je číra a rôsolovitá tekutina, ktorá vypĺňa vnútro oka. **Sietnica** obsahuje svetlocitlivé bunky – fotoreceptory. Fotoreceptory rozdeľujeme na tyčinky (skotopické videnie – za šera) a čapíky (fotopické videnie – farebné). V nich sa mení obraz tvorený svetelnými lúčmi na nervové vzruchy.



Obrázok 1.1: Nákres ľudskeho oka. Prevzaté z [27].

Oko vníma nejaký predmet, svetelné lúče prechádzajú cez jednotlivé časti oka a dopadajú na sietnicu. Bunky sietnice obsahujú nervové výbežky, ktoré sú zhromaždené do optického nervu. Optický nerv je cesta, po ktorej idú nervové vzruchy z oka do rôznych častí ľudskeho mozgu a ten analyzuje vizuálne signály [3].

1.1 Terminológia a skratky

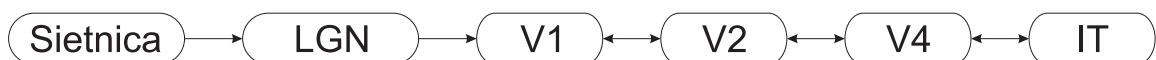
Pri popise činnosti mozgu, vizuálnych ciest a následne modelu, ktorý aproximuje tieto mechanizmy sa nevyhneme komplikovaným biologickým pojmom. Ich pomenovania vychádzajú z latinčiny a sú väčšinou ustálené v Anglickom jazyku, preto si v tejto podkapitole prehľadne popíšeme používané pojmy a ďalej budeme používať zavedené skratky alebo anglické názvy. Tak sa vyhneme nedorozumeniam plynúcich z nesprávneho prekladu špecifických pojmov.

- **Primary visual cortex** – ďalej už len **PVC**, je primárna zraková kôra, ktorá sa nachádza v temnej časti mozgu. Tu začína mozog rozpoznávať obraz zachytený na sietnici.
- **Lateral geniculate nucleus** – **LGN** spracováva signály z oka. Je prvý cieľ na ceste vizuálnej informácie k PVC. Optický nerv začína v oku a končí v LGN.
- **Striate cortex** – iné označenie používané pre primárnu zrakovú kôru, skrátene **V1**.
- **Extrastriate visual areas** – **V2** a **V4**. Z V1 pokračuje ventrálna vizuálna cesta práve cez V2 a V4, znázornené na obrázku 1.2. Všetky tri časti V1, V2 a V4 sa nachádzajú v mozgu v oblasti Occipital cortex.
- **Inferotemporal cortex** – **IT**, sa nachádza za Occipital cortexom na ventrálnej vizuálnej ceste.
- **Prefrontal cortex** – **PFC** sa podieľa na prepojení vnemu do pamäte a akcie.
- **Simple cell** – **S**, dá sa preložiť ako jednoduchá bunka, zavedená v práci Hubel a Wiesel [7].
- **Complex cell** – **C**, dá sa preložiť ako zložitá bunka, taktiež zavedená v práci Hubel a Wiesel [7].

1.2 Popis vizuálnych ciest

Obraz zachytený okom je prenášaný do mozgu optickými nervami. Tieto nervy končia v bunkách LGN. Bunky LGN posielajú informácie PVC. V nej mozog začína znovu vytvárať obraz, ktorý zachytili svetlocitlivé bunky sietnice. Poškodenie mozgu na ľubovoľnom mieste na ceste z oka do PVC vedie k rôznemu stupňu slepoty, preto sa vedci domnievajú, že práve počas tejto cesty dochádza k vedomému vizuálnemu vnímaniu.

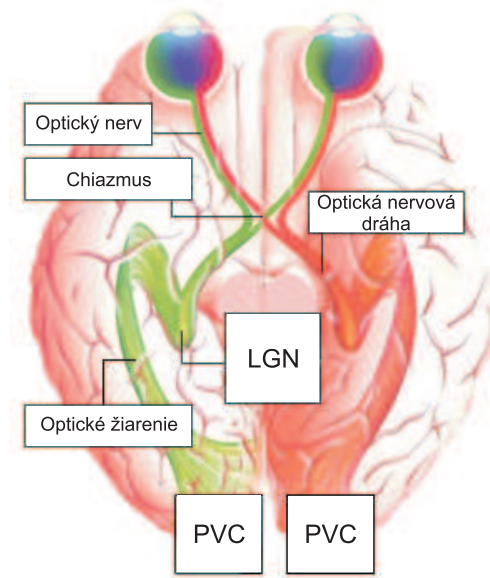
V1 je spojená s V2. Analýza vizuálneho podnetu začína vo V1 a V2. Pokračuje po ventrálnej ceste (smerujúca k prednej časti tela) a domnieva sa, že sa zúčastňuje na rozpoznávaní objektov. Nasleduje dorzálna cesta (smerujúca dozadu) a zdá sa, že je potrebná na lokalizáciu objektov.



Obrázek 1.2: Prepojenie jednotlivých častí vizuálnej cesty v mozgu.

Cesta vizuálnych signálov z oka do PVC ilustrované na obrázku 1.3:

- skríženie optických nervov (chiazmus). Niektoré nervové vzruchy si pri krížení vymenia strany, tým dôjde ku skríženému spracovaniu vizuálneho signálu (dekuzácia).
- V mozgu je pravé a ľavé LGN, pričom pravé LGN spracováva vizuálne signály zachytené ľavým okom a naopak. Spojenie LGN a PVC sa nazýva optické žiarenie.



Obrázek 1.3: Nákres vizuálnej cesty v mozgu. Prevzaté z [3].

Rozpoznávanie objektov v mozgovej kôre je ovládané ventrálnou vizuálnou cestou. Tá smeruje od PVC (V1) cez extrastriate vizuálne oblasti II (V2) a IV (V4) do IT a následne do PFC, ktorá je zapojená do spájania vnemu s pamäťou a akciou. Skúmania vedcov v oblasti mechanizmu rozpoznávania objektov mozgovou kôrou, hlavne u primátov, dospeli k viacerým kľúčovým poznaniam ventrálny vizuálnej cesty. Z V1 do IT dochádza k nárastu invariantnosti voči pozícii a veľkosti. Súčasne dochádza k nárastu veľkosti vnemových oblastí (receptive fields) ako aj k nárastu zložitosti optimálneho podnetu pre neuróny. Prispôbitosť a učenie sú pravdepodobne prítomné vo všetkých fázach, najmä v IT a PFC.

Gáborové filtre sa používajú ako model jednoduchej zmyslovej bunky v mozgu (simple cell). Jones a Palmer v roku 1987 [14] ukázali, že sa reálna časť komplexnej Gáborovej funkcie veľmi dobre hodí na popis váhovej funkcie objavenej v zmyslových bunkách mačacieho mozgu.

Kapitola 2

Príznyaky

Príznyaky (features) sú informácie získané z obrazu. Existuje viacero dôvod prečo uprednostniť používanie príznakov pri detekcii objektov v obraze pred priamym používaním pixlov daného obrazu [25]. Najčastejším dôvodom je to, že príznaky môžu kódovať ad-hoc informácie danej oblasti, ktoré sú bežne ťažko naučiteľné z konečného počtu tréningových dát. Ďalšou výhodou je, že systémy založené na rozpoznávaní objektov podľa príznakov môžu pracovať rýchlejšie ako tie založené na hodnotách pixlov.

Metódy získavania príznakov z obrazu môžeme rozdeliť podľa viacerých kritérií. Napríklad na nízko-úrovňové a vysoko-úrovňové. Medzi nízko-úrovňové patria napríklad Haarove príznaky [25], LBP (Local Binary Patterns) [15] alebo Gáborové filtre [10]. Medzi vysoko-úrovňové patria SIFT (Scale-Invariant feature transform) [12] alebo SURF (Speeded Up Robust Features) [2].

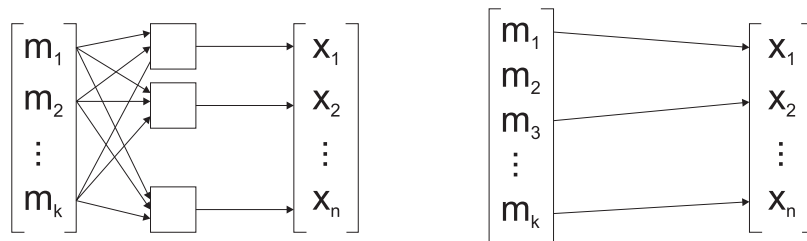
Ďalším kritériom je rozdelenie príznakov na husté (dense) a riedky (sparse). Husté príznaky sa získavajú z každého bodu obrazu. Môžeme si to predstaviť tak, že metóda prechádza sekvenčne obraz pixel po pixli a tak určí príznak respektíve príznaky pre daný obraz. Medzi metódy získavania hustých príznakov patri napríklad Haarove alebo Gáborove príznaky. Na rozdiel riedke príznaky sa určujú len z niektorých oblastí obrazu. Daná metóda si buď sama vyberá vhodné oblasti alebo jej vhodné oblasti určíme a z nich následne vypočíta riedke príznaky. Do tejto kategórie patria napríklad metódy SIFT a SURF.

Pri týchto metódach sa stretne so znižovaním dimenzionality príznakového priestoru. Znižovanie dimenzionality môžeme rozdeliť do dvoch základných skupín: **s učiteľom**, kritériom je čo najlepšia separabilita (rozlíšiteľnosť) vzorov (LDA – Linear Discriminant Analysis [5]), **bez učiteľa**, cieľom je znižovanie počtu dimenzií pri zachovaní dôležitých charakteristík (PCA – Principal Component Analysis [9]).

Dôležité je rozlišovať pojmy extrakcia príznakov (feature extraction) a výber príznakov (feature selection), ilustrované na obrázku 2.1. Pri extrakcii príznakov dochádza k redukcii počtu dimenzií lineárnou alebo nelineárnou projekciou. Pri výbere príznakov dochádza k redukcii počtu dimenzií výberom podmnožiny z pôvodných hodnôt.

Získané príznaky majú rôzne vlastnosti invariantnosti podľa použitej metódy. Môžu byť invariantné voči posuvu, rotácii, veľkosti, prekrytiu, projektívnemu skresleniu, deformácii, osvetleniu a iným zmenám objektu.

V tejto práci sú súčasťou procesu získavania príznakov použité Gáborové filtre, preto si ich popíšeme podrobnejšie. Ďalšie zo spomínaných metód si uvedieme len prehľadovo, no v kontexte získavania príznakov môžu byť použité namiesto Gáborových filtrov.



Obrázek 2.1: Rozdiel medzi extrakciou (vľavo) a selekciou (vpravo) príznakov.

2.1 Gáborové filtre

Gáborové filtre môžu slúžiť ako pásmové priepuste pre jednorozmerné signály (napr. rečový signál) [10]. Komplexný Gáborov filter je definovaný ako súčin Gaussového jadra s komplexnou sínusoidou:

$$g(t) = ke^{j\theta}\omega(at)s(t), \quad (2.1)$$

kde

$$\omega(t) = e^{-\pi t^2}, \quad (2.2)$$

$$s(t) = e^{j(2\pi f_0 t)}, \quad (2.3)$$

$$e^{j\theta}s(t)e^{j(2\pi f_0 t + \theta)} = (\sin(2\pi f_0 t + \theta), j \cos(2\pi f_0 t + \theta)). \quad (2.4)$$

Pričom parametre filtra sú k , θ , f_0 . Reálna časť filtra je popísaná vzťahom:

$$g_r(t) = \omega(t) \sin(2\pi f_0 t + \theta) \quad (2.5)$$

a imaginárna časť filtra je popísaná vzťahom:

$$g_i(t) = \omega(t) \cos(2\pi f_0 t + \theta). \quad (2.6)$$

Teraz prejdeme od jednorozmerných signálov k priestorovým. Pri filtrovaní dvojrozmerných priestorových signálov (napr. obrazový signál) použijeme komplexný priestorový Gáborov filter definovaný nasledovne:

$$g(x, y) = s(x, y)\omega_r(x, y), \quad (2.7)$$

kde $s(x, y)$ je komplexná sínusoida, označovaná ako **nosič** (carrier) a $\omega_r(x, y)$ je 2D Gaussova funkcia, označovaná ako **obálka** (envelope). Ukážka na obrázku 2.2.

Komplexná sínusoida (nosič) je definovaná nasledovne:

$$s(x, y) = \exp(j(2\pi(u_0x + v_0y) + P)), \quad (2.8)$$

kde (u_0, v_0) a P definujú priestorovú frekvenciu a fázu sínusoidy.

Gaussova obálka je definovaná vzťahom:

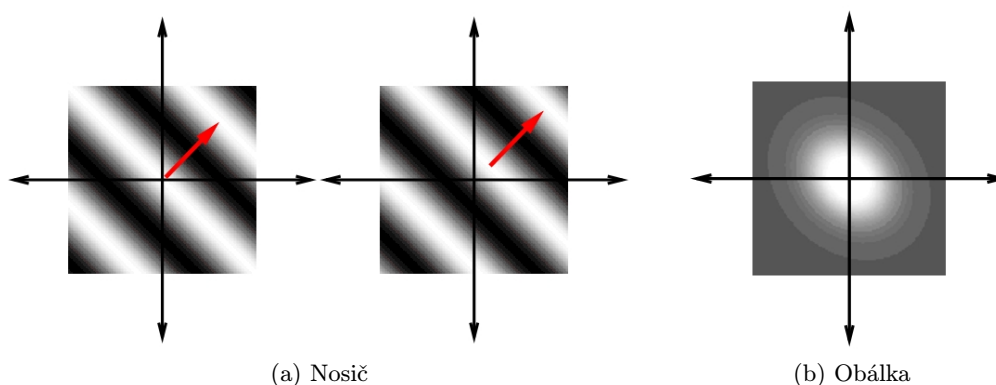
$$\omega_r(x, y) = K \exp(-\pi(a^2(x - x_0)_r^2 + b^2(y - y_0)_r^2)). \quad (2.9)$$

Kde (x_0, y_0) sú vrchol funkcie, a a b sú parametre zmeny veľkosti Gaussovej funkcie (väčšie a , b znamená menšiu funkciu) a index r označuje operáciu rotácie (rotácia je po smeru hodinových ručičiek) takto:

$$(x - x_0)_r = (x - x_0) \cos \theta + (y - y_0) \sin \theta, \quad (2.10)$$

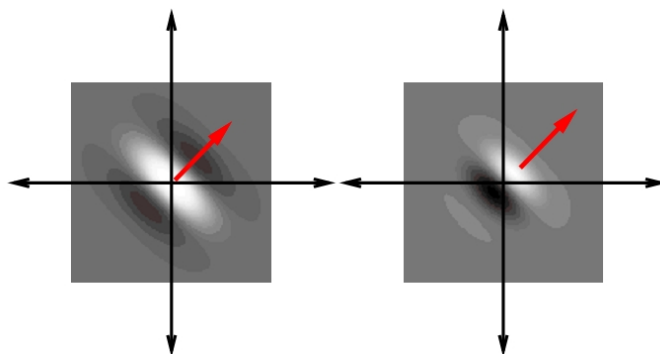
$$(y - y_0)_r = -(x - x_0) \sin \theta + (y - y_0) \cos \theta. \quad (2.11)$$

Komplexná Gáborová funkcia je definovaná nasledujúcimi 9 parametrami:



Obrázek 2.2: Ukážka vľavo je reálna a imaginárna časť komplexnej sínusoidy a vpravo je 2D Gaussova funkcia. Prevzaté z [14].

- K : určuje veľkosť Gaussovej obálky,
- (a, b) : určujú veľkosť hlavnej a vedľajšej osy Gaussovej obálky,
- θ : uhol rotácie Gaussovej obálky,
- (x_0, y_0) : miesto vrcholu funkcie Gaussovej obálky,
- (u_0, v_0) : priestorové frekvencie sínusoidy v karteziánskej sústave, v polárnej dvojica hodnôt F_0, w_0 ,
- P : fázové posunutie sínusoidy nosiča.



Obrázek 2.3: Reálna a imaginárna časť komplexnej Gáborovej funkcie. Veľkosť obrázkov je 128×128 pixlov. Parametre sú: $x_0 = y_0 = 0$, $a = 1/50$ pixlov, $b = 1/40$ pixlov, $\theta = -45^\circ$, $F_0 = \sqrt{2}/80$ periódy /pixel, $\omega_0 = 45^\circ$, $P = 0^\circ$. Prevzaté z [14].

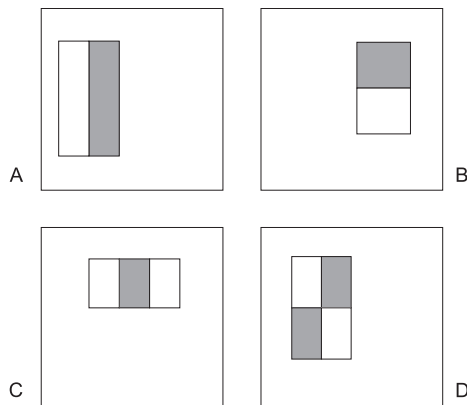
Výsledný vzťah komplexnej Gáborovej funkcie [10]:

$$g(x, y) = K \exp(-\pi(a^2(x - x_0)_r^2 + b^2(y - y_0)_r^2)) \exp(j(2\pi(u_0x + v_0y) + P)). \quad (2.12)$$

V kontexte získavania príznakov z obrazu sa z prezentovanej Gáborovej funkcie používa reálna zložka. Tá nám vygeneruje konvolučné jadro, s ktorým sa následne konvuluje filtrovaný obraz.

2.2 Haarove príznaky

Ide o množinu bázových funkcií, ktoré popisujú rozdiely v priemerných intenzitách medzi jednotlivými oblasťami obrazu [18].



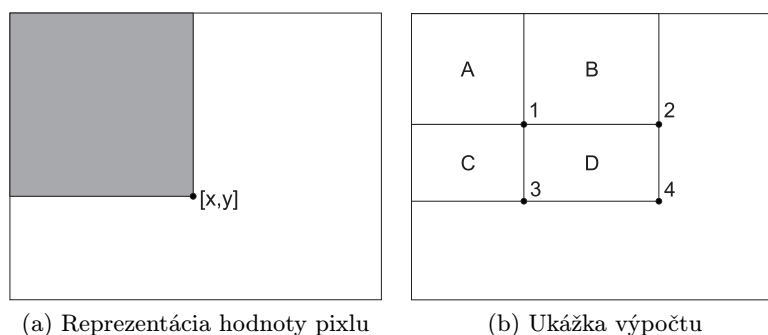
Obrázok 2.4: Haarove príznaky. Prevzaté z [25].

Obrázok 2.4 obsahuje ukážku obdĺžnikových príznakov umiestnených v obraze. Suma hodnôt pixlov, ktoré ležia v bielej oblasti obdĺžnika je odčítaná od sumy hodnôt pixlov v šedej oblasti. *A*, *B* zobrazujú 2 obdĺžnikové príznaky, *C* zobrazuje 3 a *D* zobrazuje 4 obdĺžnikové príznaky.

Príznaky v tvare pravouholníka, ako sú Haarove príznaky, sú citlivé na prítomnosť hrán, pruhov a iných jednoduchých obrazových štruktúr. Základný model pracuje v dvoch orientáciách: vertikálnej a horizontálnej.

S používaním pravouholníkových príznakov zaviedli [25] novú reprezentáciu obrazu nazvanú integrálny obraz, ktorá umožňuje veľmi rýchly výpočet príznakov. Integrálny obraz na pozícii $[x,y]$ obsahuje hodnotu sumy všetkých pixlov, ktoré sa nachádzajú v obdĺžnikovej oblasti medzi bodmi $[0,0]$ a $[x,y]$, viz. obrázok 2.5a.

Na obrázku 2.5b si ukážeme ako sa využíva integrálny obraz pri výpočte Haarových príznakov. Máme napríklad za úlohu určiť súčet hodnôt pixlov v oblasti *D*. V normálnom obraze by sme museli prejsť všetky body oblasti od bodu 1 po bod 4, ktorá v závislosti na rozmeroch obrazu môže obsahovať stovky až tisíce pixlov. V integrálnom obraze nám na to postačia len 4 body obrazu bez závislosti na rozmeroch. Stačí ak si uvedomíme, že bod



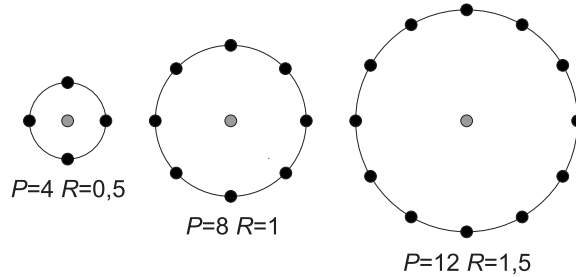
Obrázok 2.5: Integrálny obraz. Prevzaté z [25].

1 obsahuje hodnotu súčtu pixlov v oblasti A , bod 2 hodnotu $A + B$, bod 3 hodnotu $A + C$ a bod 4 hodnotu $A + B + C + D$. Z toho sme schopný vypočítať súčet pixlov v oblasti D ako súčet hodnôt pixlov $4 + 1 - 2 - 3$. Overíme si to dosadením za jednotlivé body: $A + B + C + D + A - A - B - A - C = D$.

Lienhart v práci [11] rozširuje Haarove príznaky o orientácie pootočené o 45° . Zdá sa, že množina obdĺžnikových príznakov poskytuje bohatú reprezentáciu obrazu, ktorá podporuje efektívne učenie. Navyše sú tieto príznaky relatívne rýchlo spočítateľné v konštantnom čase.

2.3 Local Binary Patterns

LBP je zaujímavá kvôli invariantnosti voči zmenám monotónnosti v šedo-tónových obrazoch a nízkej výpočtovej náročnosti [15]. Každý bod obrazu v analyzovanej oblasti je porovnaný so stredovým pixlom a označený buď 1 alebo 0, podľa toho či je jeho hodnota intenzity väčšia alebo menšia ako hodnota stredového pixlu. Analyzovaná oblasť má tvar kruhu s parametrami R (polomer oblasti) a P (počet pixlov), rôzne príklady nastavenia parametrov sú znázornené na obrázku 2.6.



Obrázek 2.6: Rôzne veľkosti LBP, R : polomer, P : počet pixlov. Prevzaté z [15].

Takto dostaneme binárny kruhový príznak pre bod obrazu (x, y) označený:

$$B(x, y) = \langle b_{p-1}, \dots, b_1, b_0 \rangle,$$

kde $b_i \in \{0, 1\}$. Kruhový príznak je často reprezentovaný ako jednorozmerný vektor a následne prevedený na číslo desiatkovej sústavy podľa vzťahu:

$$LBP_{P,R}(x, y) = \sum_{i=0}^{P-1} b_i 2^i. \quad (2.13)$$

Postupne sa získajú LBP príznaky pre celý obraz a výsledne sa reprezentujú formou 1D histogramu.

Existuje osobitný prípad LBP, uniformná LBP. LBP príznak sa nazýva uniformný vtedy a len vtedy ak sa v ňom nachádzajú najviac 2 bitové prechody medzi 0 a 1 respektíve 1 a 0 pri kruhovom prechode. Napríklad 00000000 (0 prechodov), 11100011 (2 prechody) sú uniformné zatiaľ čo 01010000 (4 prechody), 01110101 (6 prechodov) sú neuniformné. Ojala *et al.* [17] zistili, že v obrazoch s textúrou je väčšina LBP príznakov uniformná.

Zhang *et al.* [28] používajú modifikáciu LBP označenú ako MB-LBP (Multi block) na detekciu tvári. V predstavenej metóde LBP sa príznaky určujú pomocou hodnôt jednotlivých pixlov oproti tomu MB-LBP používa priemerné hodnoty pixlov z lokálnych oblastí.

Kapitola 3

Metóda extrakcie príznakov

3.1 Dopredná architektúra

Mnoho štúdií sa zaoberalo skúmaním rozpoznávania objektov v mozgu. Prehľadne to sumarizuje práca [21], z ktorej som čerpal poznatky uvedené v tejto časti.

Hypotéza, že základné spracovanie informácií je dopredné bez spätných väzieb je podporená hlavne krátkymi časovými intervalmi (desiatky milisekúnd) potrebnými na získanie reakcie v bunkách IT. Nie je vylúčené, že existujú lokálne spätné väzby ale je vhodné začať s modelom základnej doprednej architektúry na vysvetlenie okamžitého rozpoznania objektu vo vizuálnej kôre.

Model je rozšírením Hubel a Wiesel [7] simple-to-complex bunkovej hierarchie. Vstupom je šedotónový obrázok, ktorý je najprv analyzovaný multidimenzionálnym poľom S1 jednotiek (simple cells). Tie presne ako mozgové jednoduché bunky najlepšie reagujú na orientované pásy a hrany. S1 jednotky majú viacero orientácií. Ďalšia vrstva je C1, ktorá odpovedá zložitým bunkám (complex cells). Každá jednotka C1 má na vstupe výstup skupiny S1 jednotiek s rovnakou orientáciou ale mierne odlišnou pozíciou a veľkosťou. To má za následok, že jednotky C1 sú necitlivé na pozíciu a veľkosť, čo je charakteristickým znakom mozgových complex buniek.

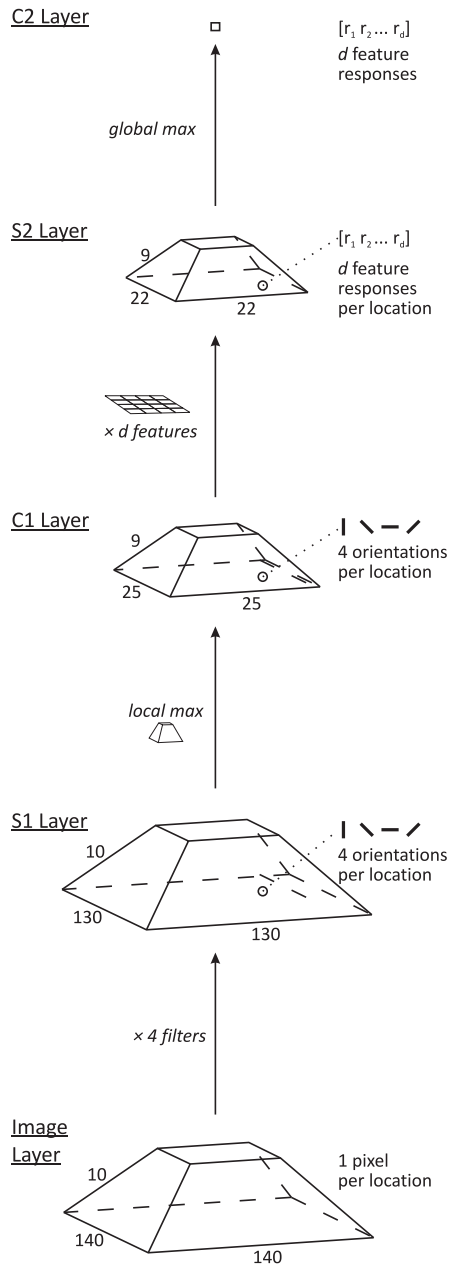
3.2 Popis modelu

Ide o model predstavený pánmi Riesenhuber a Poggio [19]. Tento model je použitý vo viacerých prácach, napríklad [22] či [16]. Podľa [22] patrí model do kategórie hierarchických modelov, inšpirovaných hierarchickou štruktúrou vizuálnej kôry u primátov. Hierarchické modely výkonnosťou prevyšujú holistické (potrebujú súčasne všetky dáta obrazu) a preto je ich obľúbenosť a používanie v poslednej dobe častejšie.

Rozpoznávanie typicky zahrňuje výpočet množiny cieľových črt (features) v prvom kroku a následne ich kombinovanie v kroku druhom. Výhodou metód založených na šablónach je vysoká úspešnosť pri detekcii objektu z jednej kategórie (napr. tváre, autá, chodci) a úspešnosť tréningu s malým počtom vzorov. Nevýhodou týchto metód je, že nedokážu adekvátne zachytiť variácie vzhľadu objektu (nie sú invariantné voči transformáciám objektu) na rozdiel od metód založených na histogramoch (napr. SIFT).

Tento model predstavuje kompromis medzi invariantnosťou a selektivitou čím spája výhody metód založených na šablónach a histogramoch. Invariantnosť je inými slovami nemennosť, všeobecne ide o stálosť javov, veličín voči nejakým zmenám. V rámci strojového

učenia bolo zistené [16], že zvyšovanie riedkosti bazových funkcií má vplyv na zlepšovanie schopnosti generalizácie a v rámci neurovedy je kritické pre učenie biologicky prijateľných modelov z obrazových štatistík pridaním riedkosti obmedzenia. Pri kategorizácii objektov spočíva pridaním riedkosti v použití laterálnej inhibície (vylučuje slabé odozvy, ktoré sú nesúhlasné s lokálne dominantnými odozvami), používanie dominantných orientácií pri získavaní črt a pri záverečnej klasifikácii zahodenie prvkov s nízkymi váhami a použitie len tých, ktoré sú najefektívnejšie.



Obrázek 3.1: Náčrt modelu. Každá vrstva obsahuje 3 rozmery (x, y, veľkosť).

tiež v tejto vrstve dochádza k podvzorkovaniu S1 pyramíd. Výsledkom je C1 vrstva pyramíd

Model pozostáva z 5 vrstiev: prvá je vstupná vrstva obrazu (image layer) a nasledujú 4 vrstvy, každá postavená z predchádzajúcej striedaním porovnávania so šablónou a výberom maxima. Grafické znázornenie modelu je na obrázku 3.1, prevzatý z [16]:

Vstupná vrstva obrazu (image layer). V nej sa obrázok prevedie na šedotónový a upraví sa rozmery obrázku. V tomto modeli sa proporcionálne zmenia rozmery tak, že kratšia strana obrázku bude mať 140 pixlov. Tento spracovaný obrázok slúži na vytvorenie pyramídy (10 obrázkov) postupne zmenšených obrázkov s deliacim faktorom $2^{1/4}$ s použitím bikubickej interpolácie.

S1 vrstva – Gáborové filtre. Vzniká konvolúciou vstupnej vrstvy s 2D Gáborovými filtermi. Dôležitým parametrom je počet orientácií Gáborových filtrov, tento model používa 4 orientácie. Tým sa zvyšuje počet pyramíd z jednej vstupnej na štyri vo vrstve S1. Táto vrstva odpovedá V1 jednoduchým bunkám (simple cells) v mozgu. Gáborové filtre majú rozmer 11×11 pixlov a sú popísané vzťahom:

$$G(x, y) = \exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda} X\right)$$

Kde $X = x \cos \theta - y \sin \theta$ a $Y = x \sin \theta + y \cos \theta$. Následne je jadro Gáborového filtra normalizované. Viac o Gáborových filteroch v kapitole 2.1.

C1 vrstva – Lokálna invarianťnosť. Táto vrstva vyberá z predchádzajúcej lokálne maximá z hodnôt s rovnakou orientáciou. Tým sa dosiahne invarianťnosti voči pozícii a veľkosti (scale). Tak-

(rovnaký počet ako v S1 vrstve) menších rozmerov. Táto vrstva odpovedá V1 zložitým bunkám (complex cells) v mozgu.

S2 vrstva – Prechodná črta. Na každej pozícii obrazu vo všetkých jeho veľkostiach vo vrstve C1 robíme porovnávanie so šablónou (template matching). Ide o porovnávanie medzi oblasťou obrázku z C1 vrstvy a každou z d šablón (d – počet šablón). Tieto šablóny (prototypy) reprezentujú prechodnú črtu modelu.

Šablóny vzniknú náhodným výberom oblastí obrázkov z vrstvy C1 určených na trénovanie. Počas fázy trénovanie sa vyberajú oblasti z obrázkov C1 o veľkosti 4×4 , 8×8 , 12×12 , 16×16 pixlov z náhodnej pozície a náhodnej veľkosti obrázku. Tieto hodnoty sú uložené bokom ako šablóna. Čiže pre oblasť s veľkosťou 4×4 a pre každú orientáciu obsahuje šablóna $4 \times 4 \times 4 = 64$ C1 hodnôt.

Podľa [16] je výhodnejšie používať pri výbere šablón rôzne veľkosti oblastí. Menšie časti obrazu (4×4) môžu byť videné ako kódovanie tvaru zatiaľ čo väčšie oblasti (16×16) sú pravdepodobne vhodnejšie pre textúry (popis povrchovej štruktúry, vzhľadu). Vzhľadom k tomu, že výber šablón je náhodný, mnoho z týchto šablón nepokrýva oblasť záujmu prípadne nie sú vhodné pre klasifikáciu. Nastavenie váh jednotlivých šablón sa ponecháva na daný klasifikátor (napr. SVM), ktorý ja použitý na výslednú kategorizáciu objektov. Je potrebné poznamenať, že výsledná prechodná črta je zložená zo šablón vybraných z rôznych kategórií obrazov.

Po fáze výberu šablón dochádza k výpočtu samotnej vrstvy S2. Šablóny si môžeme predstaviť ako konvolučné filtre, ktorými bude filtrovaná pyramída obrázkov vrstvy C1. Tým nám vznikne d pyramíd (pripomínam, že d je počet šablón) reprezentujúcich odozvy jednotlivých obrázkov na šablóny. S2 vrstva reprezentuje oblasť mozgovej kôry V4 respektíve zadný IT.

Odozva oblasti X z obrazu vrstvy C1 a konkrétnej šablóny P o veľkosti $n \times n$ je daná vzťahom:

$$R(X, P) = \exp\left(-\frac{\|X - P\|^2}{2\sigma^2\alpha}\right). \quad (3.1)$$

Aj X aj P majú rozmery $n \times n \times 4$ (počet orientácií je 4), kde n je z množiny 4, 8, 12, 16. Smerodajná odchýlka je nastavená pri všetkých pokusoch na hodnotu 1. Parameter α je normalizačný faktor pre rôzne veľkosti vyberanej oblasti. Vyjadruje pomer rozmerov P k rozmerom najmenšej oblasti, čo je 4×4 a je daný vzťahom:

$$\alpha = \left(\frac{n}{4}\right)^2. \quad (3.2)$$

C2 vrstva – globálna invariantnosť. Vytvorenie tejto vrstvy je posledným krokom výpočtu modelu a vedie k získaniu vektora črt (feature vector). Ten vznikne výberom jedného globálneho maxima v každej pyramíde obrázkov vrstvy S2 čím prichádzame o informácie o pozícii a veľkosti. Výsledkom je vektor s d hodnotami.

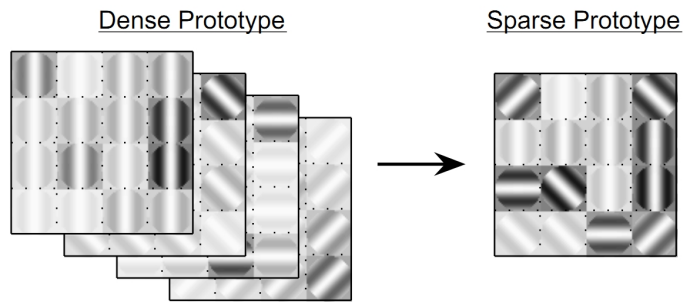
Takto navrhnutý model má viacero parametrov. Najdôležitejšie sú počet orientácii Gáborových filtrov, počet šablón (prototypes), veľkosť šablón (uvažujú sa rozmery 4×4 , 8×8 , 12×12 a 16×16), počet kategórií a počet obrázkov z jednotlivých kategórií použitých na trénovanie. Samotné trénovanie je rozdelené do troch krokov:

1. výber šablón z množiny trénovacích vzorkov,
2. porovnávanie trénovacích vzorov so šablónami (template matching), výpočet príznakov,

3. tréovanie klasifikátora na príznakoch.

Výsledkom tréovania je klasifikátor, ktorý je naučený klasifikovať príznaky (výstupy C2 vrstvy) získane z obrazu pomocou vyššie uvedenej metódy.

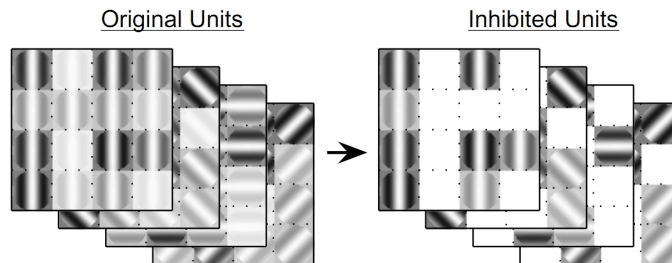
K tomuto základnému modelu je pridané rozšírenie **sparsifikácia S2 vstupov** navrhnuté v [16]. V základnom modeli sa pri výpočte vrstvy S2 používajú všetky orientácie vrstvy C1, ktoré vzniknú použitím Gáborových filtrov. V úvode je naznačený fakt, že výsledný vektor črt by mal byť čo najredší. To vedie k redukcii počtu vstupov vrstvy S2 na jednu orientáciu vrstvy C1. Vo fázy získavania šablón si ukladáme hodnoty len najdominantnejšej orientácie pre každú pozíciu v oblasti $n \times n$. Tým dôjde k redukcii hodnôt šablóny napríklad pre oblasť o veľkosti 4×4 z pôvodných 64 C1 hodnôt na 16 C1 hodnôt. To je znázornené na obrázku 3.2.



Obrázek 3.2: Sparsifikácia S2 vstupov. Prevzaté z [16].

Pri použití tohto rozšírenia základného modelu sa odporúča [16] použiť viac orientácií ako 4, napríklad 12. Podobne ako je tomu aj v mozgu, kde bunky vo vizuálnej časti mozgovej kôry reagujú na viacero orientácií ako len na 4.

Ďalším rozšírením modelu, taktiež navrhnutým v [16], je **inhibícia S1 a C1 výstupov**. Ide o potlačenie nedominantných výstupných hodnôt jednotlivých orientácií z vrstiev S1 a C1. Definujeme globálny parameter h (*úroveň inhibície*), ktorý môže mať hodnoty od 0 do 1 a reprezentuje aká časť výstupného rozsahu sa má potlačiť. Ak je nastavený na 0 nepotlačia sa žiadne hodnoty, ak je nastavený na 1 potlačia sa všetky výstupné hodnoty. V tejto práci bude globálny parameter h konštantne nastavený na hodnotu 0,5. Postupne prechádzame hodnoty obrazu a na každej pozícii nájdeme minimálnu (R_{min}) a maximálnu (R_{max}) odozvu medzi jednotlivými orientáciami. Každý bod obrazu, ktorého hodnota bude menšia ako $R_{min} + h(R_{max} - R_{min})$ bude mať nastavenú odozvu na 0, to ilustruje obrázok 3.3.



Obrázek 3.3: Inhibícia S1 a C1 výstupov. Prevzaté z [16].

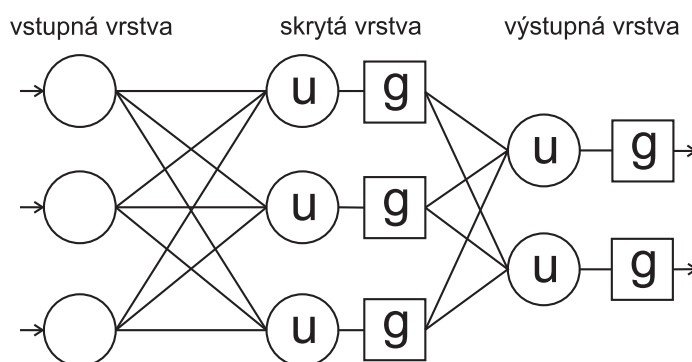
Kapitola 4

Klasifikácia

Základnou myšlienkou klasifikácie je rozpoznať objekt na základe príznakov [8]. Metódy, ktoré sa venujú klasifikácii predpokladajú na vstupe N príznakov získaných z obrazu a následne normalizovaných aby mohli byť reprezentované v rovnakom metrickom priestore. Predstavíme sa tri rozličné metódy klasifikácie: k -NN, Neurónové siete a Support Vector Machines.

Metóda k -najbližších susedov (k -NN [4]) je jednou z najjednoduchších metód v oblasti počítačového učenia. Patrí medzi metódy učenia s učiteľom, kedy sa klasifikujú objekty reprezentované viacdimezióznymi vektormi do dvoch alebo viacerých tried. Objekt je priradený do triedy na základe najkratšej vzdialenosti od svojich k susedov. Používajú sa rôzne metriky na určenie vzdialenosti objektov ako Euklidovská, Hammingova alebo kosínová vzdialenosť. Napríklad ak je $k = 1$, tak je objekt priradený do rovnakej triedy do akej patrí jeho najbližší susedný objekt.

Umelá neurónová sieť (ANN z anglického Artificial Neural Network) je zložená z navzájom prepojených základných jednotiek nazývaných neuróny, ktoré medzi sebou komunikujú pomocou signálov, ukážka na obrázku 4.1. Neuróny majú viacero vstupov a jediný výstup. Vstupy majú pridelené váhy a výstup sa určí ako funkcia váženej sumy vstupov. Rozpoznávanie vzorov je jedna z mnohých aplikácií neurónových sietí. Hlavnou výhodou ANN je schopnosť riešiť úlohy, ktoré iné lineárne systémy nezvládajú [23].



Obrázek 4.1: Neurónová sieť, s označenými funkciami neurónu: u -bázová funkcia, g -výstupná funkcia.

Použitím Lagrangeovej teórie je proces minimalizácie s ohľadom na nerovnosť preformulovaný a tým aj zjednodušený. Lagrangeova funkcia je definovaná nasledovne:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i \omega_i (w_i \cdot x_i + b) + \sum_{i=1}^N \alpha_i, \quad (4.4)$$

kde α_i sú Lagrangeove násobitele. $L(w, b, \alpha)$ je minimalizovaná s ohľadom na w a b , kým α_i je obmedzená na $\alpha_i \geq 0$. Parciálnou deriváciou podľa w a b dostaneme vzťahy:

$$\frac{\delta L(w, b, \alpha)}{\delta w} = w - \sum_{i=1}^N \alpha_i \omega_i x_i, \quad (4.5)$$

$$\frac{\delta L(w, b, \alpha)}{\delta b} = - \sum_{i=1}^N \alpha_i \omega_i. \quad (4.6)$$

Obe vzťahy položíme rovné 0 a dostaneme

$$w = \sum_{i=1}^N \alpha_i \omega_i x_i, \quad (4.7)$$

$$\sum_{i=1}^N \alpha_i \omega_i = 0. \quad (4.8)$$

Tieto vzťahy sú dosadené späť do pôvodnej Lagrangeovej funkcie, kde sa berú do úvahy vzájomné vzťahy po pároch medzi tréningovými vzormi x_i a x_j respektíve ich označenia tried ω_i a ω_j

$$L(w, b, \alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j \omega_i \omega_j (x_i \cdot x_j), \quad (4.9)$$

kde $L(w, b, \alpha)$ je maximalizované s ohľadom na α_i , $\alpha_i \geq 0$. Hoci hodnota b zo vzťahu vypadla, môže byť jednoducho dopočítaná po optimalizácii.

Klasifikácia do dvoch tried vzoru x je uskutočnená podľa nasledujúcej diskriminačnej funkcie:

$$f(x) = w \cdot x + b, \quad (4.10)$$

kde

$$\omega_x = +1 \quad \text{ak} \quad f(x) \geq 0, \quad (4.11)$$

$$= -1 \quad \text{ak} \quad f(x) < 0. \quad (4.12)$$

Diskriminačná funkcia je preformulovaná z hľadiska tréningových vektorov a násobiteľov substitúciou w použitím rovnice 4.7

$$f(x) = \sum_{i \in SV} \alpha_i \omega_i (x_i \cdot x) + b. \quad (4.13)$$

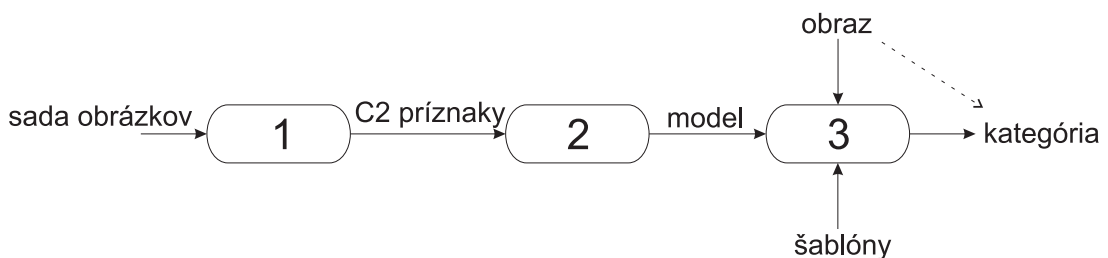
Každý Lagrangeov násobiteľ α_i zdieľa odpovedajúci tréningový vektor x_i . Support vectors majú α_i rôzne od 0. Ostatné tréningové vektory neprispievajú k výslednej diskriminačnej funkcii, preto je suma vo vzťahu 4.13 uskutočnená len na tých hodnotách i , pre ktoré x_i je support vector ($i \in SV$).

Klasifikácia do N tried je dosiahnutá kombináciou N 2-triedových klasifikátorov. Každý z týchto klasifikátorov oddeľuje konkrétnu triedu od zvyšnej množiny. Počas klasifikácie je vzor priradený do triedy s najväčšou pozitívnou vzdialenosťou medzi klasifikovaným vzorom a jednotlivými deliacimi nadrovinami. Problém môže nastať ak jedna z dvoch tried z množiny N tried použitých pri klasifikácii obsahuje podstatne menší počet vzorov ako druhá. Táto nevyrovnanosť môže mať negatívny efekt pri tréovaní support vector klasifikátora, preto je potrebné povoliť oddelenú kontrolu zle klasifikovaných vzorov do dvoch kategórií. Tým sa zabráni ignorovaniu (označeniu ako šum) vzorov z triedy s nízkou kardinalitou.

Kapitola 5

Návrh aplikácie

Cieľom tejto práce je implementovať biologicky inšpirovanú metódu rozpoznávania objektov, ktorá je popísaná v kapitole 3 Metóda extrakcie príznakov. Ide o metódu získavania príznakov z obrazu, ktorá pracuje analogicky ako ľudský mozog v prvých 100ms spracovania vizuálneho signálu. Táto metóda patrí medzi výpočtetne náročnejšie. Samotná metóda zahŕňa len výpočet príznakov z obrazu a o výslednú klasifikáciu sa stará ľubovoľný zvolený klasifikátor. V tejto práci budeme používať na klasifikáciu Support vector machine. Práca sa zameriava na kategorizáciu objektov v statickom obraze (nie vo videu). Aplikáciu je možné rozdeliť do troch hlavných modulov znázornených na obrázku 5.1: Výpočet príznakov (1), Trénovanie SVM (2), Kategorizácia objektu (3) a jeden pomocný modul: Testovanie SVM.



Obrázek 5.1: Schéma modulov aplikácie so znázornenými vstupmi, výstupmi a ich vzájomnou závislosťou.

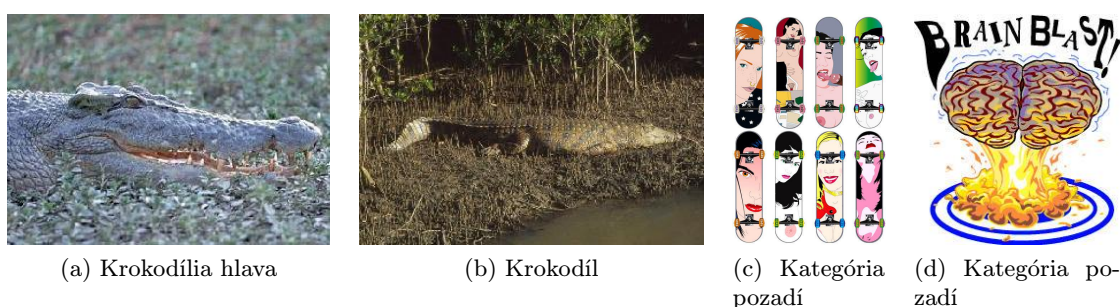
5.1 Databáze obrázkov

Databáza Caltech 101¹ obsahuje 101 kategórií objektov reálneho sveta plus kategóriu pozadí. Obrázky sú farebné, rozlíšenie obrázkov je väčšinou okolo 300×200 px a sú uložené vo formáte jpg. Každá kategória obsahuje od 30 do 800 obrázkov no typicky je to 50 obrázkov na kategóriu, dokopy obsahuje databáza 9 197 obrázkov. Objekty sú v obrázkoch centrovane so žiadnym alebo minimálnym zašumením. Databáza Caltech 101 obsahuje napríklad kategórie akordeón, lietadlo, bonsaj, krokodíl, ľudská tvár a iné. Je to celkom náročná databáza na kategorizáciu, lebo niektoré kategórie sú si dosť podobné. Napríklad stolička a windsorská stolička prípadne krokodíl a krokodíla hlava či divá mačka a leopard.

¹Databáza Caltech 101 je dostupná na adrese http://www.vision.caltech.edu/Image_Datasets/Caltech101/



Obrázek 5.2: Ukážka obrázkov z databáze Caltech 101



Obrázek 5.3: Ukážka obrázkov z databáze Caltech 101

Zámerne sú zvolené dve ukážky obrázkov z databáze Caltech 101. Na obrázku 5.2 sú objekty jasne viditeľné, umiestnené na svetlom pozadí no napriek tomu je klasifikácia náročná pre príklad *a*, *b*. V ňom ide o veľkú podobnosť medzi dvoma rôznymi kategóriami Stolička a Windsorská stolička na rozdiel od príkladov *c*, *d*, kde je podobnosť medzi kategóriami Bonsaj a Mozog nízka.

Druhá ukážka z databáze Caltech 101 na obrázku 5.3 vyberá objekty ešte náročnejšie na klasifikáciu. V prvej dvojici *a*, *b* sú opäť dve podobné kategórie Krokodíl a Krokodília hlava, navyše umiestnené na podklade, s ktorým splývajú. Druhá dvojica *c*, *d* obsahuje ukážky z kategórie pozadí. Táto kategória obsahuje rôzne iné objekty reálneho sveta, útržky textov či ako v prípade *d* deformovaný objekt z inej kategórie.

Ako sa môžete sami presvedčiť, databáza Caltech 101 je náročná databáza určená na rozpoznávanie objektov. Preto aj súčasne najpresnejšie metódy „state of the art” dosahujú na tejto databáze úspešnosť okolo 60%.

Druhou použitou databázou obrázkov je databáza Písmen, obrázok 5.4 obsahuje obrázky z tejto databázy, ktorá vznikla v rámci tejto diplomovej práce. Databáza písmen obsahuje 26 kategórií objektov veľkých písmen bez diakritiky, čiže písmena od A po Z. Každá kategória obsahuje 500 čierno-bielych obrázkov s rozlíšením 128×128 px uložených vo formáte jpg, dohromady má teda databáza 13 000 obrázkov. Objekty (písmená) sú v obraze umiestnené na rôznych pozíciách, majú rôznu veľkosť a sú natočené o uhol v rozmedzí od $+45^\circ$ do -45° voči vzpriamenej pozícii. Tieto vlastnosti sú dôležité, lebo databáza písmen ma za úlohu testovať schopnosť invariantnosti voči pozícii, veľkosti a rotácii objektu v obraze použitej metódy získavania príznakov z obrazu.



Obrázek 5.4: Ukážka obrázkov z databáze Písmen

Databáza Písmen síce obsahuje menej kategórií ako Caltech 101 no celkový počet obrázkov je vyšší. Hlavný rozdiel medzi nimi je v tom, že všetky objekty v databáze Písmen sú umiestnené na bielom pozadí, jasne viditeľné bez rušivých prvkov. Preto pre databázu Písmen očakávam lepšie výsledky ako pre databázu Caltech 101.

5.2 Modul 1: Výpočet príznakov

Výpočet príznakov je najdôležitejšia časť aplikácie, čo sa týka témy práce. Obsahuje predstavený model získavania príznakov z obrazu schematický znázornený na obrázku 5.5. Vstupom modulu je databáza obrázkov, ktoré sú rozdelené do kategórií. Táto databáza predstavuje databázu znalostí, lebo práve kategórie, ktoré sú pokryté v tejto databáze, budú v poslednom kroku klasifikácie cieľové. Táto práca bude používať databázu Caltech 101 a databázu Písmen, ktorá vznikla v rámci diplomovej práce a je jedným z vlastných prínosov tejto práce. Výstupom modulu je množina príznakov reprezentujúcich vstupné obrázky.

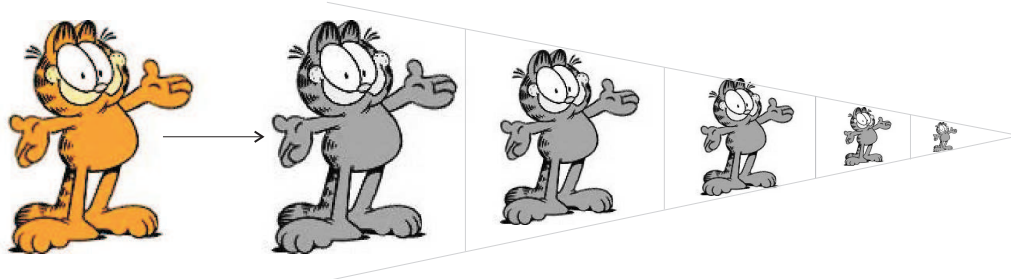


Obrázek 5.5: Schéma modulu 1: Výpočet príznakov.

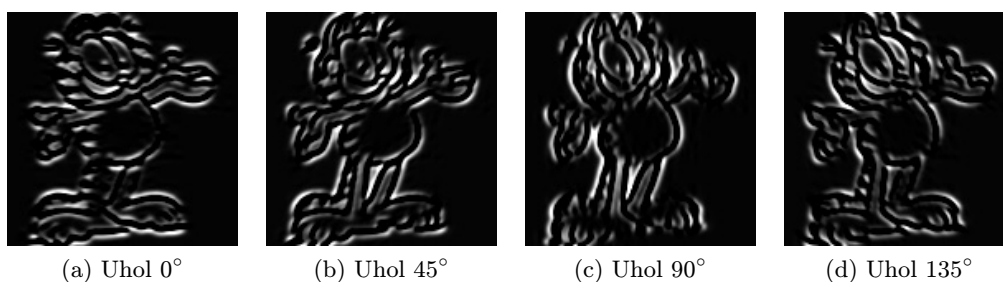
Prvý krok práce modulu Výpočet príznakov je rozdelenie vstupnej databáze na dve disjunktné množiny, trénovaciu a testovaciu množinu. Databáza obrázkov sa na tieto množiny rozdeľuje náhodne. Typicky je to 15, 30 alebo 45 obrázkov z každej kategórie vybraných na trénovanie klasifikátora a zvyšné obrázky v danej kategórii sú určené na testovanie. Z toho vyplýva, že pri učení klasifikátora je každá kategória objektu zastúpená rovnakým počtom obrázkov, no pri testovaní tomu tak byť nemusí.

Každý obrázok z databáze sa prevádza na šedo-tónový a jeho hodnoty sa normalizujú do intervalu $\langle 0, 1 \rangle$. Obrázok sa proporcionálne zmenší tak, aby kratšia strana obrázku mala 140px a z tohto obrázku sa vygeneruje pyramída obrázkov. Pyramída obrázkov je vlastne reprezentácia jedného, toho istého obrázku, súčasne v rôznych rozmeroch pri zachovaní pomeru strán, znázornené na obrázku 5.6. Táto reprezentácia je vhodná v rámci časovej úspory, kedy si dopredu vypočítame všetky potrebné veľkosti obrázku. Pri zmenách rozmerov obrázkov sa používa bikubická interpolácia.

Nasleduje výpočet odzvyv obrazu na Gáborov filter. Všetky vrstvy pyramídy obrázkov sú filtrované Gáborovým filtrom, ktorého jeden z parametrov je orientácia. Výstup po filtrovaní je zobrazený na obrázku 5.7. Pre orientácie 0° a 90° si môžeme všimnúť zvýraznené horizontálne respektíve vertikálne línie. Podľa počtu orientácií nám narastá počet pyramíd



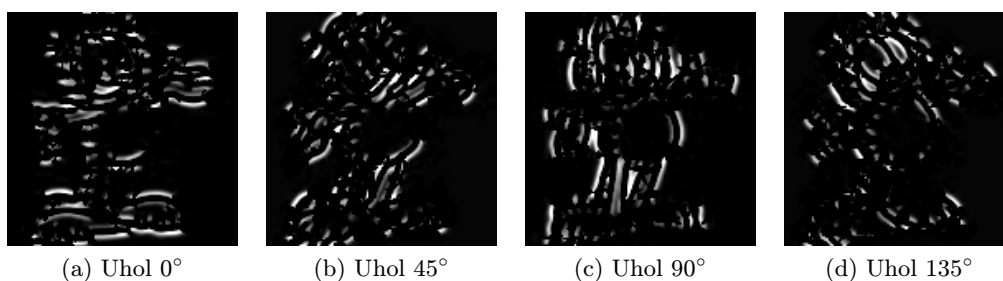
Obrázek 5.6: Ukážka pyramídy obrázkov. Vertikálnymi čiarami sú naznačené vrstvy pyramídy.



Obrázek 5.7: Ukážka filtrovania obrázku Gáborovou funkciou s rôznymi orientáciami.

pre každý obrázok. To znamená, že ak máme 4 orientácie Gábrového filtra, tak pre jeden vstupný obrázok dostaneme 4 pyramídy obrázkov (0° , 45° , 90° , 135°). Hodnoty obrázku po filtrácii sú ešte potlačené podľa vypočítaného prahu (viac v kapitole 3 Metóda extrakcie príznakov), dochádza k inhibícii hodnôt. Vplyv inhibície znázorňuje obrázok 5.8. Táto fáza odpovedá S1 vrstve z popisu modelu.

Ďalším krokom je výber lokálnych maxim z pyramíd obrázkov. Lokálne maximá sa vyberajú z dvoch po sebe idúcich vrstiev v pyramíde obrázkov, tým dochádza k zníženiu výšky pyramídy (počtu vrstiev v pyramíde) o 1. Veľkosť okolia pre výber maxima sa znižuje v rovnakom pomere ako veľkosť obrázkov v pyramíde. Rovnako ako v predchádzajúcom kroku sú potlačené hodnoty ležiace pod novo vypočítaným prahom (inhibícia). Zároveň v tejto časti výpočtu dochádza k podvzorkovaniu, čiže obrázky v pyramíde budú mať menšie rozmery. Táto fáza odpovedá C1 vrstve z popisu modelu.



Obrázek 5.8: Ukážka filtrovania obrázku Gáborovou funkciou s rôznymi orientáciami po inhibícii. Úroveň inhibície je pre názornosť nastavená na $h = 0,9$.

Z takto pripravených dát sa vyberajú šablóny použité pre výpočet porovnávania so šablónami (template matching). Šablóny sú malé časti obrázkov vyberané náhodne. To znamená, že si náhodne určíme obrázok (momentálne reprezentovaný viacerým počtom pyramíd podľa počtu orientácií), vyberieme vrstvu z pyramídy (rôzne rozmery) a z daného obrázku vyberieme malú časť (patch). Šablóny sú potrebné v module 3 Kategorizácia objektu, preto sú uložené na ďalšie použitie. Ukladajú sa v tzv. sparsifikovanej forme. Zjednodušene ide o redukcii hodnôt, kedy si miesto viacerých šablón pre jeden obrázok zložený z viacerých orientácií ukladáme len jednu šablónu, ktorá obsahuje vybrané hodnoty zo všetkých orientácií. Viac v popise modelu v kapitole 3.

Po časti výberu šablón nasleduje samotné porovnávanie so šablónami. Každý obrázok na vstupe je porovnávaný s každou šablónou. Pripomínam, že obrázky sú v tejto fáze reprezentované ako pyramídy obrázkov a navyše každá orientácia Gáborového filtra znamená pridanie pyramídy obrázkov. Výsledkom porovnávania je hodnota ako sú si daný obrázok a šablóna podobné. Výsledkom tejto časti je počet pyramíd zhodný s počtom šablón. Výber šablón a porovnávanie so šablónami odpovedá S2 vrstve v popise modelu.

Posledným krokom pri výpočte príznakov obrazu je výber jedného globálneho maxima pre každé porovnanie obrázka a šablóny. To znamená, že z každej pyramídy z S2 vrstvy sa vyberie jedno globálne maximum. Výsledkom je, že pre jeden vstupný obrázok dostaneme taký počet príznakov, aký je počet šablón. Táto fáza odpovedá C2 vrstve z popisu modelu, preto sa aj tento vektor príznakov označuje ako vektor C2 príznakov.

Zmenou v tejto práci oproti pôvodnej metóde je použitie normalizácie výsledného vektora. Ide o klasický výpočet euklidovskej dĺžky vektora $\|v\|$ a následne rozdelenie každej položky vektora touto dĺžkou. Túto optimalizáciu volíme kvôli zvolenému klasifikátoru SVM.

5.3 Modul 2: Trénovanie SVM

Modul 2 môžeme všeobecne považovať za trénovanie ľubovoľného klasifikátora, v našom prípade ide o trénovanie SVM klasifikátora. Vstupom modulu je množina C2 príznakov a výstupom modulu je natrénovaný model SVM, ilustrované na obrázku 5.9.



Obrázek 5.9: Schéma modulu 2: Trénovanie SVM.

Typ zvolenej SVM je multi-class SVM konkrétne C-SVC, tá dokáže v jednom behu určiť aký objekt sa nachádza v obraze z množiny známych objektov. Inou možnou voľbou je množina one-class SVM, tie sa používajú na rozpoznanie všetkých známych objektov v jednom obraze (multi-labeling).

Výhodou SVM oproti iným klasifikátorom je jednoduchosť použitia, jednoduché vytvorenie, trénovanie a testovanie. Človek aj bez väčších znalostí SVM dokáže vytvoriť úspešný klasifikátor. Nevýhodou je nutnosť definovať parametre SVM, ktoré majú veľký vplyv na dosiahnutý výsledok klasifikácie.

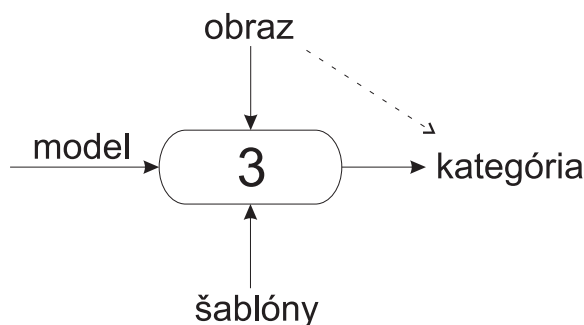
V tejto práci budeme používať typ SVM C-SVC s jadrom Gaussovská RBF (radial basis function), ktorá je definovaná ako $e^{-(\gamma\|u-v\|)^2}$. Parametre takto zvolenej SVM sú C (cost, parameter trestu) a γ (polomer Gaussovského jadra). Tu sa objavuje už spomínaná nevýhoda SVM, lebo mi dopredu nevieme, ako zvoliť nastavenie parametrov C a γ tak, aby

klasifikátor dosahoval čo najvyššiu presnosť. Preto musíme urobiť hľadanie parametrov experimentálne. To sa robí pomocou cross-validácie, kedy sa tréningová množina rozdelí na n rovnakých podmnožín. SVM je natrénované na $n - 1$ podmnožinách a testované na zvyšnej jednej podmnožine dát. Pri cross-validácii sa postupne skúšajú jednotlivé kombinácie parametrov a uchováva sa najlepší výsledok. Hľadanie optimálnych parametrov je časovo náročné, lebo rôznych vzájomných kombinácií C a γ je mnoho.

Po nájdení parametrov SVM sa vytvorí a na tréningovej množine naučí model klasifikátora. Model SVM sa následne testuje pomocou modulu Testovanie SVM. Vstupom modulu je model a množina C2 príznakov získaná z testovacej množiny obrázkov. Výstupom je úspešnosť klasifikácie vytvoreného modelu. Po tomto kroku sa model SVM uloží pre použitie v ďalšom module, module Kategorizácia objektu.

5.4 Modul 3: Kategorizácia objektu

Posledný modul predstavuje praktické použitie aplikácie. Má tri vstupy schematicky znázornené na obrázku 5.10. Prvým vstupom modulu je obrázok. Druhým vstupom je natrénovaný model SVM klasifikátora. A tretím vstupom sú šablóny vybrané pri fáze tréningovania SVM v module 1. Opakujem, že šablóny nám slúžia na výpočet C2 príznakov vstupného obrazu, ktoré sú následne predložené na klasifikáciu pomocou SVM. Výstupom modulu je určenie kategórie objektu, ktorý sa nachádza vo vstupnom obrázku.



Obrázek 5.10: Schéma modulu 3: Kategorizácia objektu.

Modul 3 je navrhnutý na kategorizáciu objektu v jednom obraze. Ak potrebujeme kategorizovať hromadne väčší počet obrázkov navrhujem použitie modulu 2 s modifikáciou modulu 1. Pred použitím modulu 2 na hromadnú kategorizáciu je potrebné zrušiť automatické rozdeľovanie vstupných dát na tréningovú a testovaciu množinu v module 1 a pridať namiesto jedného vstupu pre databázu obrázkov dve osobitné vstupy pre obrázky určené na tréningovanie a obrázky určené na kategorizáciu. Ako hromadný klasifikátor nám poslúži pomocný modul Testovanie SVM, ktorého výstupom okrem úspešnosti klasifikácie je aj zoznam priradených obrázkov do jednotlivých kategórií.

Kapitola 6

Implementácia

Na implementáciu navrhnutej aplikácie bol zvolený jazyk C++ s objektovo orientovaným návrhom. Aplikácia bola vyvíjaná a odladená súbežne na operačnom systéme Microsoft Windows 7 Professional so 64 bitovou architektúrou a Linuxovej distribúcii Ubuntu 10.04.2 LTS taktiež so 64 bitovou architektúrou. Kód je prenositeľný, čo bolo jedným z bodov zadania diplomovej práce.

Aplikácia používa tri knižnice. *Open Source Computer Vision* (OpenCV)¹ na prácu s obrazovými dátami ako je redukcia farebného priestoru, normalizácia hodnôt a konvolúcia obrazu so zadaným konvolučným jadrom. *CvGabor*² na výpočet konvolučného jadra pomocou Gáborovej funkcie a *LIBSVM*³ na prácu s SVM klasifikátorom. Všetky použité knižnice sú voľne dostupné. Podstatne nám uľahčili prácu a dovolili sa nám sústrediť na extrakciu príznakov z obrazu, ktorá je najdôležitejšou časťou tejto práce.

Veľkou výhodou knižnice *LIBSVM* je, že sa dá používať ako hotový software. To znamená, že po skompilovaní máme dostupné binárne súbory, ktoré môžeme používať na trénovanie SVM, testovanie SVM a klasifikáciu pomocou vytvoreného modelu SVM. Modul 2 Trénovanie SVM používa binárny súbor *svm-train* a Modul 3 Kategorizácia objektu ako aj pomocný modul Testovanie SVM používajú binárny súbor *svm-predict*.

V predchádzajúcej kapitole sme spomínali komplikované hľadanie optimálnych parametrov SVM klasifikátora. Tie sú potrebné pri spúšťaní *svm-train* ako argumenty programu. Hľadanie týchto parametrov pomocou metódy cross-validácie poskytuje skript *grid.py*, ktorý je súčasťou knižnice *LIBSVM*.

Dôležité je spomenúť, že všetky náhodne generované čísla sú implementované pomocou vlastného generátora pseudonáhodných čísel (*RNG.h*). Tým dosiahneme možnosť opakovať pokusy s rovnakým nastavením pre rovnaké dáta, napríklad pre porovnanie úspešnosti implementovaných zmien.

Ako vyplýva z návrhu aplikácie, systém tvoria 3 hlavné a 1 pomocný modul. Aplikácie nemá grafické užívateľské rozhranie a je ovládaná z konzoly.

¹OpenCV je dostupná na adrese <http://opencv.willowgarage.com/wiki/>

²CvGabor je dostupná na adrese <http://code.google.com/p/cvgabor/>

³LIBSVM je dostupná na adrese <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

6.1 Moduly aplikácie

Modul 1: Výpočet príznačkov

Modul 1 implementuje rozdelenie vstupnej databázy obrázkov na tréningovú a testovaciu množinu a následne výpočet C2 príznačkov oboch množín. Spúšťa sa pomocou skriptu *C2features.sh* a reaguje práve na 2 parametre, ktorých poradie je pevne dané:

1. súbor so zoznamom všetkých obrázkov
2. konfiguračný súbor

Príklad korektného spustenia:

```
C2features.sh images.txt config.txt
```

Obrázky databázy očakáva v zložke *images* s rovnakým umiestnením so skriptom *C2features.sh*. Formát súborov *images.txt* a *config.txt* je uvedený v prílohe B. Výstupom modulu sú dva súbory v textovom formáte *Train.txt* a *Test.txt*. Tieto súbory na jednotlivých riadkoch obsahujú označenie kategórie obrazu nasledované vektorom príznačkov pre daný obraz. Taktiež ukladá textový súbor *Prototypes.txt*, ktorý obsahuje vybrané šablóny uložené v sparsifikovanej forme, a ktorý potrebuje modul 3 na výpočet C2 príznačkov klasifikovaného objektu.

Modul 2: Tréning SVM

Modul 2 implementuje vytvorenie, tréning SVM a uloženie výsledného modelu SVM. Je v ňom zahrnuté vyhľadávanie parametrov klasifikátora C a γ a navyše je spojený s pomocným modulom Testovanie SVM, takže výsledkom je model klasifikátora aj s jeho úspešnosťou dosiahnutou na testovacích dátach.

Spúšťa sa pomocou skriptu *trainSVM.py* bez parametrov, no z lokálneho priečinka používa súbory *Train.txt*, *Test.txt*, *grid.py*, *svm-train*, *svm-predict* a *testparams.sh*.

Príklad korektného spustenia:

```
trainSVM.py
```

Skript *testparams.sh* porovnáva presnosť klasifikátora natrénovaného s parametrami nájdenými pomocou *grid.py* s klasifikátormi natrénovanými s parametrami zvolenými experimentálne. To je jedným z ďalších prínosov práce. Meraním sme zistili, že *grid.py* má problém nájsť optimálne parametre, ak je tréningová množina príliš malá (15 obrázkov na kategóriu).

Modul 3: Kategorizácia objektu

Posledným modulom implementuje použitie modelu klasifikátora SVM na kategorizáciu. Spúšťa sa pomocou skriptu *categorize.sh* a reaguje práve na 3 parametre, ktorých poradie je pevne dané:

1. obrázok určený na klasifikáciu
2. konfiguračný súbor
3. model SVM

Príklad korektného spustenia:

```
classify.sh image.jpg config.txt model
```

Nesmieme zabudnúť, že v lokálnom priečinku musí byť prítomný súbor *Prototypes.txt* vytvorený pomocou modulu 1. Výstupom je celé číslo, ktoré slúži ako index kategórie.

Kapitola 7

Experimentálne výsledky

Táto kapitola je podľa môjho názoru najzaujímavejšou kapitolou z celej práce. Prezentuje uskutočnené experimenty. Ich popis, realizáciu a dosiahnuté výsledky. Dozvieme sa v nej výsledky hľadania optimálnych parametrov SVM, závislosť úspešnosti kategorizácie na počte orientácií Gáborových filtrov pri výpočte príznakov obrazu, závislosť úspešnosti kategorizácie na počte šablón a záverom porovnáme dosiahnuté výsledky s inými metódami. To všetko nájdete v tejto časti prehľadne reprezentovaná tabuľkami a grafmi.

Všetky prezentované experimenty sú veľmi časovo náročné. Prebiehali na školskom serveri *Medusa*, na ktorom beží linuxová distribúcia Ubuntu 10.04.2 LTS so 64 bitovou architektúrou a parametre servera sú: 8 procesorov, každý typu Intel Xeon X3470 2,93 GHz so 4 jadrami a celkovo s 8 GB RAM pamäte.

Nasledujúca tabuľka 7.1 obsahuje ukážkové hodnoty dĺžky behu jednotlivých fázy pre najkratšie a najdlhšie prípady. Merania prebehli na databáze Caltech 101 bez kategórie pozadí, všetky pre 4 orientácie Gáborových filtrov. Výpočet C2 príznakov je mnohonásobne dlhší ako tréovanie a testovanie SVM dohromady.

Nastavenie modelu	Výpočet C2 príznakov	Tréovanie SVM	Testovanie SVM
15 obrázkov, 100 šablón	73min	1,823s	4,315s
30 obrázkov, 100 šablón	75min	3,145s	7,937s
15 obrázkov, 5000 šablón	3025min	40,388s	2min 21,550s
30 obrázkov, 5000 šablón	3083min	2min 2,936s	3min 28,114s

Tabuľka 7.1: Časová náročnosť výpočtu C2 príznakov pre tréováciu a testováciu množinu, tréovania a testovania SVM.

Tabuľka 7.1 neobsahuje čas potrebný na nájdenie optimálnych parametrov SVM, ktoré je taktiež časovo náročné a závisí na viacerých parametroch ako je napríklad rozsah intervalu, v ktorom sa hľadajú hodnoty parametrov.

7.1 Hľadanie optimálnych parametrov SVM

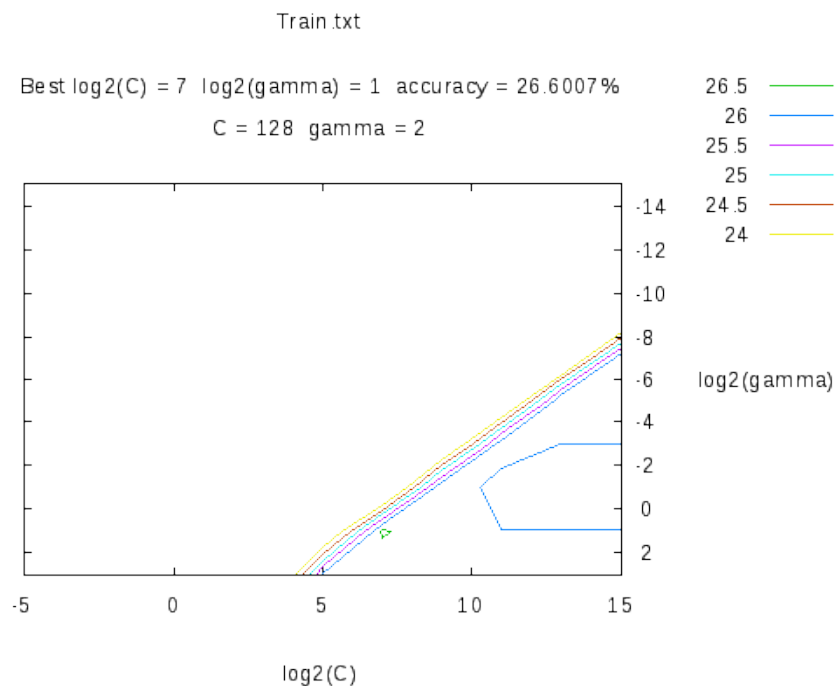
Nájdenie optimálnych parametrov SVM je dôležitý krok k získaniu čo najlepšieho klasifikačného modelu. Poslúžil nám nato skript *grid.py*, ktorý je súčasťou knižnice LIBSVM. Budeme hľadať dva parametre: C a γ . Vstupom skriptu je množina C2 príznakov, ktorú si musíme vypočítať pred samotným hľadaním parametrov.

Popis experimentu:

- zvolíme si vstupné dáta, množinu príznakov $C2$,
- zvolíme si na koľko častí sa má deliť trénovacia množina $C2$ príznakov pri cross-validácii,
- zvolíme si interval, v ktorom sa budú hľadať hodnoty C a γ ,
- opakujeme výpočet pre rôzne nastavenia a rôzne vstupné dáta,
- výsledky prezentujeme v podobe grafu.

Takto popísaný experiment sme uskutočnili pre všetky merania, popísané v ďalších častiach tejto kapitoly. Teraz si pre prehľadnosť a jednoduchosť ukážeme výstupy z 2 meraní. Obe merania boli uskutočnené na databáze Caltech 101 bez kategórie pozadí. Tá bola vylúčená z merania na základe odporúčania vedúceho práce.

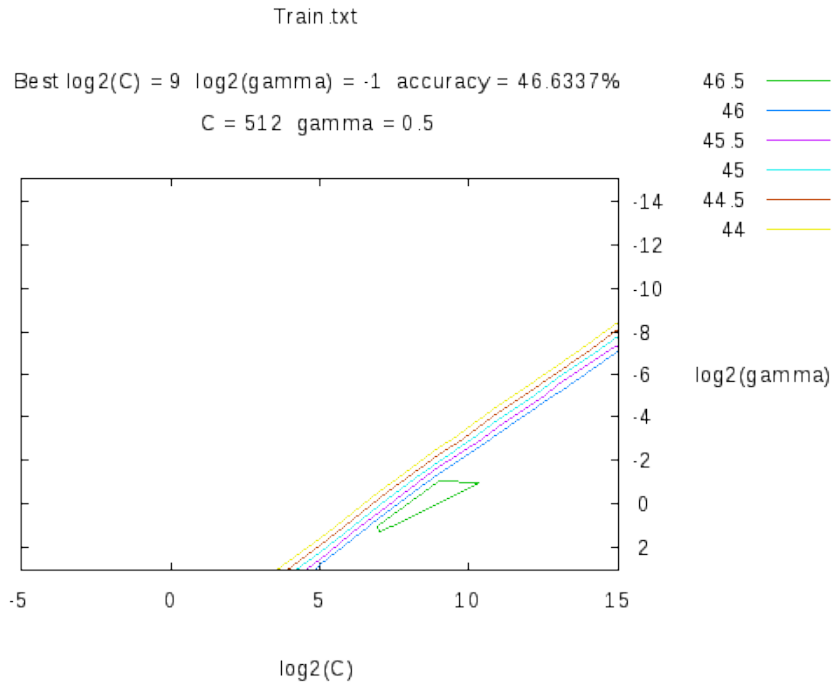
Na prvom obrázku 7.1 je graf znázorňujúci hľadanie optimálnych parametrov C a γ pre prípad: 15 trénovacích obrázkov na kategóriu, 4 orientácie, 500 šablón, 5 častí pri cross-validácii, interval pre hľadanie parametru $C < -5, 15 >$ s krokom 2 a pre parameter $\gamma < -15, 3 >$ taktiež s krokom 2. Krok a hodnoty intervalu sú hodnoty logaritmu pri základe 2, presne tak je to definované v skripte *grid.py*.



Obrázek 7.1: Hľadanie optimálnych parametrov. Dosiagnutá úspešnosť pri cross-validácii 26.6007%

Druhý obrázok 7.2 obsahuje graf hľadania optimálnych parametrov pre prípad 30 trénovacích obrázkov na kategóriu, 16 orientácií, 500 šablón a rovnaký počet častí pre cross-validáciu a intervaly ako predchádzajúci graf.

Ide o hrubé hľadanie parametrov s relatívne veľkým krokom no napriek tomu je to časovo náročné. Do budúca by bolo vhodné vyskúšať po hrubom hľadaní parametrov zúžiť



Obrázek 7.2: Hľadanie optimálnych parametrov. Dosiagnutá úspešnosť pri cross-validácii 46.6337%

intervaly oboch parametrov a zjemniť krok pri hľadaní optimálnych parametrov a spustiť vyhľadávanie parametrov znova.

Nasledujúca tabuľka 7.2 obsahuje porovnanie úspešnosti kategorizácie (pre obrázky 7.1 a 7.2) na testovacej množine s použitím parametrov nájdených pomocou skriptu *grid.py* a s použitím parametrov určených experimentálne. Potvrdila sa domnienka z predchádzajúcej kapitoly, kedy hľadanie parametrov na malej množine dát (15 respektíve 30 tréningových obrázkov na kategóriu) je problematické pre určenie optimálnych parametrov. Pre prípad 15 tréningových obrázkov si taktiež môžeme všimnúť, že úspešnosť pri cross-validácii je podstatne nižšia ako pri kategorizácii na testovacej množine. V oboch ukázkových prípadoch nenašiel skript *grid.py* optimálne parametre SVM.

Počet tréningových obrázkov/kat.	grid.py	Experimentálne určené parametre		
		c 32,0 γ 8,0	c 128,0 γ 8,0	c 128,0 γ 2,0
15 obrázkov	c 128,0 γ 2,0	39,7515	39,9190	39,4443
	39,4443			
30 obrázkov	c 512,0 γ 0,5	c 32,0 γ 8,0	c 128,0 γ 8,0	c 128,0 γ 2,0
	49,6724	50,5047	50,3984	50,3276

Tabuľka 7.2: Úspešnosť kategorizácie v % pre rôzne nastavenia parametrov SVM.

7.2 Závislosť úspešnosti na počte orientácií

Jedným z hlavných parametrov prezentovanej metódy získavania príznakov je počet orientácií použitých Gáborových filtrov. Cieľom tohto merania je experimentálne zistiť pre aký počet orientácií dosahuje model najvyššiu úspešnosť kategorizácie. Podľa [21] by mala úspešnosť s narastajúcim počtom orientácií stúpať.

Popis experimentu:

- vypočítame 4 sady trénovacích a testovacích množín C2 príznakov pre 4 rôzne počty orientácií: 4, 8, 12 a 16,
- počet šablón nastavíme na 500, veľkosti šablón náhodne od 4×4 po 16×16 ,
- pre každú sadu nájdeme optimálne parametre SVM,
- natrénujeme 4 modely SVM,
- každý model otestujeme a úspešnosť zaznamenáme,
- namerané hodnoty porovnáme v grafe.

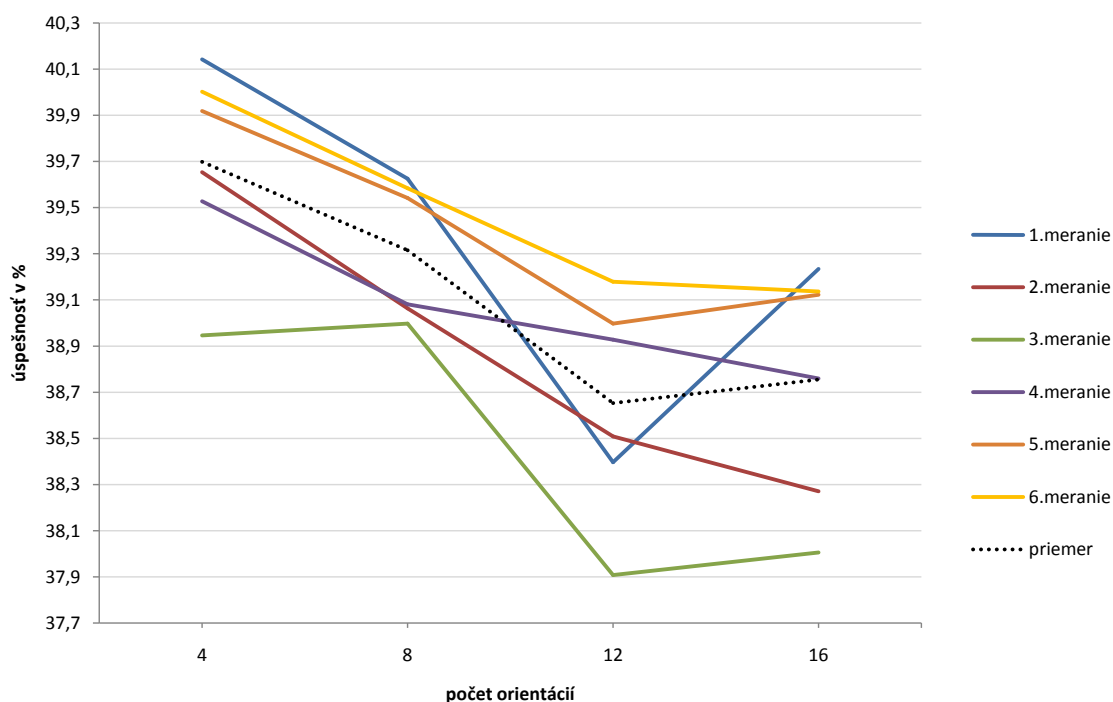
Prvý experiment sme uskutočnili pre 15 trénovacích obrázkov na kategóriu na databáze Caltech 101 bez kategórie pozadí. Už po prvom meraní sme dosiahli hodnoty v rozpore s našimi očakávaniami. Zvyšovaním počtom orientácií nám úspešnosť nenarastala, práve naopak. Preto sme tento experiment opakovali 6 krát pre vylúčenie náhodnej chyby no výsledok sa nezmenil. Momentálne pre 15 trénovacích obrázkov dosahujeme maximálnu úspešnosť na databáze Caltech 101 pre 4 orientácie. Namerané hodnoty obsahuje tabuľka 7.3.

Meranie	Počet orientácií			
	4	8	12	16
1.	40,1424	39,6258	38,3971	39,2349
2.	39,6537	39,0637	38,5088	38,2714
3.	38,9461	38,9975	37,9084	38,0061
4.	39,5281	39,0813	38,9277	38,7601
5.	39,9190	39,5420	38,9975	39,1231
6.	40,0028	39,5839	39,1790	39,1371
Priemer	39,6987	39,3157	38,6531	38,7555

Tabuľka 7.3: Závislosť úspešnosti na počte orientácií. Namerané hodnoty pre 15 trénovacích obrázkov na kategóriu.

Obrázok 7.3 obsahuje graf znázorňujúci namerané hodnoty spolu s priemernou hodnotou šiestich meraní. Takmer všetky merania mali rovnaký priebeh. Pre 4 orientácie dosiahli maximálnu úspešnosť, pre 8 orientácií úspešnosť klesla, pre 12 orientácií úspešnosť pokračovala v klesaní a až pre 16 orientácií sa úspešnosť zlepšila no iba mierne.

Druhý experiment sme realizovali s rovnakým nastavením ako pri prvom experimente. Jediný rozdiel bol v počte trénovacích obrázkov, kde sme použili 30 trénovacích obrázkov na kategóriu na databáze Caltech 101 bez kategórie pozadí. Namerané hodnoty obsahuje tabuľka 7.4.



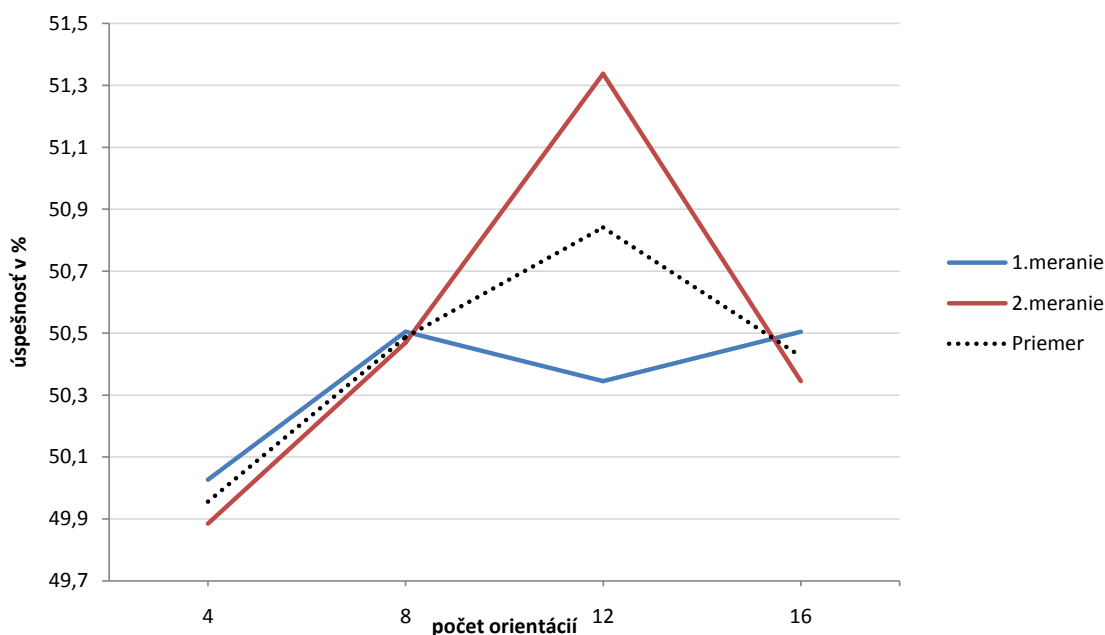
Obrázek 7.3: Závislosť úspešnosti na počte orientácií. Priebeh pre 15 trénovacích obrázkov na kategóriu.

Meranie	Počet orientácií			
	4	8	12	16
1.	50,0266	50,5047	50,3453	50,5047
2.	49,8849	50,4693	51,3370	50,3453
Priemer	49,9556	50,4870	50,8412	50,4250

Tabulka 7.4: Závislosť úspešnosti na počte orientácií. Namerané hodnoty pre 30 trénovacích obrázkov na kategóriu.

Tento experiment sme opakovali dva krát a namerali sme hodnoty, ktoré sa podstatne líšia od hodnôt dosiahnutých v prvom experimente. Vývoj úspešnosti prezentuje graf na obrázku 7.4. Vývoj úspešnosti začína podľa očakávaní a pre 4 orientácie je úspešnosť kategorizácie najnižšia, pre 8 orientácií úspešnosť stúpa, pre 12 orientácií priemerná hodnota úspešnosti opäť stúpa no pre 16 orientácií klesne pre oba prípady na úroveň zhruba odpovedajúcu úspešnosti pre 8 orientácií.

Z experimentov vyplýva, že počet trénovacích obrázkov má vplyv na počet orientácií, pri ktorom metóda získavania príznakov dosahuje najvyššie úspešnosti klasifikácie. Pre 15 trénovacích obrázkov na kategóriu je úspešnosť klasifikácie najvyššia pre 4 orientácie a pre 30 trénovacích obrázkov na kategóriu je úspešnosť klasifikácie najvyššia pre 12 orientácií.



Obrázek 7.4: Závislosť úspešnosti na počte orientácií. Priebeh pre 30 trénovacích obrázkov na kategóriu.

7.3 Závislosť úspešnosti na počte šablón

Ďalším z hlavných parametrov metódy získavania príznačov z obrazu je počet šablón, ktorý zároveň určuje aj počet príznačov reprezentujúcich jeden obrázok. Cieľom tohto merania je experimentálne zistiť, pre aký počet šablón dosahuje model najvyššiu úspešnosť kategorizácie. Očakávame priamu úmernosť medzi počtom šablón a úspešnosťou kategorizácie, čiže s väčším počtom šablón bude úspešnosť kategorizácie narastať. Do tohto merania prvý krát zapojíme aj našu vlastnú databázu Písmen, na ktorej očakávame lepšie výsledky ako na databáze Caltech 101.

Popis experimentu:

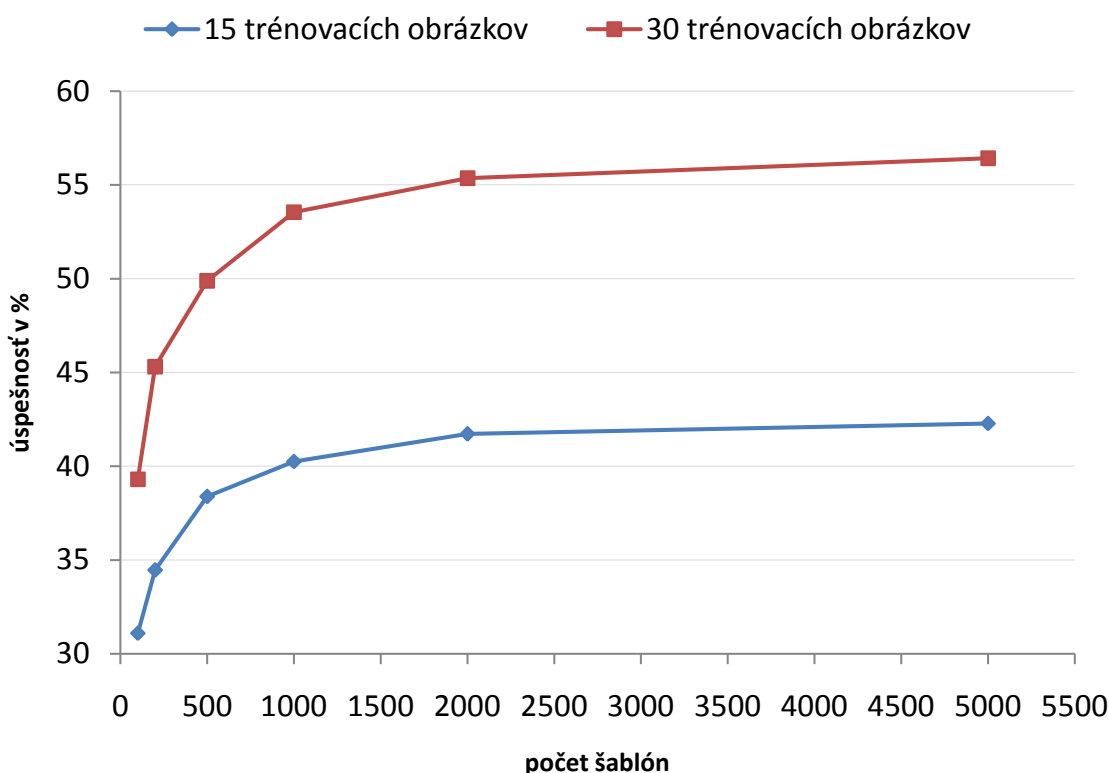
- vypočítame 6 sad trénovacích a testovacích množín C2 príznačov pre 6 rôznych počtov šablón: 100, 200, 500, 1000, 2000 a 5000. Veľkosť šablón náhodne od 4×4 po 16×16 ,
- počet orientácií nastavíme na 4,
- pre každú sadu nájdeme optimálne parametre SVM,
- natrénujeme 6 modelov SVM,
- každý model otestujeme a úspešnosť zaznamenáme,
- namerané hodnoty porovnáme v grafe.

Na databáze Caltech 101 sme uskutočnili dva experimenty pre 15 a pre 30 trénovacích obrázkov na kategóriu. Z databáze bola vylúčená kategória pozadí na základe odporúčania vedúceho práce. Merania pre oba experimenty prebehli jeden krát. Namerané hodnoty obsahuje tabuľka 7.5.

Počet trénovacích obrázkov/kat.	100	200	500	1000	2000	5000
15 obrázkov	31,0947	34,4596	38,3831	40,2541	41,7202	42,2787
30 obrázkov	39,3129	45,3161	49,8849	53,5506	55,3568	56,4193

Tabuľka 7.5: Závislosť úspešnosti kategorizácie na počte šablón. Namerané pre databázu Caltech 101 bez kategórie pozadí.

Namerané hodnoty naplnili naše predpoklady a s vyšším počtom šablón narastá úspešnosť kategorizácie, čo znázorňuje obrázok 7.5. To znamená, že čím viac šablón použijem, tým dosiahnem väčšiu presnosť kategorizácie no na druhú stranu sa zvýši celkový čas potrebný na výpočet C2 príznakov.



Obrázok 7.5: Závislosť úspešnosti kategorizácie na počte šablón. Priebeh pre databázu Caltech 101 bez kategórie pozadí.

V grafe na obrázku 7.5 si môžeme všimnúť, že od počtu šablón 2000 sa pridaním viac ako dvojnásobného počtu šablón zvýši úspešnosť klasifikácie len minimálne. No čas potrebný na kategorizáciu, najmä na výpočet C2 príznakov, sa zdvojnásobí. Preto v praktickom nasadení metódy nemá význam používať čo najvyšší počet šablón, hoci bude úspešnosť vysoká, čas potrebný na výpočet bude taktiež veľký.

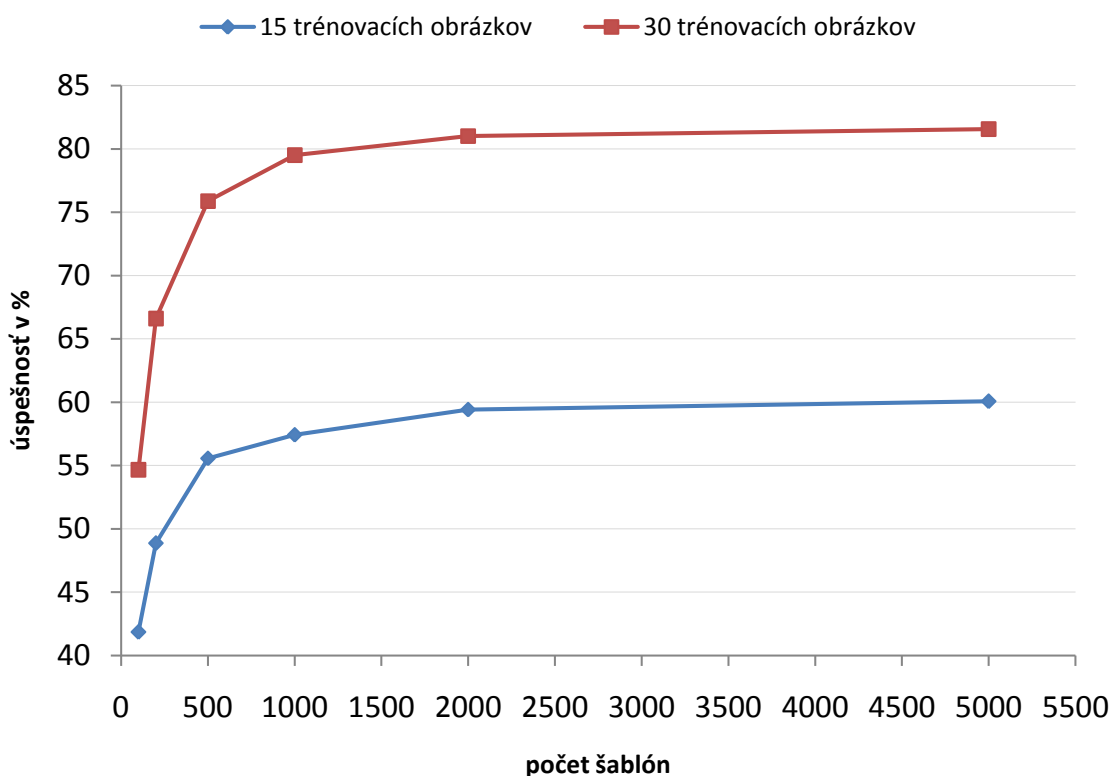
Podobne ako na databáze Caltech 101 sme uskutočnili dva experimenty aj na databáze Písmen pre prípad 15 a 30 trénovacích obrázkov na kategóriu. Merania pre oba experimenty prebehli jedenkrát a namerané hodnoty obsahuje tabuľka 7.6.

Opäť sa naplnilo naše očakávanie a medzi počtom šablón a úspešnosťou kategorizácie

Počet trérovacích obrázkov/kat.	100	200	500	1000	2000	5000
15 obrázkov	41,8557	48,8660	55,5670	57,4306	59,4211	60,0714
30 obrázkov	54,6563	66,6039	75,8838	79,5008	81,0229	81,5548

Tabulka 7.6: Závislosť úspešnosti kategorizácie na počte šablón. Namerané pre databázu Písmen.

platí priama úmera. Graf tejto závislosti je znázornený na obrázku 7.6. Toto meranie tak tiež potvrdilo naše očakávanie, že úspešnosť kategorizácie na databáze Písmen bude vyššia ako na databáze Caltech 101.



Obrázek 7.6: Závislosť úspešnosti kategorizácie na počte šablón. Priebek pre databázu Písmen.

Tvar krivky závislosti medzi počtom šablón a úspešnosťou kategorizácie je pre všetky meranie v tejto podkapitole (obrázky 7.5 a 7.6) rovnaký a počet trérovacích obrázkov nemal žiaden vplyv na jej priebeh. Od počtu šablón 100 po 500 dochádza k prudkému zvyšovaniu úspešnosti no s pribúdajúcimi šablónami sa toto zvyšovanie spomaľuje. Stále však platí, že každé zvýšenie počtu šablón zvýšilo úspešnosť kategorizácie na úkor časovej náročnosti.

7.4 Dosiahnuté výsledky a porovnanie s inými metódami

Po experimentoch zameraných na ladenie parametrov metódy získavania príznačkov z obrázu nasleduje aplikácia metódy na kompletne databázy obrázkov. Testovali sme úspešnosť metódy na databáze obrázkov Caltech 101 s kategóriou pozadí a na databáze Písmen. Úspešnosť sme merali pre prípad 15 a 30 tréningových obrázkov na kategóriu, najmä kvôli možnosti porovnať dosiahnuté výsledky s inými prácami.

Popis experimentu:

- pre prípad 15 tréningových obrázkov na kategóriu vypočítame C2 príznaky s nastavením: 4 orientácie, 5000 šablón s veľkosťou náhodne od 4×4 po 16×16 ,
- pre prípad 30 tréningových obrázkov na kategóriu vypočítame C2 príznaky s nastavením: 12 orientácií, 5000 šablón s veľkosťou náhodne od 4×4 po 16×16 ,
- pred každým tréningom SVM nájdeme optimálne parametre SVM,
- natrénujeme a následne otestujeme SVM,
- dosiahnuté výsledky zaznamenáme a porovnáme s inými prácami.

Tabuľka 7.7 obsahuje namerané hodnoty úspešnosti kategorizácie pre jednotlivé prípady. Dosiahnuté výsledky sú uspokojivé. Podľa očakávaní je úspešnosť vyššia na databáze Písmena ako na databáze Caltech 101.

Počet tréningových obrázkov/kat.	Caltech 101	Písmena
15 obrázkov	41,9884	60,0714
30 obrázkov	54,6680	82,2504

Tabuľka 7.7: Namerané hodnoty úspešnosti v %.

V ďalšej tabuľke 7.8 porovnáваме namerané hodnoty s inými metódami, ktoré prezentovali svoje výsledky dosiahnuté na databáze Caltech 101. Porovnaním zistíme, že táto práca sa umiestnila zhruba v strede, čo je úspech lebo ostatné práce sa s problematikou rozpoznávania objektov zaoberajú už dlhší čas.

Model	15 tréningových obrázkov/kat.	30 tréningových obrázkov/kat.
Serre <i>et al.</i>	35	42
Holub <i>et al.</i>	37	43
<i>táto práca</i>	42	55
Grauman & Darrell	50	58
Mutch & Lowe	51	56
Lazebnik <i>et al.</i>	56	65
Zhang <i>et al.</i>	59	66

Tabuľka 7.8: Publikované výsledky úspešnosti kategorizácie pre Caltech 101 v %.

Na záver tejto kapitoly aj celej práce si ukážeme príklad úspešnej a neúspešnej kategorizácie objektu pre databázu Caltech 101. Ide o ukážku zaradenia objektov do kategórie



Obrázek 7.7: Ukážka správnej a chybnej kategorizácie pre kategóriu Ying Yang z databáze Caltech 101.

Ying Yang, ktorá obsahuje rôzne vyobrazenia tohto motívu. Obrázky *a*, *b*, *c* na obrázku 7.7 obsahujú úspešne klasifikované objekty no obrázok *d* obsahuje futbalovú loptu taktiež zaradenú klasifikátorom do kategórie Ying Yang. Môžeme si všimnúť, že toto zaradenie nie je neopodstatnené, lebo lopta na obrázku *d* má veľmi podobné rysy ako znak Ying Yang na obrázkoch *a*, *b*.

Kapitola 8

Záver

Cieľom tejto práce bolo preštudovať metódy rozpoznávania objektov v obraze so zameraním predovšetkým na metódy inšpirované v biológii. Vybrať si konkrétnu metódu a použiť ju na rozpoznávanie a kategorizáciu objektov v obraze.

V tejto práci je použitá metóda prezentovaná pánmi Mutch a Lowe [16]. Model patrí do kategórie hierarchických modelov, inšpirovaných hierarchickou štruktúrou vizuálnej kôry u primátov. Model získava príznaky z obrazu pomocou gáborových filtrov a porovnávaním so šablónami. Hoci Gáborové filtre patria medzi metódy získavania hustých príznakov, výsledná metóda sa zaraďuje medzi riedke príznaky. Je to kvôli tomu, že metóda redukuje počet príznakov a k výslednej klasifikácii používa len niektoré z nich, nie všetky, ktoré boli získané z obrazu.

Výsledkom práce je implementácie modelu získavania príznakov z obrazu, ktorej úspešnosť je porovnateľná so súčasnými „state of the art“ metódami rozpoznávania objektov. Klasifikátor pracuje s úspešnosťou 54,7% pri 102 kategóriách databáze Caltech 101, čo znamená, že každý druhý obrázok priradí do správnej kategórie a to je pri takom množstve kategórií na databáze Caltech 101 úspech.

S prácou som sa zúčastnil študentskej konferencie EEICT 2011, kde som sa dostal do finálového kola a prezentoval dosiahnuté výsledky v posterovej sekcii.

Osobne ma vždy zaujímalo spojenie vedy a prírody. Fakt, že mnoho úspešných ľudských vynálezov bolo inšpirovaných prírodou. Preto si myslím a verím, že to platí aj pre oblasť počítačového videnia. Skúmanie a pochopenie činnosti ľudského mozgu postupne vedie k čoraz presnejším, výkonnejším a použiteľnejším modelom, ktoré aproximujú jeho činnosť.

Do budúcnosti plánujem použiť iné druhy klasifikátorov, napríklad Neurónovú sieť. Použiť databázu obrázkov Caltech 256, ktorá je náročnejšia ako Caltech 101 a obsahuje až 256 kategórií objektov. Taktiež sa plánujem zamerať na skúmanie príznakového priestoru, konkrétne na redukciu príznakového priestoru (metóda PCA) a vizualizáciu príznakov v paralelných súradniciach.

Literatura

- [1] Abrahámová, J.; Adamec, J.; Adamová, L.; aj.: *Všeobecná encyklopedie 3. svazek*. Praha: Nakladatelský dům OP, 1996, ISBN 80-85841-35-5, s. 349–350.
- [2] Bay, H.; Ess, A.; Tuytelaars, T.; aj.: SURF: Speeded Up Robust Features. In *Computer Vision and Image Understanding*, ročník 110, 2008, s. 346–359.
- [3] Dubuc, B.: The Brain From Top To Bottom. 2010 [cit. 2010-12-23], [online]. URL <http://thebrain.mcgill.ca/>
- [4] Duda, R. O.; Hart, P. E.; Stork, D. G.: *Pattern Classification*. New York: Wiley, druhé vydání, 2001, ISBN 0-471-05669-3.
- [5] Fisher, R. A.: The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, ročník 7, 1936: s. 179–188.
- [6] Heisele, B.; Serre, T.; Pontil, M.; aj.: Categorization by learning and combining object parts. In *NIPS*, Vancouver, 2001.
- [7] Hubel, D.; Wiesel, T.: Receptive fields of single neurones in the cat's striate cortex. *Journal of Physiology*, 1959: s. 574–591.
- [8] Jain, R.; Kasturi, R.; Schnuck, B. G.: *Machine vision*. McGraw-Hill, 1995, ISBN 0-07-032018-7, s. 474–479.
- [9] Jolliffe, I. T.: *Principal Component Analysis*. New York: Springer, druhé vydání, 2002, ISBN 0-387-95442-2.
- [10] Lee, T. S.: Image Representation Using 2D Gabor Wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 18, č. 10, 1996: s. 959–971.
- [11] Lienhart, R.; Maydt, J.: An extended set of Haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, ročník 1, 2002, s. 900–903.
- [12] Lowe, D. G.: Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, Kerkyra, Greece, 1999, ISBN 0-7695-0164-8, s. 1150–1157.
- [13] Mohan, A.; Papageorgiou, C.; Poggio, T.: Example-based object detection in images by components. In *PAMI*, ročník 23, 2001, s. 349–361.
- [14] Movellan, J. R.: Tutorial on Gabor Filters. 2008 [cit. 2010-12-22], [online]. URL <http://mplab.ucsd.edu/tutorials/pdfs/gabor.pdf>

- [15] Mu, Y.; Yan, S.; Liu, Y.; aj.: Discriminative local binary patterns for human detection in personal album. In *Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, ISBN 978-1-4244-2242-5, s. 1–8.
- [16] Mutch, J.; Lowe, D. G.: Object Class Recognition and Localization Using Sparse Features with Limited Receptive Fields. *International Journal of Computer Vision*, ročník 80, č. 1, 2008: s. 45–57, ISSN 0920-5691.
- [17] Ojala, T.; Pietikäinen, M.; Harwood, D.: A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 1996: s. 51–59.
- [18] Papageorgiou, C. P.; Oren, M.; Poggio, T.: A General Framework for Object Detection. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, Washington, DC, USA: IEEE Computer Society, 1998 [cit. 2010-12-26], ISBN 81-7319-221-9, s. 555–556.
- [19] Riesenhuber, M.; Poggio, T.: Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 1999: s. 1019–1025.
- [20] Schneiderman, H.; Kanade, T.: A statistical method for 3D object detection applied to faces and cars. In *CVPR*, 2000, s. 746–751.
- [21] Serre, T.; Oliva, A.; Poggio, T.: A feedforward architecture accounts for rapid categorization. *Biological Sciences - Neuroscience*, 2007: s. 6424–6429.
- [22] Serre, T.; Wolf, L.; Poggio, T.: Object Recognition with Features Inspired by Visual Cortex. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '05, Washington, DC, USA: IEEE Computer Society, 2005, ISBN 0-7695-2372-2, s. 994–1000.
- [23] Svoboda, T.; Kyblic, J.; Hlavac, V.: *Image Processing, Analysis, and Machine Vision*. Thomson Engineering, 2008, ISBN 0-495-08525-X, s. 396–402.
- [24] Ullman, S.; Vidal-Naquet, M.; Sali, E.: Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 2002: s. 682–687.
- [25] Viola, P.; Jones, M.: Robust real-time object detection. In *ICCV*, 2001 [cit. 2010-12-28], s. 1254–1259.
- [26] Weber, M.; Welling, M.; Perona, P.: Unsupervised learning of models for recognition. In *ECCV*, Dublin, Ireland, 2000.
- [27] Wikipedia: Eye — Wikipedia, The Free Encyclopedia. 2011 [cit. 2011-04-27], [online]. URL <http://en.wikipedia.org/wiki/Eye>
- [28] Zhang, L.; Chu, R.; Xiang, S.; aj.: Face Detection Based on Multi-Block LBP Representation. In *Advances in Biometrics, Lecture Notes in Computer Science*, ročník 4642, editace S.-W. Lee; S. Li, Springer Berlin / Heidelberg, 2007, s. 11–18.

Dodatek A

Obsah DVD

Popis adresárovej štruktúry priloženého DVD:

- **doc/** priečinok obsahuje technickú správu a plagát,
- **data/** priečinok obsahuje množiny obrázkov Caltech 101 a Písmena,
- **lib/** priečinok obsahuje použité knižnice OpenCV, CvGabor a LIBSVM,
- **src/** priečinok obsahuje zdrojové súbory,
- **res/** priečinok obsahuje namerané hodnoty, natrénované SVM modely.

Dodatek B

Manuál

B.1 Ukážka konfiguračného súboru

```
# Random Number Generator
seedParser      12345      # generovanie patchSize
seedS2          12345      # vyberanie scale a pozicie [x,y]
seedController  12345      # vyberanie prototypov, vyberanie obrazkov

# IMAGE LAYER: Scale parameters
numOfScales     10         # defalut    10
scaleFactor     1.1892     # default  1.1892
shorterEdge     140        # default  140

# S1 LAYER: Gabor parameters
numOfOrientations  4      # podľa nich sa urci uhol od 0 do PI, default 4
iNu                2      # default  2
dSigma             0.6    # default  0.6
dF                 2      # default  2

# C1 LAYER: 3D Max filter parameters
size              10      # default  10
deep              2       # default  2
step              5       # default  5

# S2 LAYER:
numOfPrototypes  500      # default  4075, oznacene d
patchSizeMin     1        # moznosti 1=4x4, 2=8x8, 3=12x12, 4=16x16
patchSizeMax     4        # moznosti 1=4x4, 2=8x8, 3=12x12, 4=16x16

# TRAINING
numOfCategories   102     # pocet kategorii
numOfImagesPerCat 15     # pocet obrazkov kategorie pouzitych na trenovanie
```

B.2 Ukážka súboru so zoznamom obrázkov

```
+---accordion
|     image_0001.jpg
|     image_0002.jpg
|     image_0003.jpg
|     image_0004.jpg
|     image_0005.jpg
|
+---airplanes
|     image_0001.jpg
|     image_0002.jpg
|     image_0003.jpg
|     image_0004.jpg
|
+---ant
|     image_0001.jpg
|     image_0002.jpg
|     image_0003.jpg
|     image_0004.jpg
|     image_0005.jpg
|     image_0006.jpg
|
+---yin_yang
|     image_0001.jpg
|     image_0002.jpg
|     image_0003.jpg
|     image_0004.jpg
|     image_0005.jpg
|     image_0006.jpg
|     image_0007.jpg
|     image_0008.jpg
|     image_0009.jpg
```