

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

KLASIFIKACE OBJEKTŮ V OBRAZE PODLE TEXTURY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JIŘÍ HUTÁREK

BRNO 2008



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

KLASIFIKACE OBJEKTŮ V OBRAZE PODLE TEXTURY

TEXTURE-BASED OBJECT RECOGNITION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VEDOUCÍ PRÁCE
SUPERVISOR

JIŘÍ HUTÁREK

Ing. MICHAL ŠPANĚL

BRNO 2008

Abstrakt

Práce se zabývá detekcí textur v obraze pomocí metody *Local binary patterns* (LBP). Texturním příznakem je histogram hodnot LBP ve zkoumaném obraze a klasifikace textur je prováděna pomocí umělé neuronové sítě. S použitím těchto metod je vytvořen obecný detektor textur, jehož konkrétní aplikace je zaměřena na rozpoznávání objektů na leteckých snímcích.

Klíčová slova

textura, local binary patterns, klasifikace, segmentace, neuronová síť, počítačové vidění, MDSTk, letecký snímek

Abstract

This thesis focuses on texture detection through the use of *Local binary patterns* (LBP). The LBP histogram extracted from the image is used as the texture feature and the image is classified by an artificial neural network. By using these methods, an universal texture detector is implemented, which is then specialized for the aerial image object recognition.

Keywords

texture, local binary patterns, classification, segmentation, neural network, computer vision, MDSTk, aerial image

Citace

Jiří Hutárek: Klasifikace objektů v obraze podle textury, bakalářská práce, Brno, FIT VUT v Brně, 2008

Klasifikace objektů v obraze podle textury

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Španěla.

.....
Jiří Hutárek
12. května 2008

Poděkování

Děkuji vedoucímu práce Ing. Michalu Španělovi za vstřícnost, odbornou pomoc a cenné rady při tvorbě této práce.

© Jiří Hutárek, 2008.

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 Úvod	2
2 Zpracování obrazu, analýza a klasifikace textur	3
2.1 Textura a její vlastnosti	3
2.2 Motivace	3
2.3 Rozdělení a klasifikace textur	4
2.4 LBP Metodologie	6
2.5 Popis textur pomocí LBP	9
3 Umělé neuronové sítě a jejich užití pro klasifikaci textur	11
3.1 Vlastnosti ovlivňující úspěšnost umělé neuronové sítě	12
3.2 Kaskádovité učení umělé neuronové sítě	12
4 Návrh detektoru textur pro letecké snímky	14
4.1 Klasifikace textur	14
4.2 Segmentace obrazu	15
4.3 Struktura systému pro detekci textur	17
4.4 Detekce objektů na leteckých snímcích podle textury	19
5 Implementace detektoru textur a detekce objektů	25
5.1 Nástroje pro implementaci	25
5.2 Implementace jednotlivých částí detektoru textur	26
6 Výsledky	27
6.1 Extrakce texturních příznaků	27
6.2 Neparametrická klasifikace	28
6.3 Klasifikace pomocí umělé neuronové sítě	30
6.4 Výstupy detektoru textur	31
7 Závěr	36
A Ukázka segmentace reálných obrazových dat	37
B Příručka uživatele pro nově implementované moduly	41

Kapitola 1

Úvod

Tato práce se zabývá jedním z významných témat počítačové grafiky, kterým je počítačové vidění. Tato oblast má široké možnosti uplatnění v praxi a stejně tak bohatý je i repertoár zde použitelných metod, přičemž právě textura je jednou z cenných informací pro analýzu obrazu.

Cílem této práce je prozkoumat problematiku zpracování obrazu a detekce objektů v obraze podle textury – je tedy nutné zvolit, podle jakých texturních příznaků se budou textury klasifikovat, a vybrat vhodný nástroj pro jejich klasifikaci. Na tomto teoretickém základě je pak implementován detektor textur, který je specializován na rozpoznávání objektů na leteckých snímcích.

První teoretická kapitola (2) popisuje obecné základy zpracování obrazu a vysvětluje základní pojmy počítačové grafiky. Dále představuje texturu a její vlastnosti a věnuje se teorii popisu textur.

Druhá teoretická kapitola (3) se zevrubně věnuje problematice umělých neuronových sítí, které budou použity pro klasifikaci textur. Jsou nastíněny vlastnosti a parametry sítě důležité pro úspěšnou klasifikaci v této oblasti. Obě teoretické kapitoly byly již náplní semestrálního projektu.

Kapitola 4 se již týká vlastního návrhu detektoru textur pro letecké snímky. Zabývá se konkrétními způsoby klasifikace textur, způsobem segmentace obrazu i vlastní strukturou systému pro detekci textur. V závěru kapitoly jsou pak prezentovány objekty, které systém bude detekovat. V následující kapitole (5) jsou zdůrazněny význačné body při vlastní implementaci systému.

V kapitole 6 jsou pak prezentovány praktické výsledky činnosti detektoru textur (na umělých a následně i reálných obrazových datech) a je provedena diskuze výstupů systému.

Závěrečná kapitola práce pak shrnuje dosažené výsledky a nastiňuje možný další vývoj a rozšíření v rámci diplomové práce.

Kapitola 2

Zpracování obrazu, analýza a klasifikace textur

Počítačová grafika jako součást informatiky se zabývá tvorbou (syntézou) nebo analýzou grafické obrazové informace [3]. Tato práce spadá do oblasti *počítačového vidění*, kdy obrazovou informaci (v případě této práce se jedná o letecké snímky) chceme jistým způsobem interpretovat – konkrétně rozpoznat objekty zvoleného typu. *Obrazem* budeme dále rozumět obrazovou informaci převedenou do formy 2D rastrové matice složené z jednotlivých *obrazových bodů*, kterou lze získat např. snímáním digitální kamerou.

Klasifikace obrazu je pak proces, který bodům daného obrazu přiřadí příslušnost k nějaké třídě nebo oblasti. Způsobů klasifikace obrazu je celá řada, tato práce se zabývá statistickou klasifikací obrazu na základě textur pomocí metody *Local binary patterns*.

2.1 Textura a její vlastnosti

Pro texturu zatím nebyla nalezena obecná jednotná definice, přičemž existují pochybnosti, zda taková definice vůbec existuje. Z tohoto důvodu je analýza textur problematická, protože správnost analytických metod nemůže být matematicky dokázána a musí tedy být ověřována empiricky – navzdory tomu je analýza textur úspěšně aplikována v řadě oblastí.

Textura může být neformálně definována např. jako vizuální nebo hmatová charakteristika povrchu tělesa [4]. Téměř všechny objekty ve vesmíru, pokud je pozorujeme z vhodné vzdálenosti, mají texturu. Velikost elementů, ze kterých textura sestává, se různí – od velmi malých částic (zrnka písku) po obrovské (hvězdy v galaxii). Vnímání textury závisí mimo jiné na měřítku a úhlu pohledu (např. textura papírních linií na lidské kůži je viditelná pouze do jisté vzdálenosti), na prostředí a na osvětlení.

Textura jako vlastnost obrazu může být charakterizována jako opakující se struktura primitiv, tzv. *texelů* [10]. Texel má stejný tvar, rozložení intenzit, orientaci či velikost.

2.2 Motivace

Textury hrají důležitou roli v analýze obrazů a počítačovém vidění. Ačkoliv informace o barvách v obraze je taktéž významná, jsou situace, kdy ji nelze použít (např. při změnách osvětlení) nebo takovou informaci vůbec nemáme (např. obraz z kamery pro noční vidění).



Obrázek 2.1: Příklady různých textur

Oblasti, ve kterých analýza textur nalézá uplatnění, jsou např. klasifikace a kontrola kvality materiálů, analýza medicínských obrazů, rekonstrukce povrchu analýzou textury, biometrie nebo klasifikace a segmentace reálných scén – v případě této práce se jedná o letecké snímky.

Rozpoznávání objektů na leteckých snímcích je vhodnou aplikací analýzy textur. S jejich pomocí lze v závislosti na zvoleném měřítku rozeznávat různé druhy porostů (na základě textury je např. možné rozpoznat les napadený kůrovcem od zdravého lesa), zemědělských ploch (rozpoznávání osetých plodin), zástavby (hustota, typ) nebo dalších objektů. Problém poměrně značné variability a nehomogenity textur v reálných leteckých snímcích lze řešit klasifikací využívající metod umělé inteligence – v tomto případě pomocí umělé neuronové sítě.

2.3 Rozdělení a klasifikace textur

Textury mohou být na základě jejich vzhledu kategorizovány několika způsoby založenými na lidském vnímání. Posuzovanými vlastnostmi pak mohou být např. drsnost, pravidelnost, kontrast, směrovost nebo komplexnost [14]. Z hlediska analytických metod se rozlišují dva významné druhy textur: strukturní a stochastické.

2.3.1 Strukturní textury

Strukturní textury jsou často umělé (vytvořené člověkem – např. cihlová zeď) a velmi pravidelné. Textury jsou v takovém případě dostatečně velké a snadno odlišitelné od pozadí. Popis takových textur pak vychází z typů a rozmístění elementů, které je tvoří. Strukturní textury se nejčastěji popisují polygonálními sítěmi nebo gramatikami.



Obrázek 2.2: Příklady strukturních textur



Obrázek 2.3: Příklady stochastických textur

2.3.2 Stochastické textury

Stochastické textury jsou nejčastěji přírodního původu. Jelikož v tomto případě už nelze jednotlivé textury snadno odlišit, při popisu tohoto typu textur se vychází z rozložení intenzit. Jelikož počet primitiv v textuře je mnohem větší než jejich variabilita, jsou vhodným nástrojem pro popis stochastických textur statistické metody. Z textury jsou tak získány *příznakové vektory*, které se následně klasifikují do několika tříd. Dvě z perspektivních metod pro analýzu stochastických textur jsou uvedeny v následujících kapitolách:

Gáborovská analýza

Gáborovská analýza je založena na filtrování obrazu bankou filtrů. Texturními příznaky jsou pak odezvy filtrů v jednotlivých bodech. Tato metoda vykazuje v praxi velice dobré výsledky a je v současné době považována za špičkový nástroj v oblasti analýzy textur.

Nevýhodami této metody jsou především poměrně významná výpočetní náročnost (díky použitému filtrování) a ovlivnitelnost změnami osvětlení.

Local binary patterns

Metoda *Local binary patterns* (LBP) [6] byla původně vytvořena k doplnění informace o lokálním kontrastu obrazu. Je to velmi silná metoda, která v sobě kombinuje statistický a strukturní přístup k analýze textur. Na základě LBP vzniká v poslední době vcelku velké množství variant a odvozených metod a výzkum v této oblasti se dá označit jako slibný. Dokonce se ukazuje, že metodologie odvozené z LBP by nemusely sloužit jen pro popis textur, ale že by se mohly stát obecnými nástroji pro popis i jiných vlastností obrazu.

Princip klasifikace textur na základě LBP je jednoduchý: pro každý bod analyzovaného obrazu je vypočten LBP kód a texturním příznakem pro daný obraz je pak histogram těchto kódů. Ze základní verze LBP, která pracovala s 8-mi okolím obrazového bodu, byly postupně odvozeny další varianty, z nichž některé budou v následující kapitole prezentovány.

Hlavními přednostmi této metody jsou výpočetní nenáročnost, invariance vůči rotaci a vůči změnám osvětlení, relativně jednoduchá implementace (a to i v hardwaru) a především pak velmi dobrá schopnost extrakce texturních příznaků z obrazu. Z těchto důvodů byla tato metoda zvolena pro implementaci v rámci této práce.

2.4 LBP Metodologie

2.4.1 Odvození

Následující odvození vychází z [4]. Jak již bylo uvedeno, pro texturu neexistuje univerzální definice. Pro účely LBP je proto textura T v lokálním obrazovém okolí definována pomocí distribuce intenzit $P + 1$ ($P > 0$) obrazových bodů:

$$T = t(g_c, g_0, \dots, g_{P-1}), \quad (2.1)$$

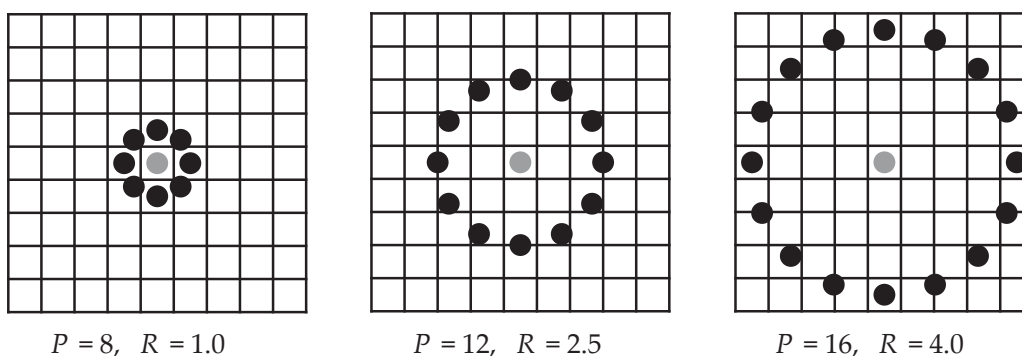
kde g_c je intenzita středového obrazového bodu a g_p ($p = 0, \dots, P - 1$) jsou intenzity P od sebe stejně vzdálených obrazových bodů ležících na kružnici o poloměru R ($R > 0$) kolem středového obrazového bodu (viz obrázek 2.4).

Souřadnice bodů ležících na kružnici kolem středového bodu jsou tedy pak

$$x_p = x_c + R \cos \frac{2\pi p}{P} \quad (2.2)$$

$$y_p = y_c + R \sin \frac{2\pi p}{P}, \quad (2.3)$$

kde (x_c, y_c) jsou souřadnice středového bodu. Intenzity okolních obrazových bodů, jejichž souřadnice neodpovídají přesně jednotlivým obrazovým bodům, jsou interpolovány¹.



Obrázek 2.4: Středový obrazový bod a okolní obrazové body ležící na kružnici kolem něj. Body, které neleží přesně na mřížce, jsou interpolovány. Zobrazeny jsou tři zřejmě nejpoužívanější kombinace hodnot P a R .

¹Pro $P = 8$ a $R = 1.0$ nejsou v praxi okolní body interpolovány, ale berou se body v 8-mi okolí středového bodu.

Lokální textura pak může být beze ztráty informace reprezentována pomocí intenzity středového obrazového bodu a rozdílů mezi ním a okolními obrazovými body:

$$T = t(g_c, g_0 - g_c, \dots, g_{P-1} - g_c). \quad (2.4)$$

Za předpokladu, že rozdíly jsou nezávislé na středovém obrazovém bodě, lze pak psát:

$$T \approx t(g_c) t(g_0 - g_c, \dots, g_{P-1} - g_c). \quad (2.5)$$

Jelikož $t(g_c)$ vyjadřuje celkový jas obrazu a nemá tedy na lokální texture vliv, lze jej ignorovat – pak tedy dostáváme:

$$T \approx t(g_0 - g_c, \dots, g_{P-1} - g_c). \quad (2.6)$$

Tím je dosaženo invariance vůči změnám intenzity obrazu. Abychom dosáhli invariance vůči změnám velikostí rozdílů $g_p - g_c$, budeme dále brát ohled pouze na znaménka těchto rozdílů:

$$T \approx t(s(g_0 - g_c), \dots, s(g_{P-1} - g_c)), \quad (2.7)$$

kde

$$s(x) = \begin{cases} 1 & \text{pro } x \geq 0 \\ 0 & \text{pro } x < 0. \end{cases} \quad (2.8)$$

Konečně každému znaménku rozdílu $s(g_p - g_c)$ je přiřazena binomická váha 2^p a dostáváme tak pro každou kombinaci rozdílů znamének unikátní LBP kód, který charakterizuje lokální texture okolo obrazového bodu na souřadnicích (x_c, y_c) :

$$\text{LBP}_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p. \quad (2.9)$$

Příklad výpočtu základního $\text{LBP}_{8,1}$ kódu je na obrázku 2.5

intenzity			prahování			váhy		
6	5	2	1	0	0	8	4	2
7	6	1	1		0	16		1
9	8	7	1	1	1	32	64	128

Obrázek 2.5: Výpočet základního $\text{LBP}_{8,1}$ kódu. Intenzity obrazu jsou podle vztahu 2.9 prahovány dle středového obrazového bodu a okolním obrazovým bodům jsou přiřazeny váhy. LBP kód pro daný obrazový bod je tedy binárně 11111000, decimálně $8 + 16 + 32 + 64 + 128 = 248$.

2.4.2 Použití v praxi

V praxi je vztah 2.9 interpretován jako bezznaménkové binární číslo o délce P bitů² (takový LBP kód pak tedy může nabývat 2^P různých hodnot). Texture (jako distribuci intenzit v obraze) lze pak popsat pomocí histogramu o 2^P sloupcích:

$$T \approx t(\text{LBP}_{P,R}(x_c, y_c)). \quad (2.10)$$

²Toto je jeden z důvodů příznivé výpočetní náročnosti LBP, jelikož současné CPU disponují instrukcemi pro bitové operace. Výhody tohoto přístupu využívají i následující rozšíření LBP.

Tento histogram je pak užit jako texturní příznak. "Základní" LBP jsou ovšem z několika důvodů nepraktické:

- Délka histogramu roste exponenciálně s počtem bodů v okolí, což ztěžuje následnou klasifikaci (žádoucí je menší délka histogramu).
- Základní verze není invariantní vůči rotaci.
- Frekvence výskytů jednotlivých LBP kódů se významně liší – některé kódy jsou statisticky nezajímavé.
- Nevýhodou základního LBP i jeho následujících variant je lokálnost – LBP o fixním poloměru špatně postihují širší prostorové závislosti v textuře. Toto řeší varianta *multi-resolution LBP* [7], která je nad rámec této práce (stručně lze říci, že multi-resolution LBP kombinuje do jednoho texturního příznaku několik LBP kódů s různými hodnotami P a R).

2.4.3 Invariance vůči rotaci

Jelikož obrazové body v okolí středového bodu jsou rozmístěny na kružnici, lze invariance vůči rotaci dosáhnout jednoduše – za střed rotace je považován každý obrazový bod a jemu příslušný $LBP_{P,R}$ kód je rotován do referenční polohy – svého minima [9]. Tím se všechny rotované verze jednoho kódu mapují na jednu hodnotu (a zároveň se zmenšuje celkový počet různých LBP kódů v této variantě).

Rotačně invariantní LBP kód je tedy definován takto:

$$LBP_{P,R}^{ri} = \min \{ ROR(LBP_{P,R}, i) \mid i = 0, 1, \dots, P-1 \} \quad (2.11)$$

Funkce $ROR(x, i)$ provádí bitový posuv binárního čísla x o délce P bitů i -krát doprava ($|i| < P$). Pro binární číslo x :

$$x = \sum_{k=0}^{P-1} 2^k a_k, \quad a_k \in \{0, 1\}, \quad (2.12)$$

je funkce ROR definována takto:

$$ROR(x, i) = \begin{cases} \sum_{k=i}^{P-1} 2^{k-i} a_k + \sum_{k=0}^{i-1} 2^{P-i+k} a_k & i > 0 \\ x & i = 0 \\ ROR(x, P+i) & i < 0 \end{cases} \quad (2.13)$$

Je evidentní, že jednotlivým $LBP_{P,R}^{ri}$ kódům přísluší různý počet $ROR_{P,R}$ kódů, které se na ně mapují (krajními případy jsou kódy obsahující samé jedničky nebo samé nuly, které se mapují samy na sebe). Příklady rotačně invariantních LBP kódů jsou na obrázku 2.6.

Nejmenší úhel, pro který je rotačně invariantní LBP kód uzpůsoben, závisí na počtu bodů v okolí středového bodu P : $\alpha_{\min} = \frac{360^\circ}{P}$. Konkrétně pro $LBP_{8,R}$ je tento úhel 45° , což je pro praxi příliš "hrubá" kvantizace. Toto lze řešit zvětšováním počtu P bodů v kruhovém okolí, avšak zároveň je třeba vzít v úvahu již zmíněný exponenciální nárůst délky histogramu a zároveň volit poloměr R tak, aby body nebyly příliš blízko sebe a tím nezanášely do získaných dat zbytečnou redundanci.

2.4.4 Uniformní LBP

Vycházíme z heuristické hypotézy, že jistá podmnožina $LBP_{8,1}$ kódů tvoří valnou většinu (někdy až přes 90 %) všech $LBP_{8,1}$ kódů extrahovaných z běžně se vyskytujících textur [8]. Tato podmnožina se nazývá *uniformní LBP*.

Uniformní LBP mají omezený počet přechodů mezi jedničkami a nulami – LBP kódy s vyšším počtem těchto přechodů jsou pro popis textury statisticky nespolehlivé.

Uniformita U okolí G obrazového bodu je definována:

$$U(G_P) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|. \quad (2.14)$$

Za uniformní jsou považovány LBP kódy, pro které je $U \leq 2$.

Pro binární číslo x o délce P bitů lze uniformitu vypočítat takto:

$$U(x) = \sum_{p=0}^{P-1} F(x \text{ xor ROR}(x, 1), p), \quad (2.15)$$

Funkce $F(x, i)$ extrahuje i -tý bit z binárního čísla x :

$$F(x, i) = \text{ROR}(x, i) \text{ and } 1. \quad (2.16)$$

Příklady uniformních LBP kódů jsou na obrázku 2.6.

Spojením výhodných vlastností rotačně invariantního a uniformního LBP dostáváme následující, v praxi velice dobře použitelnou variantu LBP:

2.4.5 Rotačně invariantní uniformní LBP

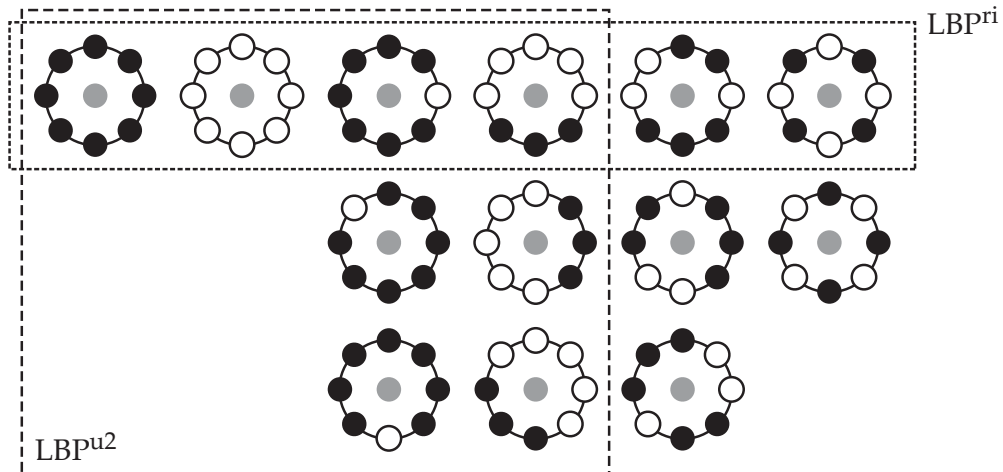
Na kruhové okolí obrazového bodu, jehož LBP kód je uniformní, můžeme pohlížet tak, že obsahuje pouze dvě souvislé "oblasti". Pak rotačně invariantní varianta tohoto uniformního kódu (která vznikne rotací do minimální polohy) může být charakterizována pouze poměrem souvislých "oblastí" nul a jedniček. Různých rotačně invariantních uniformních kódů ($LBP_{P,R}^{\text{riu}_2}$) je tedy celkem $P + 1$.

$LBP_{P,R}^{\text{riu}_2}$ kód je prakticky určen počtem jedniček, pokud je základní LBP kód uniformní – neuniformní kódy jsou mapovány všechny na jednu hodnotu:

$$LBP_{P,R}^{\text{riu}_2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & U(G_P) \leq 2 \\ P + 1 & \text{jinak.} \end{cases} \quad (2.17)$$

2.5 Popis textur pomocí LBP

V této kapitole bude rozebráno, jak lze prakticky využít nástroje LBP ke klasifikaci textur, a budou prezentovány použité metody klasifikace založené na neparametrické (statistické) klasifikaci a klasifikaci s využitím učení s učitelem.



Obrázek 2.6: Příklady kódu $LBP_{8,1}$. Kódy v horní řadě jsou rotačně invariantní. Kódy v každém sloupci vzniknou rotací kódu v prvním řádku daného sloupce. Kódy v levém rámečku jsou uniformní. Kódy v průsečíku rámečků jsou tedy $LBP_{8,1}^{riu_2}$.

2.5.1 Texturní příznak

Pro každý bod zkoumaného obrazu je vypočten LBP kód. Texturním příznakem je pak histogram LBP kódů získaných ze všech bodů zkoumaného obrazu. V praxi je nejvhodnější variantou LBP pro toto použití $LBP_{P,R}^{riu_2}$ (viz kapitolu 2.4.5), jelikož ostatní prezentované varianty LBP nejsou invariantní vůči rotaci nebo obsahují statisticky nevýznamné hodnoty, resp. jejich histogram má příliš velký počet sloupců.

Problém klasifikace textur tedy budeme dále řešit jako problém klasifikace histogramů LBP.

Normalizace texturního příznaku

Abychom mohli vzájemně porovnávat textury získané z různě velkých obrazů (čili LBP histogramy obsahující různé počty vzorků), je třeba LBP histogramy před klasifikací normalizovat – vhodné řešení je:

$$b'_j = \frac{b_j}{\sum_{i=1}^B b_i}, \quad (2.18)$$

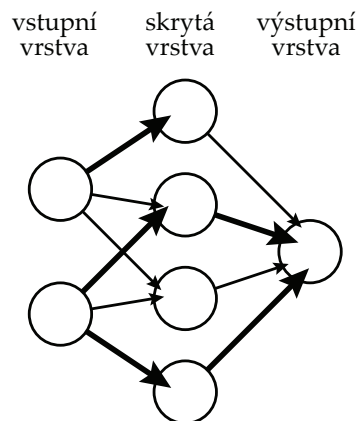
kde b_j je právě je normalizovaný sloupec původního histogramu, b_i je sloupec původního histogramu, b'_j je sloupec normalizovaného histogramu a B je počet sloupců histogramu.

Kapitola 3

Umělé neuronové sítě a jejich užití pro klasifikaci textur

Problematika umělých neuronových sítí je velmi rozsáhlá a není těžištěm této práce, proto zde budou naznačeny pouze základní principy důležité pro aplikaci v oblasti klasifikace textur.

Na umělou neuronovou síť můžeme zjednodušeně pohlížet jako na adaptivní výpočetní model inspirovaný biologickými neuronovými sítěmi, kde vzájemně propojená skupina neuronů mění v tréninkové fázi svoji strukturu na základě vstupních a výstupních dat. Pokud jsou parametry umělé neuronové sítě pro daný problém vhodně nastaveny, mohou být výsledky poskytované umělou neuronovou sítí velmi robustní (zároveň je ale poměrně obtížné pro daný problém najít nejvhodnější parametry sítě).



Obrázek 3.1: Příklad jednoduché neuronové sítě sestávající ze 2 vstupních neuronů, 4 neuronů ve skryté vrstvě a jednoho výstupního neuronu.

Základní topologie umělé neuronové sítě sestává ze skupiny vstupních a výstupních neuronů, mezi kterými se nachází vrstva (případně několik vrstev) skrytých neuronů (viz obrázek 3.1). Ve fázi učení přivádíme na vstupní neurony trénovací data a zároveň síti předkládáme očekávaný výsledek. Síť pak v závislosti na těchto datech mění svoji strukturu tak, aby při vstupu neznámých dat dovedla tato data na základě tréninku klasifikovat.

3.1 Vlastnosti ovlivňující úspěšnost umělé neuronové sítě

V aplikaci umělé neuronové sítě pro účely této práce je třeba vhodně zvolit zejména parametry, které mají vliv na následující vlastnosti sítě:

3.1.1 Volba architektury sítě

Obecně je volba architektury sítě (počet vstupů a výstupů, počet skrytých neuronů, spoje mezi nimi, tréninkový algoritmus) pro konkrétní problém nesnadná, protože ji lze jen těžko předvídat, ale přitom může mít na úspěšnost sítě fatální vliv. Příliš malé sítě nemusí dobře reprezentovat zvolenou funkci, naproti tomu příliš velké sítě jsou náchylné k přetrénování.

3.1.2 Volba aktivační funkce

Volba aktivační funkce může zásadním způsobem ovlivnit fungování celé sítě. U některých aktivačních funkcí lze volit i jejich strmost a tím dále měnit citlivost sítě. Často používanou aktivační funkcí je symetrický sigmoid, neboli hyperbolický tangens:

$$y = \tanh(sx) = \frac{2}{1 + e^{-2sx}} - 1, \quad (3.1)$$

kde s je požadovaná strmost. Oborem hodnot této funkce je interval $\langle -1, 1 \rangle$.

3.1.3 Přetrénování

Při trénování sítě se může stát, že je síť vystavena příliš velkému množství trénovacích dat, na která se adaptuje příliš "těsně" a ztrácí tak schopnost generalizace. Tomuto lze předejít testováním sítě na neznámých datech v průběhu učení – učení pak končí, když chyba v detekci neznámých dat začne po určitém období klesání opět stoupat.

3.2 Kaskádovité učení umělé neuronové sítě

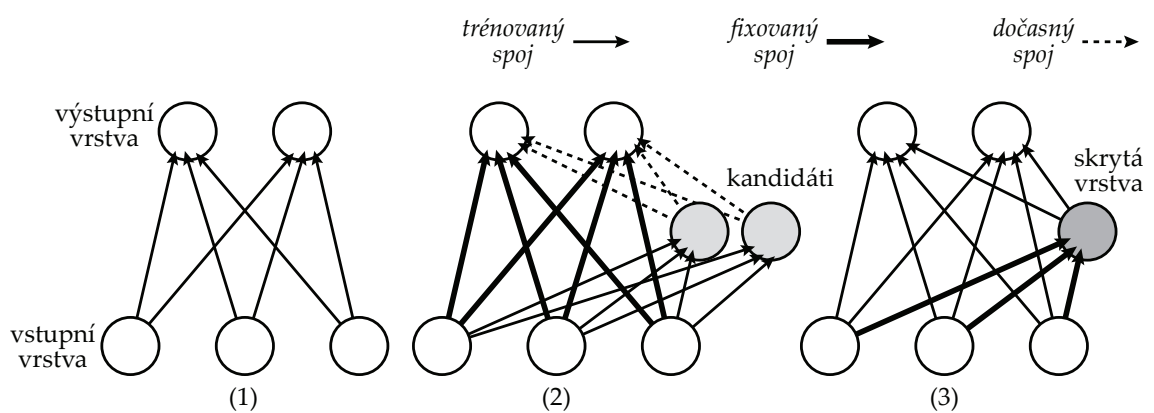
Na rozdíl od "tradiční" architektury umělé neuronové sítě, kdy v průběhu učení jsou upravovány váhy v již pevně dané topologii sítě, kaskádovité učení pracuje tak, že do prázdné sítě (obsahující pouze vstupní a výstupní neurony) postupně přidává jednotlivé neurony. Rodina struktur tohoto typu se nazývá *Cascade-Correlation* [2, 12].

V jednom kroku je generována množina "kandidátských" neuronů. Každý z nich je inicializován jinak než ostatní a trénován nezávisle na ostatních. Následně je vybrán nejslibnější kandidát a ten je přidán do sítě natrvalo. Stávající část sítě je pak upravena pro nejlepší kooperaci s novým neuronem. Jakmile je do sítě přidán nový neuron, váhy na jeho vstupu jsou dále neměnné. Postup činnosti algoritmu je ilustrován na obrázku 3.2.

Tento přístup má několik výhod:

- velmi rychlé učení
- síť sama určí svoji velikost (není tedy třeba odhadovat počet neuronů) a topologii
- síť si zachovává již vytvořenou strukturu, i pokud se soubor trénovacích dat změní

Kritérium pro zastavení učení může být dvojí: konec nastane po dosažení určité chyby (jako v klasické topologii) nebo po určité době stagnace tréninku kandidátních neuronů, což zvyšuje odolnost proti přetrénování sítě.



Obrázek 3.2: První tři fáze kaskádového učení: (1) V síti obsahující pouze vstupní a výstupní vrstvu jsou trénovány spoje mezi těmito dvěma vrstvami. (2) Je vytvořena první skupina kandidátních neuronů a ty jsou trénovány v rámci sítě. (3) Nejslibnější kandidát je vybrán do skryté vrstvy, jeho vstupní spoje jsou fixovány a jsou trénovány spoje vedoucí do výstupních neuronů. Další iterace začíná krokem (2).

Kapitola 4

Návrh detektoru textur pro letecké snímky

Tato kapitola popisuje návrh detektoru textur se specializací pro rozpoznávání objektů na leteckých snímcích. Jsou popsány principy klasifikace a segmentace obrazu i vlastní struktura detektoru textur.

4.1 Klasifikace textur

4.1.1 Neparametrická klasifikace

Shodu distribucí LBP o B sloupcích mezi vzorem M a obrazem S lze měřit neparametrickými statistickými testy. Velkou výhodou tohoto přístupu je, že o distribuci LBP v tomto případě není nutné činit žádné předpoklady. Také časová náročnost je ve srovnání s metodami klasifikace využívajícími učení s učitelem podstatně nižší, avšak neparametrická klasifikace není tak robustní.

Metoda G statistic

První navrženou statistickou metodou je tzv. G statistic [13]. Verze používaná pro klasifikaci je tvaru:

$$L(S, M) = - \sum_{b=1}^B S_b \log M_b \quad (4.1)$$

Čím vyšší je výsledná hodnota, tím jsou distribuce S a M podobnější.

Metoda χ^2

Tato metoda je vhodná i pro malé množství vzorků.

$$\chi^2(S, M) = \sum_{b=1}^B \frac{(S_b - M_b)^2}{S_b + M_b} \quad (4.2)$$

Zde naopak čím menší je výsledná hodnota, tím jsou distribuce S a M podobnější.

Jen o něco menší přesnosti lze dosáhnout pomocí průniku histogramů, ovšem se značně menšími výpočetními nároky:

$$H(S, M) = \sum_{b=1}^B \min(S_b, M_b) \quad (4.3)$$

V tomto případě opět čím vyšší je výsledná hodnota, tím jsou distribuce S a M podobnější.

4.1.2 Klasifikace pomocí umělé neuronové sítě

Jak bylo uvedeno, texturním příznakem je LBP histogram, který bude vstupem pro umělou neuronovou síť. Ta tedy bude mít tolik vstupních neuronů, kolik sloupců má histogram. Pro výstupní neurony jsou dvě možnosti:

1. Jednou sítí budeme klasifikovat všechny typy textur – pak tedy neuronová síť musí mít pro každý typ textur jeden výstup (případně ještě jeden výstup pro “neurčitou” klasifikaci).
2. Pro klasifikaci n typů textur použijeme n neuronových sítí, kdy každá z nich bude mít jen jeden výstup. Teoreticky tato varianta přináší časovou (doba učení a detekce) i prostorovou (počet použitých neuronů) úsporu, ovšem pro klasifikaci n typů textur je pak nutné nějakým způsobem “složit” jednotlivé dílčí klasifikace. Navíc přidání dalšího typu rozpoznávaných objektů je tak mnohem jednodušší.

Pro implementaci v rámci této práce (viz dále) byla zvolena druhá možnost.

4.2 Segmentace obrazu

V rámci segmentace musíme řešit vhodné dělení obrazu na regiony pro použití s LBP a také (vzhledem ke zvolenému způsobu klasifikace pomocí více neuronových sítí s jedním výstupem) segmentaci obrazu na základě několika dílčích klasifikací.

4.2.1 Dělení obrazu na regiony

Vzhledem k principu fungování LBP, kdy texturním příznakem je histogram vypočtený ze skupiny obrazových bodů, je třeba nějakým vhodným způsobem rozdělit segmentovaný obraz na regiony, které budeme klasifikovat zvlášť, a dílčí klasifikace pro tyto regiony pak opět spojit.

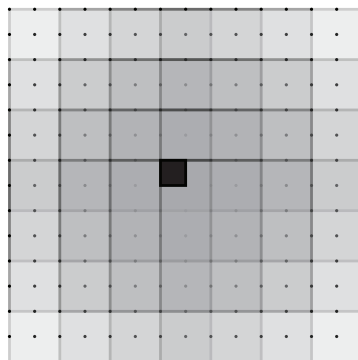
Vhodným řešením je rozdělení segmentovaného obrazu na pravidelná čtvercová okna. Každé okno pak klasifikujeme jako samostatný obraz a obrazovým bodům, které jsou součástí tohoto okna, přiřadíme klasifikaci z nadřazeného okna. Takto získáme poměrně hrubě segmentovaný obraz, kdy hranice klasifikovaných oblastí jsou pravoúhlé.

Volba velikosti okna

Důležitým parametrem je velikost okna. Čím menší okno bude, tím “jemnější” bude výsledný segmentovaný obraz, ale hrozí nebezpečí, že pro velmi malá okna bude klasifikace nepřesná nebo dokonce chybná, protože v takovém případě může být texturní příznak – LBP histogram – zkrácen malým počtem hodnot.

Překryv oken

Nevýhodu popsanou v předchozím oddíle odstraňuje metoda segmentace s překrýváním oken. Okna jsou klasifikována stejně jako v předchozím případě, ale vzájemně se překrývají, a jeden obrazový bod je tedy klasifikován vícekrát (viz obrázek 4.1). Klasifikace jednoho obrazového bodu je pak určena klasifikacemi všech oken, které tento obrazový bod obsahují.



Obrázek 4.1: Příklad oken velikosti 8 s překryvem 2. Vyznačený obrazový bod je tedy součástí celkem 16 oken a efektivní segmentace obrazu je po oknech velikosti 2.

Tento princip efektivně vede k tomu, že nejmenší segmentovaná oblast obrazu je také čtverec, ale o straně rovné vzdálenosti sousedních překrývajících se oken a nikoliv o straně rovné velikosti oken. Tak lze s pomocí (z hlediska extrakce texturního příznaku) bezpečně velkých oken dosáhnout poměrně "jemné" segmentace obrazu.

Určení absolutní klasifikace obrazového bodu při překryvu oken

Jelikož každý obrazový bod bude klasifikován několikrát, je třeba jeho celkovou klasifikaci určit tak, aby její hodnota byla vždy absolutní, tzn. nezávislá na velikosti a překryvu oken. Pro čtvercová okna velikosti w s překryvem s ($s \in \{1, 2, \dots, w\}$) je určena váha obrazového bodu m :

$$m = \begin{cases} \left(\frac{w}{s}\right)^2 & w \bmod s = 0 \\ \left(\frac{w+s-(w \bmod s)}{s}\right)^2 & \text{jinak,} \end{cases} \quad (4.4)$$

Absolutní hodnota klasifikace C_i obrazového bodu po i -tém průchodu přes daný obrazový bod je pak rovna:

$$C_i = C_{i-1} + \frac{C_{\max} \cdot c_w}{m} \quad (4.5)$$

kde C_{\max} je maximální přípustná absolutní hodnota klasifikace obrazového bodu, c_w je normalizovaná hodnota klasifikace pro aktuální okno ($c_w \in \langle 0, 1 \rangle$) a $C_0 = 0$.

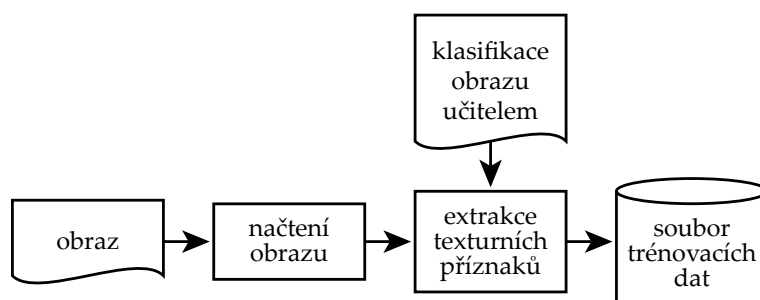
Jiné, méně přesné, avšak výpočetně rychlejší metody pro získání absolutní klasifikace obrazového bodu (např. plovoucí průměr) se v praxi ukázaly jako nevyhovující (zmíněný plovoucí průměr do klasifikovaného obrazu zanáší šum a vytváří artefakty).

4.3 Struktura systému pro detekci textur

Vytvořený systém sestává ze čtyř hlavních částí, z nichž každá se skládá z několika modulů:

1. systém pro extrakci trénovacích dat z obrazů
2. systém pro trénink umělé neuronové sítě na základě těchto dat
3. systém pro segmentaci obrazu za pomoci neuronové sítě (případně více sítí) získaných v předchozím kroku
4. podpůrné moduly pro zobrazení LBP, neparametrickou klasifikaci textur aj.

4.3.1 Extrakce trénovacích dat



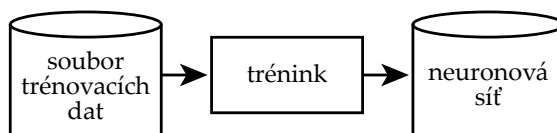
Obrázek 4.2: Extrakce trénovacích dat.

Struktura tohoto podsystému je na obrázku 4.2. Z obrazu načteného pomocí zvláštního modulu jsou modulem pro extrakci trénovacích dat získána trénovací data tak, že pro zdrojový obraz (nebo jeho část) je vypočten histogram uživatelem zvolené varianty LBP kódu. V tomto kroku musí uživatel přidat informaci o klasifikaci zpracovávaného obrazu, která je pak použita pro učení umělé neuronové sítě.

Kromě toho, že zdrojový obraz lze zpracovat jako celek, lze i nechat ze zdrojového obrazu náhodně generovat okna, která jsou pak zpracována jako samostatné obrazy. To je užitečné, pokud chceme z jednoho zdrojového obrazu získat větší množství trénovacích dat.

Výstupem této části systému jsou data pro učení neuronové sítě, která obsahují texturní příznaky (vstupy neuronové sítě) a jejich klasifikaci učitelem (výstupy neuronové sítě).

4.3.2 Trénink umělé neuronové sítě

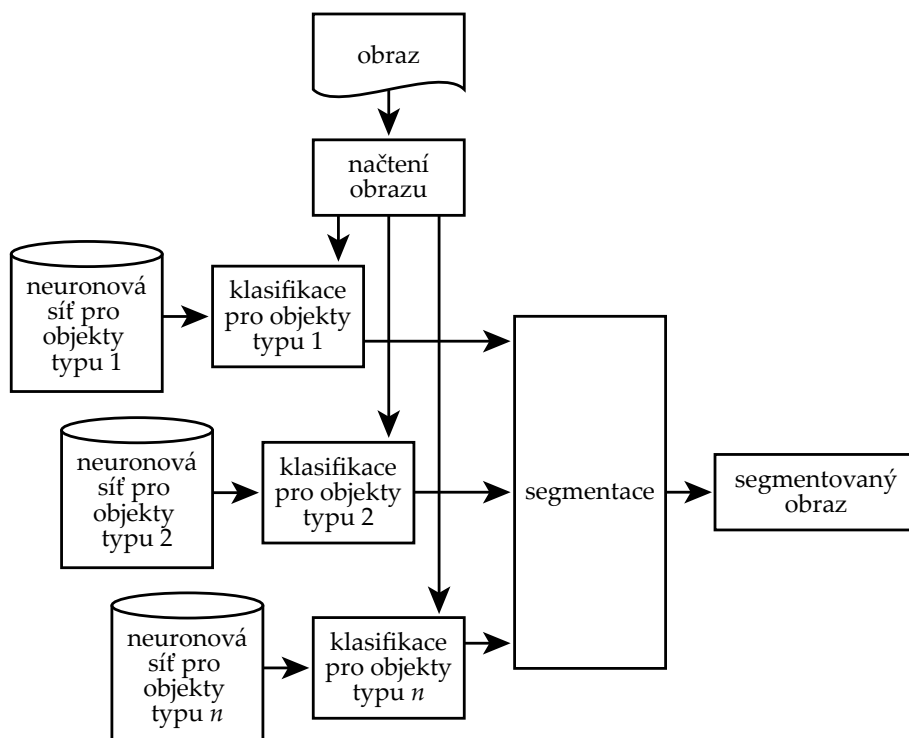


Obrázek 4.3: Trénink umělé neuronové sítě na základě trénovacích dat.

Struktura tohoto podsystemu (viz obrázek 4.3) je jednoduchá – data vytvořená v předchozím kroku jsou použita pro vytvoření umělé neuronové sítě za pomoci speciálního modulu. Spolu s vhodnými trénovacími daty je nastavení parametrů učení kritické pro úspěšnost následně vytvořené neuronové sítě. Lze zvolit chybu a maximální počet neuronů v síti.

Výstupem této části systému je neuronová síť naučená rozpoznávat zadaný typ objektů.

4.3.3 Segmentace obrazu pomocí umělé neuronové sítě



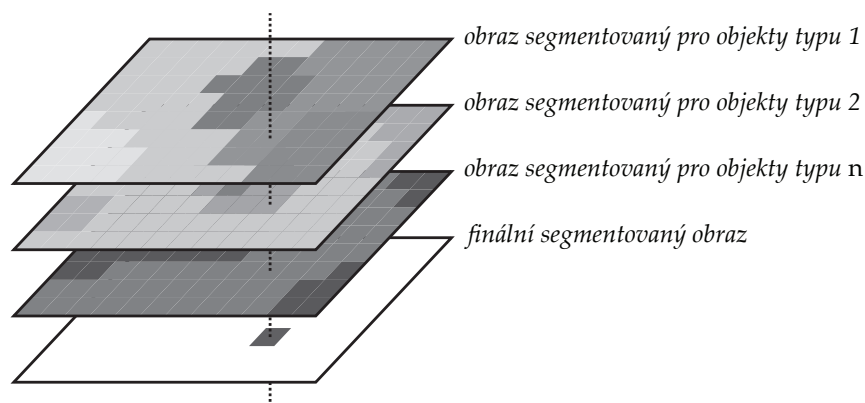
Obrázek 4.4: Segmentace obrazu s pomocí umělé neuronové sítě.

Pro klasifikaci každého typu objektů je použita zvláštní neuronová síť s jedním výstupem. Segmentace obrazu pro n typů objektů je tedy řešena tak, že zdrojový obraz je n -krát klasifikován, pokaždé zvláštní neuronovou sítí pro jeden typ objektu (viz obrázek 4.4). Těchto n klasifikací pak musíme spojit do finálního segmentovaného obrazu (viz obrázek 4.5).

Pro každý obrazový bod obrazu, který chceme segmentovat, máme vektor $\mathbf{A} = (a_1, \dots, a_n)$ intenzit obrazových bodů a_i na stejných souřadnicích získaných klasifikací z n klasifikátorů. To, do jakého segmentu zkoumaný obrazový bod zařadíme, je určeno:

$$r(\mathbf{A}) = \begin{cases} i_{\max} & \forall a_j; j \in \{1, 2, \dots, n\} - i_{\max} : a_j + t < a_{i_{\max}} \\ 0 & \text{jinak,} \end{cases} \quad (4.6)$$

kde i_{\max} je index prvku \mathbf{A} s ostře maximální intenzitou a t je práh segmentace (čím větší práh, tím větší musí být rozdíl mezi maximální intenzitou a každou další z vektoru \mathbf{A}). Segment obrazových bodů, které toto nesplňují, je označen jako neurčitý.



Obrázek 4.5: Skládání obrázků zvlášť klasifikovaných pro n typů objektů.

Pro segmentaci obrazu tímto způsobem je určen zvláštní modul, který je použitelný i samostatně. Vstupy tohoto modulu jsou obraz, který chceme klasifikovat, a neuronová síť získaná tréninkem v předchozím podsystému. Výstupem tohoto modulu je obraz, ve kterém obrazové body svojí intenzitou vyjadřují klasifikaci obrazových bodů původního obrazu podle dané neuronové sítě.

Výstupy klasifikace původního obrazu pro každý z hledaných typů objektů jsou pak spojeny v segmentačním modulu do jednoho výstupního obrazu, kde intenzity obrazových bodů vyjadřují příslušnost do jednotlivých segmentů obrazu.

4.4 Detekce objektů na leteckých snímcích podle textury

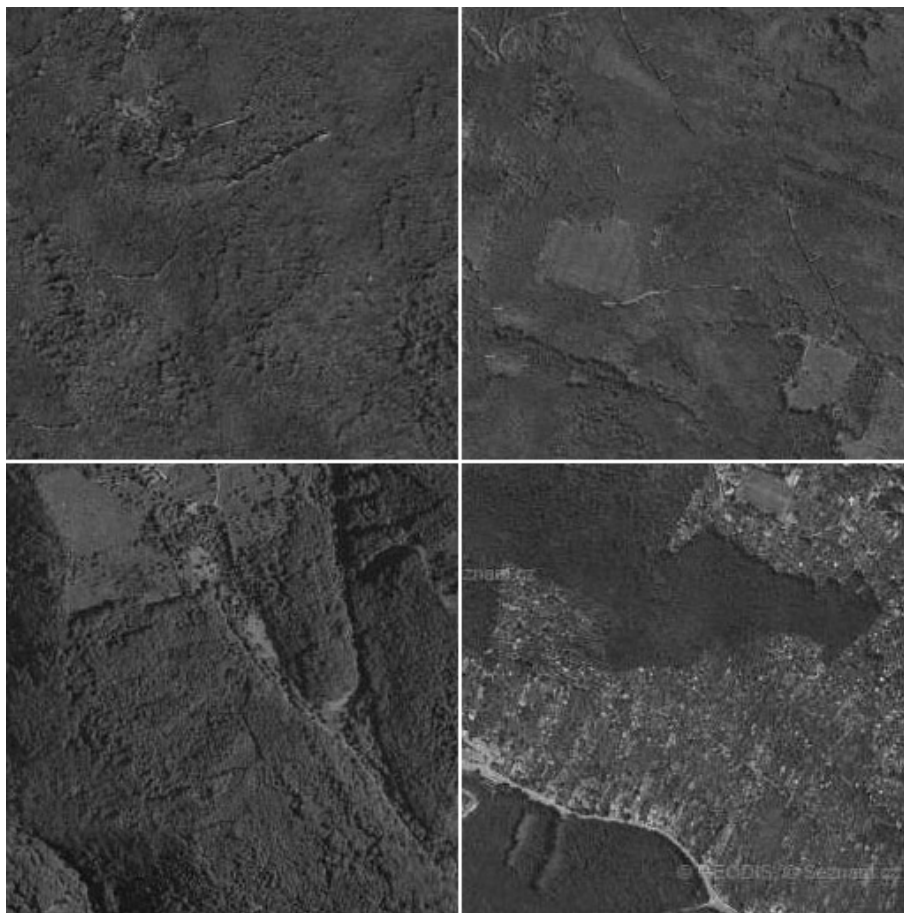
V této kapitole bude popsána specializace vytvořeného detektoru pro konkrétní oblast. Bude popsána volba a typy rozpoznávaných objektů a učení umělé neuronové sítě pro obrazová data obsahující tyto objekty.

4.4.1 Obrazová data a rozpoznávané objekty

Vnímání a rozpoznávání textur podstatně závisí na volbě měřítka (viz kapitolu 2.1), proto se také volba rozpoznávaných objektů odvíjí od zvoleného měřítka. Obrazová data byla převzata z [1] v měřítku přibližně 3.4 m na jeden obrazový bod. Jedná se o letecké snímky České republiky. Volba typů rozpoznávaných objektů proběhla na základě vizuálního porovnání, přičemž je potřeba si uvědomit, že implementovaný detektor pracuje pouze s intenzitou obrazu, ne s jeho jednotlivými barevnými složkami.

Lesní porosty

Textura lesních porostů je poměrně jednoduše odlišitelná od ostatních a ve srovnání s ostatními vykazuje také malou variabilitu (viz obrázek 4.6). Menší problémy mohou způsobovat stromy v zástavbě, například na zahradách nebo dvorech domů. Také mohou být problematické chatové nebo zahrádkářské kolonie (příklad je na posledním obrázku z řady), které je tedy vhodné explicitně zařadit buď mezi lesní porosty, nebo zástavbu.



Obrázek 4.6: Příklady textur různých lesních porostů

Zástavba

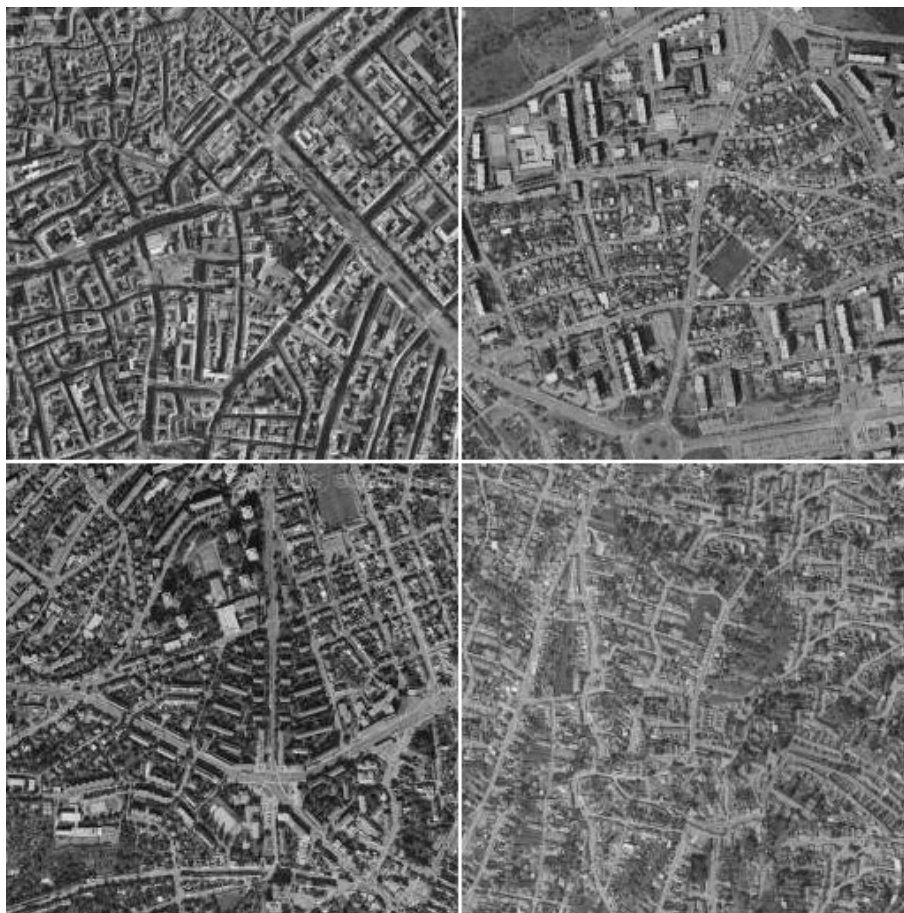
Jak je vidět z příkladů, textury, na kterých je obsažena zástavba, jsou velice variabilní (viz obrázek 4.7). Tato variabilita se odvíjí od hustoty osídlení (poměr budov a ostatních objektů, nejčastěji zeleně – např. vesnice vs. centra velkých měst), od stáří zástavby (např. budovy moderních sídlišť vs. historická sídla) i od urbanismu různých oblastí (“makrostruktura” historických osídlení s nepravidelným uspořádáním vs. sídliště stavěná “na zelené louce”).

Ačkoliv jsou tyto textury značně různorodé, stále jsou celkem jednoduše odlišitelné od ostatních vybraných typů textur (až na výše uvedené problémy při detekci lesních porostů). Stále se zvyšující variabilita zástavby (např. různé architektonické styly, vliv urbanismu atd.) klade velké nároky na obsažnost vzorových obrazových dat pro tento typ textur.

Zemědělské plochy

Pole jsou další významnou skupinou textur obsaženou na leteckých snímcích. Variabilita je zde značná, ovšem velmi výhodná je rotační invariance LBP (významným texturním prvkem jsou často viditelné “brázdy”, přirozeně vytvořené ve všech možných směrech – viz první příklad na obrázku 4.8).

Nevýhodou je častá “plochosť” textury, kdy textura pole neobsahuje žádné výraznější



Obrázek 4.7: Příklady textur různých druhů zástavby

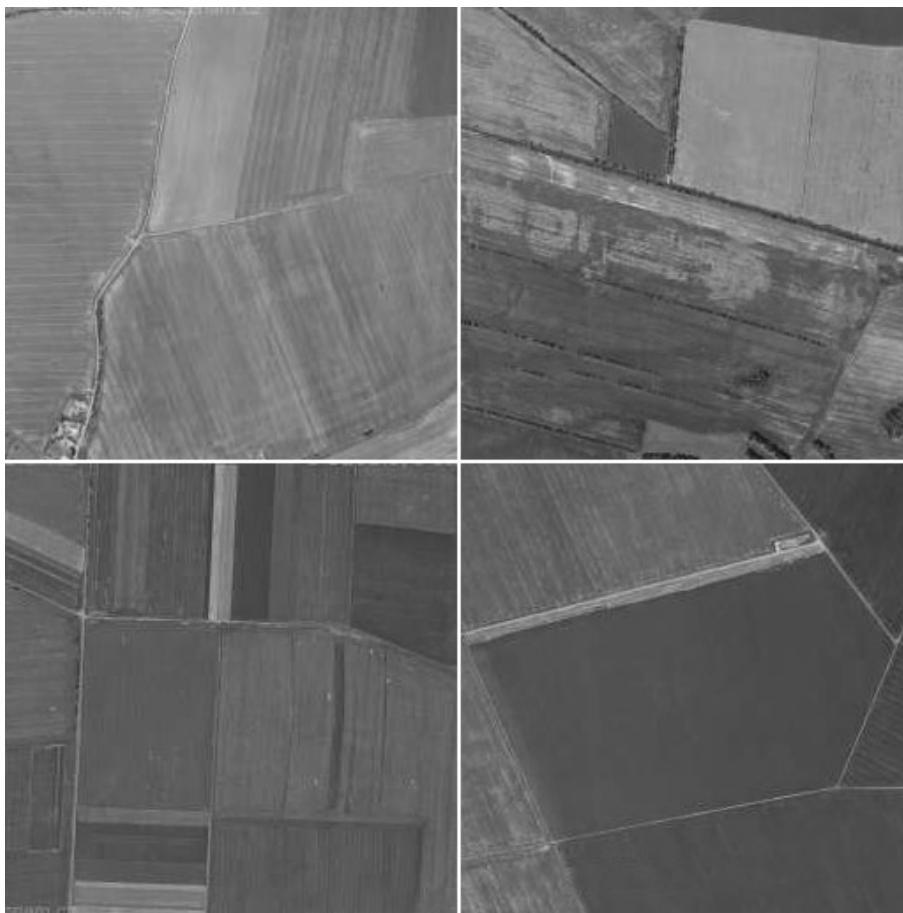
znaky, resp. viditelnou kresbu (zřetelné na posledním příkladu z obrázku 4.8). V takových případech může docházet k záměnám s texturou lesních porostů (zvláště v případech, kdy textura lesa je např. vlivem komprese obrazu příliš “vyhlazená”) nebo texturou zástavby (ploché střechy domů atp.).

Vodstvo

Na získaných obrazových datech nejsou tak velké vodní toky, aby se jejich textura mohla projevit vzhledem k minimální vhodné velikosti okna (viz první příklad na obrázku 4.9). Dalším problémem je, že velké vodní plochy (přehrady, jezera atp.) prakticky nemají texturu (viz druhý příklad na obrázku 4.9), což mimo jiné vede při rozpoznávání ke kolizím s některými podobně “plochými” texturami polí. Jelikož ale zastoupení zemědělských ploch v získaných obrazových datech několikanásobně převyšuje vodní plochy, nejsou vodní plochy (ve prospěch polí) rozpoznávány.

Další objekty

Na základě vizuální inspekce dostupných obrazových dat ze zdroje [1] lze tvrdit, že textury lesních porostů, zemědělských ploch a zástavby v těchto obrazových datech zcela



Obrázek 4.8: Příklady textur různých druhů zemědělských ploch

převažují.

Pro další možné typy objektů (např. průmyslové areály, lomy nebo povrchové doly) není v porovnání s výše uvedenými typy dostatek obrazových dat tak, aby umělá neuronová síť mohla na jejich základě získat požadovanou obecnost, resp. minimální chybu. Další typy objektů (např. silnice, železnice atp.) nejsou vzhledem k minimální velikosti okna natolik prostorově výrazné, aby se u nich dalo mluvit o textuře.

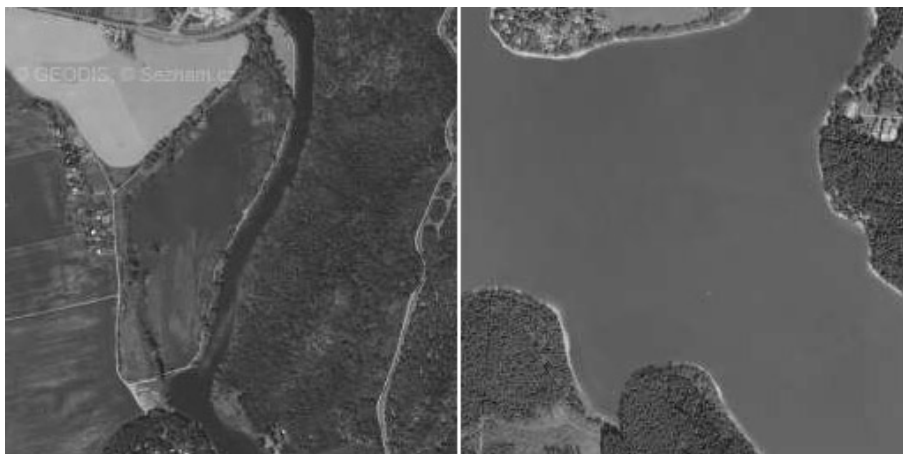
Rozpoznávané objekty

Rozpoznávanými objekty tedy z výše uvedených důvodů budou:

- všechny druhy zástavby
- zemědělské plochy
- lesní porosty

4.4.2 Trénovací data

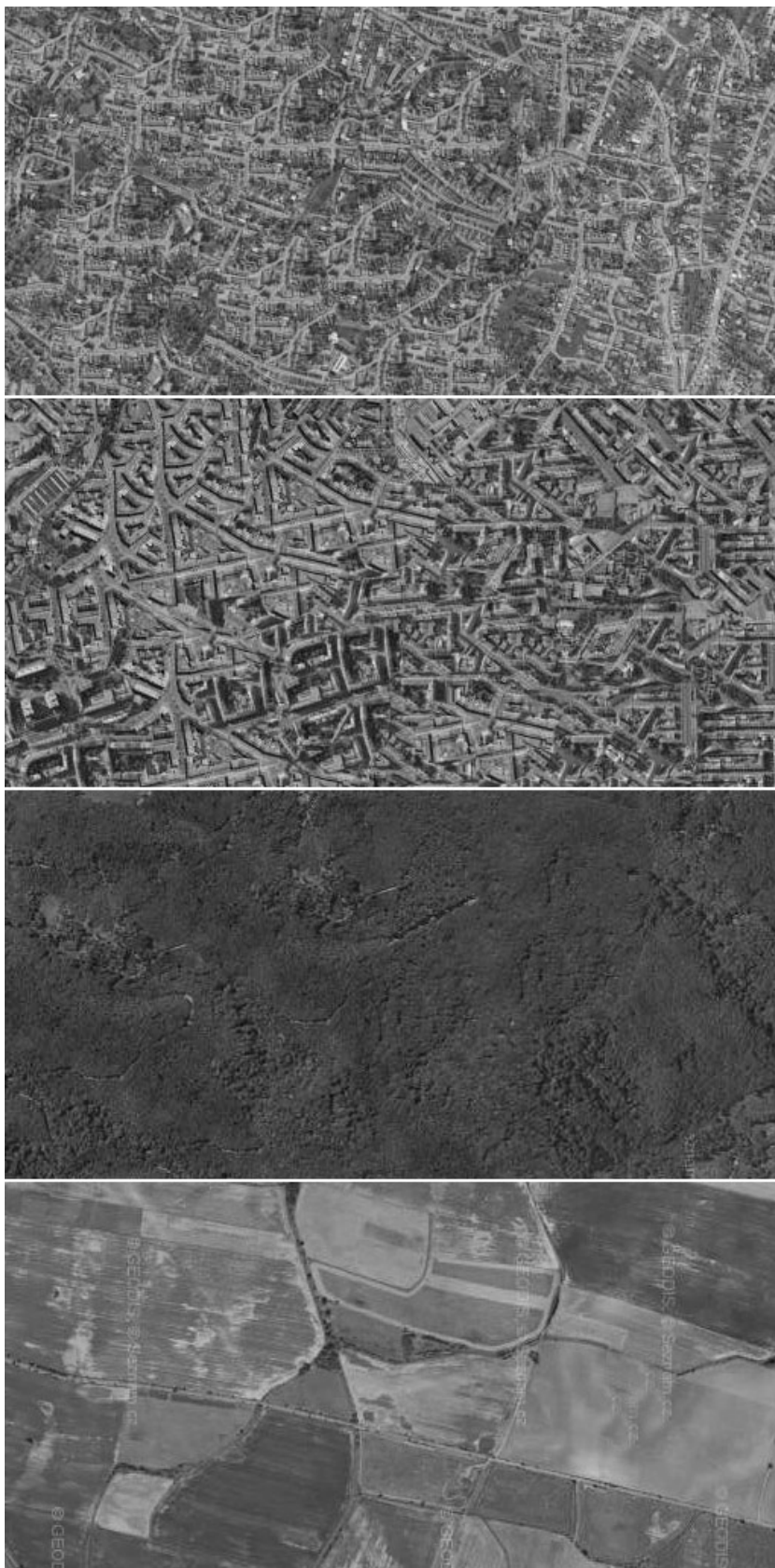
Trénovací data jsou generována ze zdroje [1]. Jsou nalezeny vhodné fragmenty odpovídající jednotlivým typům rozpoznávaných objektů. Ty jsou pak pro snazší generování trénovacích



Obrázek 4.9: Příklady různých vodních ploch

dat spojeny pomocí grafického editoru do jednoho uměle vytvořeného “bezešvého” obrazu (viz příklady na obrázku 4.10), který tak vypadá jako jediná celistvá textura.

Z tohoto uměle vytvořeného obrazu jsou pak pomocí příslušného modulu náhodně generována okna, ze kterých jsou extrahována trénovací data (tento proces je podrobně popsán v kapitole 4.3.1).



Obrázek 4.10: Ukázky trénovacích dat (výřezy)

Kapitola 5

Implementace detektoru textur a detekce objektů

Tato kapitola popisuje vlastní implementaci detektoru textur. Pro implementaci detektoru byl zvolen jazyk C/C++.

5.1 Nástroje pro implementaci

5.1.1 Toolkit MDSTk

Pro manipulaci s obrazy byl použit toolkit MDSTk [11] (*Medical Data Segmentation Toolkit*) vyvíjený na FIT VUT v Brně Ing. Michalem Španělem. MDSTk obsahuje knihovnu a moduly pro manipulaci s 2D a 3D obrazy zaměřené původně na segmentaci medicínských obrazových dat.

Z hlediska uživatele je systém složen z jednotlivých modulů, z nichž každý je zaměřen na určitý úkol (např. načítání obrazů v různých formátech, filtrování, segmentace, zobrazování aj.). Moduly lze několika způsoby spojovat a vytvářet tak složitější aplikace pro konkrétní potřeby uživatele.

V rámci této práce byla knihovna MDSTk doplněna o kód pro práci s LBP a další potřebné nástroje. Následně byly vytvořeny moduly umožňující klasifikaci a segmentaci obrazu na základě textur (viz dále).

Přehled implementovaných modulů a jejich parametrů je v dodatku B.

5.1.2 Knihovna FANN

FANN [5] (*Fast Artificial Neural Network Library*) je open source knihovna, která v jazyce C implementuje umělé neuronové sítě s různou topologií a obsahuje i nástroje pro práci s příslušnými daty. Tato knihovna byla zvolena z důvodu příznivé rychlosti a jednoduchosti použití.

Datové soubory generované jednotlivými moduly (např. data pro trénink neuronové sítě nebo soubory obsahující natrénovanou neuronovou síť) jsou právě ve formátu knihovny FANN.

5.2 Implementace jednotlivých částí detektoru textur

5.2.1 Výpočet a extrakce LBP

Toolkit MDSTk poskytuje nástroje pro manipulaci s obrazem a přístup k jednotlivým obrazovým bodům. Za pomoci těchto funkcí jsou implementovány funkce pro výpočet několika variant LBP, přičemž u všech lze libovolně volit hodnoty počtu vzorků P a poloměru R .

V rámci optimalizace rychlosti výpočtu je vhodné na základě vztahu 2.2 předem vypočítat relativní souřadnice všech okolních obrazových bodů v závislosti na hodnotách P a R (při řádově tisících až milionech výpočtů jednotlivých LBP pro jeden obraz je časově náročné pro každý obrazový bod počítat $2P$ hodnot s využitím goniometrických funkcí). Pro $P = 8$ a $R = 1$ je výpočet relativních souřadnic dále optimalizován přímo na body v 8-mi okolí středového bodu.

Implementovány jsou základní verze $LBP_{P,R}$, rotačně invariantní $LBP_{P,R}^{ri}$, uniformní $LBP_{P,R}^{u_2}$ a rotačně invariantní uniformní $LBP_{P,R}^{riu_2}$. Všechny tyto verze jsou implementovány tak, že je nejprve pro každý obrazový bod vypočten základní $LBP_{P,R}$ kód a hodnota z něj získaná je za pomoci vyhledávací tabulky mapována na zvolenou variantu LBP. Vyhledávací tabulka je zvolena opět z důvodu optimalizace rychlosti výpočtu, neboť rozšířené varianty LBP popsané v kapitolách 2.4.3, 2.4.4 a 2.4.5 jsou poměrně výrazně výpočetně náročnější než základní varianta.

Z principu LBP vyplývá, že pro body, které jsou okraji obrazu blíže než R obrazových bodů, nelze LBP vypočítat, jelikož k takovým bodům nelze získat kompletní okolí.

Hodnoty LBP získané z daného obrazu jsou ukládány do histogramu, jehož velikost (počet sloupců) závisí na zvoleném způsobu mapování. Tento histogram je pak před použitím pro klasifikaci normalizován dle vztahu 2.18.

5.2.2 Neparametrická klasifikace

Byly implementovány všechny způsoby neparametrické klasifikace textur popsané v kapitole 4.1.1. Zvláštní modul umožňuje porovnat dvojici obrazů pomocí zvoleného typu a mapování LBP. Všechny porovnávané histogramy jsou vždy normalizovány.

5.2.3 Klasifikace pomocí umělé neuronové sítě

Pro implementaci umělé neuronové sítě bylo využito nástrojů z knihovny FANN, přičemž síť využívá architekturu s kaskádovitým učením popsanou v kapitole 3.2. Pro problém analýzy textur s využitím LBP se tato architektura ukázala ve srovnání s klasickou topologií jako řádově rychlejší.

Jako kritéria pro ukončení učení lze zvolit maximální počet neuronů v síti a požadovanou chybu. Nastavení ostatních parametrů (maximální počet epoch stagnace v různých fázích učení atd.) na výchozí hodnoty poskytované knihovnou FANN se ukázalo jako vyhovující.

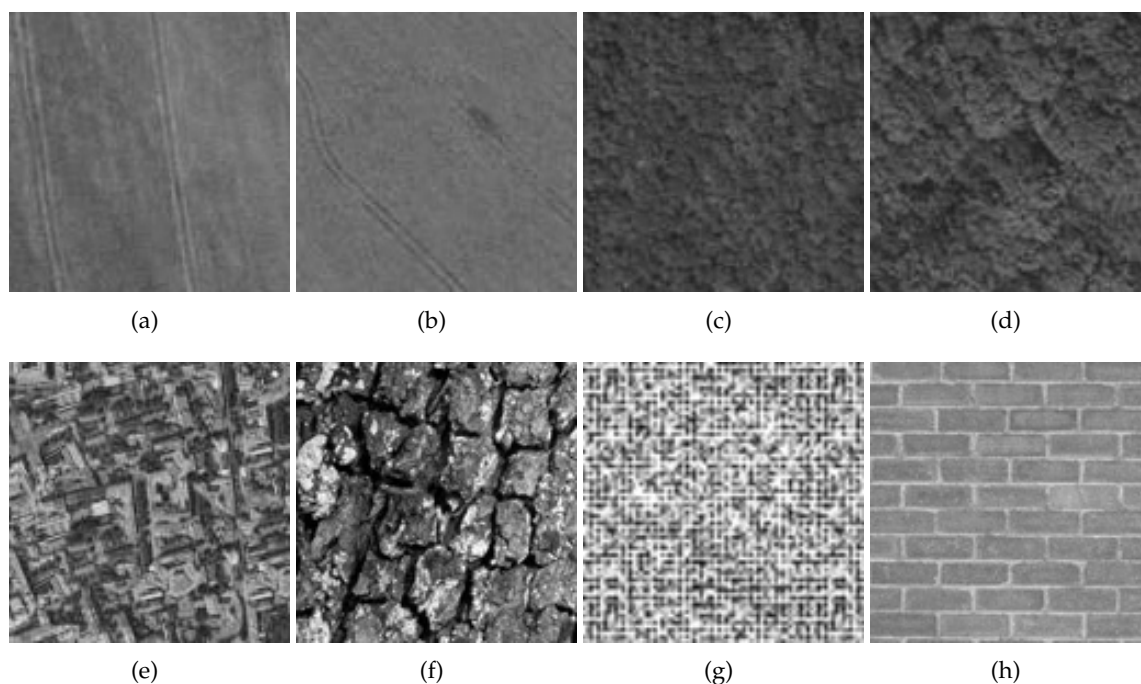
Vhodnou aktivační funkcí se pro skryté i výstupní neurony je hyperbolický tangens (3.1) s výchozí strmostí. Při klasifikaci příkladů učitelem ve fázi učení by tedy měly být klasifikované hodnoty v oboru hodnot této funkce, přičemž vhodné je využít celý rozsah oboru hodnot.

Kapitola 6

Výsledky

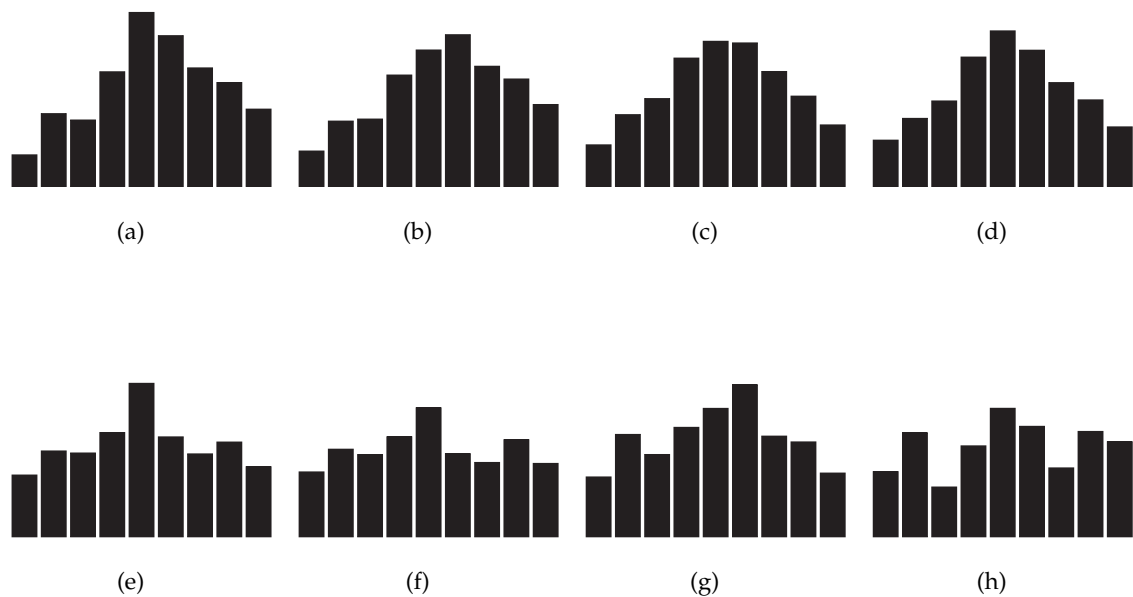
V této kapitole budou představeny zvolené parametry detektoru a následně výsledky práce detektoru textur na umělých i reálných obrazových datech.

6.1 Extrakce texturních příznaků



Obrázek 6.1: Různé textury použité v testech extrakce texturních příznaků a neparametrické klasifikace.

Na obrázku 6.2 jsou LBP_{8,1}^{riu₂} histogramy příslušné texturám z obrázku 6.1. I na základě vizuálního porovnání histogramů lze říci, že např. textury na obrázcích 6.1(c) a 6.1(d) jsou si podobnější než textury na obrázcích 6.1(g) a 6.1(h). LBP histogram sám o sobě má tedy poměrně velkou vypovídací hodnotu. Exaktní srovnání textur na základě těchto



Obrázek 6.2: Normalizované histogramy $LBP_{8,1}^{riu_2}$ pro příslušné příklady textur z obrázku 6.1.

histogramů nabízí následující kapitola.

6.2 Neparametrická klasifikace

Následuje ukázka praktických výsledků pro textury z obrázku 6.1. Textury na obrázcích 6.1(a) a 6.1(b), resp. 6.1(c) a 6.1(d) jsou vizuálně podobné, ostatní dvojice už ne. Tuto skutečnost by měly experimentální výsledky reflektovat.

	6.1(a)	6.1(c)	6.1(e)	6.1(f)
6.1(b)	0.008	0.009	0.033	0.057
6.1(d)	0.016	0.002	0.017	0.041
6.1(g)	0.024	0.012	0.022	0.045
6.1(h)	0.065	0.058	0.019	0.017

Tabulka 6.1: Neparametrická klasifikace – porovnání dvojic textur z obrázku 6.1. Výsledky pro statistický test χ^2 a operátor $LBP_{8,1.0}^{riu_2}$.

Tabulky 6.1, 6.2 a 6.3 obsahují výsledky porovnání všech dvojic textur z obrázku 6.1 pomocí všech statistických testů z kapitoly 4.1.1. Texturním operátorem byl v tomto případě $LBP_{8,1.0}^{riu_2}$.

Ve výsledcích si můžeme povšimnout několika jevů: Rozptyl hodnot χ^2 je největší ze všech implementovaných metod, což činí tento statistický test nejvíce "citlivým". Všechny statistické testy navíc správně vykazují shodnost dvou výše uvedených dvojic textur. Na

	6.1(a)	6.1(c)	6.1(e)	6.1(f)
6.1(b)	0.96	0.95	0.89	0.86
6.1(d)	0.92	0.97	0.92	0.87
6.1(g)	0.92	0.94	0.92	0.89
6.1(h)	0.84	0.84	0.92	0.93

Tabulka 6.2: Neparametrická klasifikace – porovnání dvojic textur z obrázku 6.1. Výsledky pro statistický test H a operátor $LBP_{8,1.0}^{riu_2}$.

	6.1(a)	6.1(c)	6.1(e)	6.1(f)
6.1(b)	0.96	0.97	0.97	0.99
6.1(d)	0.97	0.96	0.97	0.96
6.1(g)	0.97	0.97	0.97	0.96
6.1(h)	0.98	0.99	0.97	0.95

Tabulka 6.3: Neparametrická klasifikace – porovnání dvojic textur z obrázku 6.1. Výsledky pro statistický test L a operátor $LBP_{8,1.0}^{riu_2}$.

druhou stranu, všechny testy označí dvojici textur 6.1(b) a 6.1(c) jako podobnou, ačkoliv vizuálně tyto textury příliš podobné nejsou. Vizuálně nepřilíš shodné dvojice textur označí testy χ^2 i H jako nepodobné, ale test L v několika případech na rozdíl od ostatních selhává.

Z tohoto testu vyplývá, že pro neparametrickou klasifikaci jsou vhodné statistické testy χ^2 , případně H . Tyto testy většinou správně identifikují podobné textury, ovšem v některých případech jako podobné označí i vizuálně naprosto odlišné textury.

Následuje porovnání neparametrické klasifikace pro různé LBP operátory. Tabulky 6.4, 6.5 a 6.6 shrnují výsledky porovnání vybraných dvojic textur z obrázku 6.1.

		$LBP_{8,1.0}^{riu_2}$	$LBP_{12,2.5}^{riu_2}$	$LBP_{16,4.0}^{riu_2}$
6.1(a) × 6.1(b)		0.008	0.026	0.031
6.1(b) × 6.1(c)		0.009	0.012	0.010
6.1(f) × 6.1(g)		0.045	0.075	0.087
6.1(a) × 6.1(h)		0.065	0.029	0.020

Tabulka 6.4: Neparametrická klasifikace – porovnání vybraných dvojic textur z obrázku 6.1. Výsledky pro statistický test χ^2 a různé parametry LBP operátorů.

Lze pozorovat značný vliv měřítka textury na porovnání textur. U nejperspektivnějšího testu (χ^2) se projevuje nepravidelné kolísání hodnot. Rozptyl hodnot se zároveň se zvětšujícím se počtem vzorků a poloměrem prohlubuje, avšak ne vždy správným směrem (první a poslední případ v tabulce). Větší poloměr a počet vzorků tedy v porovnání se základní variantou nepřinesly zásadní zpřesnění výsledků, navíc základní varianta je díky menšímu počtu vzorků méně výpočetně náročná. Nadále tedy budeme pracovat s $LBP_{8,1}^{riu_2}$.

	$LBP_{8,1.0}^{riu_2}$	$LBP_{12,2.5}^{riu_2}$	$LBP_{16,4.0}^{riu_2}$
6.1(a) × 6.1(b)	0.96	0.92	0.93
6.1(b) × 6.1(c)	0.95	0.94	0.96
6.1(f) × 6.1(g)	0.89	0.87	0.87
6.1(a) × 6.1(h)	0.84	0.91	0.94

Tabulka 6.5: Neparametrická klasifikace – porovnání vybraných dvojic textur z obrázku 6.1. Výsledky pro statistický test H a různé parametry LBP operátorů.

	$LBP_{8,1.0}^{riu_2}$	$LBP_{12,2.5}^{riu_2}$	$LBP_{16,4.0}^{riu_2}$
6.1(a) × 6.1(b)	0.96	0.90	0.69
6.1(b) × 6.1(c)	0.97	0.95	0.75
6.1(f) × 6.1(g)	1.00	0.86	0.63
6.1(a) × 6.1(h)	0.98	0.88	0.71

Tabulka 6.6: Neparametrická klasifikace – porovnání vybraných dvojic textur z obrázku 6.1. Výsledky pro statistický test L a různé parametry LBP operátorů.

6.3 Klasifikace pomocí umělé neuronové sítě

6.3.1 Volba LBP operátoru

Zvoleným LBP operátorem je varianta $LBP_{8,1}^{riu_2}$. Vhodnou velikostí okna je pro tuto variantu 32 obrazových bodů. Při této velikosti již nehrozí nebezpečí malého (zkreslujícího) počtu vzorků a zároveň je tato velikost ještě přijatelně malá v poměru k velikosti rozpoznávaných objektů v daném měřítku.

6.3.2 Vlastnosti a objem trénovacích dat

Experimentálně bylo zjištěno, že vhodný počet trénovacích dat (tj. jednotlivých oken) pro každý typ rozpoznávaných objektů je (s ohledem na dobu učení, velikost sítě a úspěšnost detekce) v řádu jednotek tisíců. Umělá neuronová síť pak dosahuje velikosti řádově stovek neuronů (vzhledem k použité architektuře sítě je v každé skryté vrstvě právě jeden neuron) a dosažená chyba se pohybuje v řádech menších než 0.0001, aniž by se projevilo přetrénování sítě. Počet trénovacích epoch je v řádech desítek tisíc a maximální počet epoch stagnace učení je výchozí hodnota poskytovaná knihovnou FANN, tj. 12.

Bylo pozorováno, že důraz na obsažnost trénovacích dat (tj. zahrnutí pokud možno co nejvíce různorodých příkladů textur) je v tomto případě důležitější než snaha o minimalizaci chyby nebo dosažení určité velikosti sítě – jinými slovy, nejlepším způsobem, jak zvýšit úspěšnost detektoru textur, je pokrytí co největší variability textur v trénovacích datech, jelikož různorodost klasifikovaných obrazových dat je značná.

6.4 Výstupy detektoru textur

V této kapitole budou prezentovány výsledky práce detektoru textur na konkrétních obrazových datech.

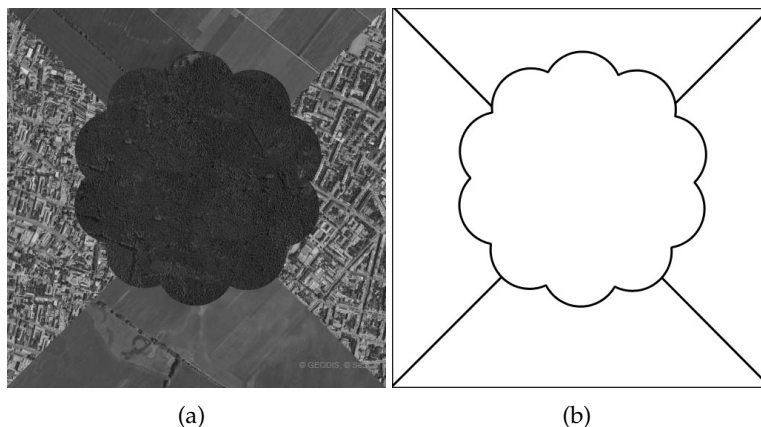
6.4.1 Klasifikace

Následuje ukázka klasifikace provedená pomocí umělé neuronové sítě s parametry popsanými v předcházející kapitole. Jelikož pro klasifikaci n typů objektů máme n neuronových sítí, v této kapitole budeme prezentovat jejich výstupy zvlášť a následující kapitola se bude věnovat spojení těchto výstupů a finální segmentaci leteckého snímku podle textur.

Zásadní otázkou při klasifikaci je volba velikosti oken a jejich překryvu (viz kapitolu 4.2). Volba okna by měla přibližně korespondovat s velikostí okna použitého při extrakci trénovacích dat (aby nedocházelo ke zkreslení vlivem příliš rozdílných počtů vzorků v LBP histogramech), ovšem překryv oken zásadně ovlivňuje výsledné “rozlišení” klasifikovaného obrazu (a na druhou stranu také dobu výpočtu).

Test klasifikace na umělém obrazu

Klasifikaci otestujeme nejprve na umělém obrazu, tj. obrazu sestaveném v grafickém editoru z textur tak, že známe jejich hranice, a můžeme je tedy přesně konfrontovat s výsledkem klasifikace.



Obrázek 6.3: Uměle vytvořený testovací obraz a hranice textur v něm.

Velikost překryvu oken je v této práci “volná proměnná”, jelikož není zadán přímo účel detekce objektů na leteckých snímcích (jiné rozlišení vyžaduje prosté zjištění přítomnosti nějakého objektu na snímku a jiné např. segmentace za účelem vytvoření mapy). Proto bude pro každý obraz vždy prezentováno několik výsledků klasifikace s měnící se velikostí překryvu oken.

Jak bylo popsáno v kapitole 4.3.3, výstupem každého klasifikátoru je obraz, v němž body svojí intenzitou vyjadřují svoji klasifikaci (bod s minimální intenzitou, tj. černý, znamená, že textura v tomto bodě vůbec neodpovídá textuře z trénovacích dat, pro bílou naopak). Jelikož praktické výstupy z klasifikátorů jsou v nezpracované podobě vizuálně

nepříliš vypovídající (hlavně při malém překryvu oken), jsou klasifikované obrazy pro větší názornost obarvený pomocí prostředků toolkitu MDSTk¹.

Na obrázku 6.4 je klasifikace umělého testovacího obrazu pro všechny tři typy rozpoznávaných objektů s různými velikostmi překryvu oken. Vidíme, že hranice textur přibližně odpovídají, pro malé hodnoty překryvu oken jsou hranice přesnější. Je ovšem vhodnější klasifikované obrazy použít jako mezistupeň v segmentaci, než samostatně – v takovém případě by bylo lepší klasifikovaný obraz prahovat, abychom dostali jasnější výsledek.

6.4.2 Segmentace

Důležitou volbou je nastavení prahu segmentace, který určuje, jak velký musí být vzájemný rozdíl mezi jednotlivými klasifikovanými obrazovými body, kterým chceme přiřadit nějaký segment obrazu. Pomocí nastavení prahu² můžeme určovat míru (ne)jistoty segmentace – pokud použijeme malý práh, stačí relativně malé rozdíly k tomu, aby byl segment přiřazen. Tak získáme obraz s relativně malým počtem obrazových bodů v segmentu označeném jako “nejasný” – na druhou stranu, některé body mohou být segmentovány nesprávně. Pro velký práh je situace opačná.

Přirozeně by se měly body označené jako “nejasné” vyskytovat hlavně v místech přechodu textur nebo na jejich hranicích. Velké oblasti takových bodů na jiných místech při běžném (tj. nijak extrémním) nastavení prahu znamenají, že na takových místech selhává klasifikace – jinak řečeno, v takových místech je více druhů textur klasifikováno s přibližně stejnou silou, což není žádoucí. Vhodným řešením takové situace je zlepšení fungování klasifikátorů, nejlépe pomocí přidání problematických textur do sady trénovacích dat a nového trénování sítě.

Test segmentace na umělém obrazu

Ve čtvrtém sloupci obrázku 6.4 je zobrazena finální segmentace testovacího obrázku 6.3(a). Práh je nastaven na hodnotu 256, která odpovídá $\frac{1}{16}$ celkového rozsahu intenzity obrazových bodů.

Vidíme, že segmentovaný obraz relativně dobře odpovídá skutečným hranicím textur z obrázku 6.3(b), přičemž výsledek segmentace se přirozeně zlepšuje se zmenšující se velikostí překryvu okna. V prvních dvou řádcích se objevuje poměrně velký počet chybně segmentovaných regionů – řešením je přirozené zjemnění segmentace tak, jak je vidět v dalších řádcích výsledků.

Obrazové body náležející do “nejasného” segmentu pozorujeme v drtivé většině na hranicích segmentovaných oblastí, což je správné chování. Segmentace je více chybová hlavně na okrajích obrázku, což je ale u použité metody normální jednak z důvodu principu fungování texturního operátoru (viz kapitolu 5.2.1) a jednak proto, že na okrajích obrázku není pro jeden obrazový bod tolik překrývajících se oken, jako “uvnitř” obrazu. Ve zvoleném řešení jsou okrajové části obrázku, které tedy z principu nemohou být přesně segmentovány, vždy ponechány.

¹Nutno podotknout, že toto kolorování do obrazu nezanášá žádnou novou informaci, ale slouží pouze k vizuálnímu zvýraznění.

²Práh se zadává jako minimální požadovaný rozdíl intenzit klasifikovaných obrazových bodů (absolutně). V implementovaném programu je intenzita obrazového bodu celé číslo v rozsahu (0, 4096).

Segmentace reálných obrazů

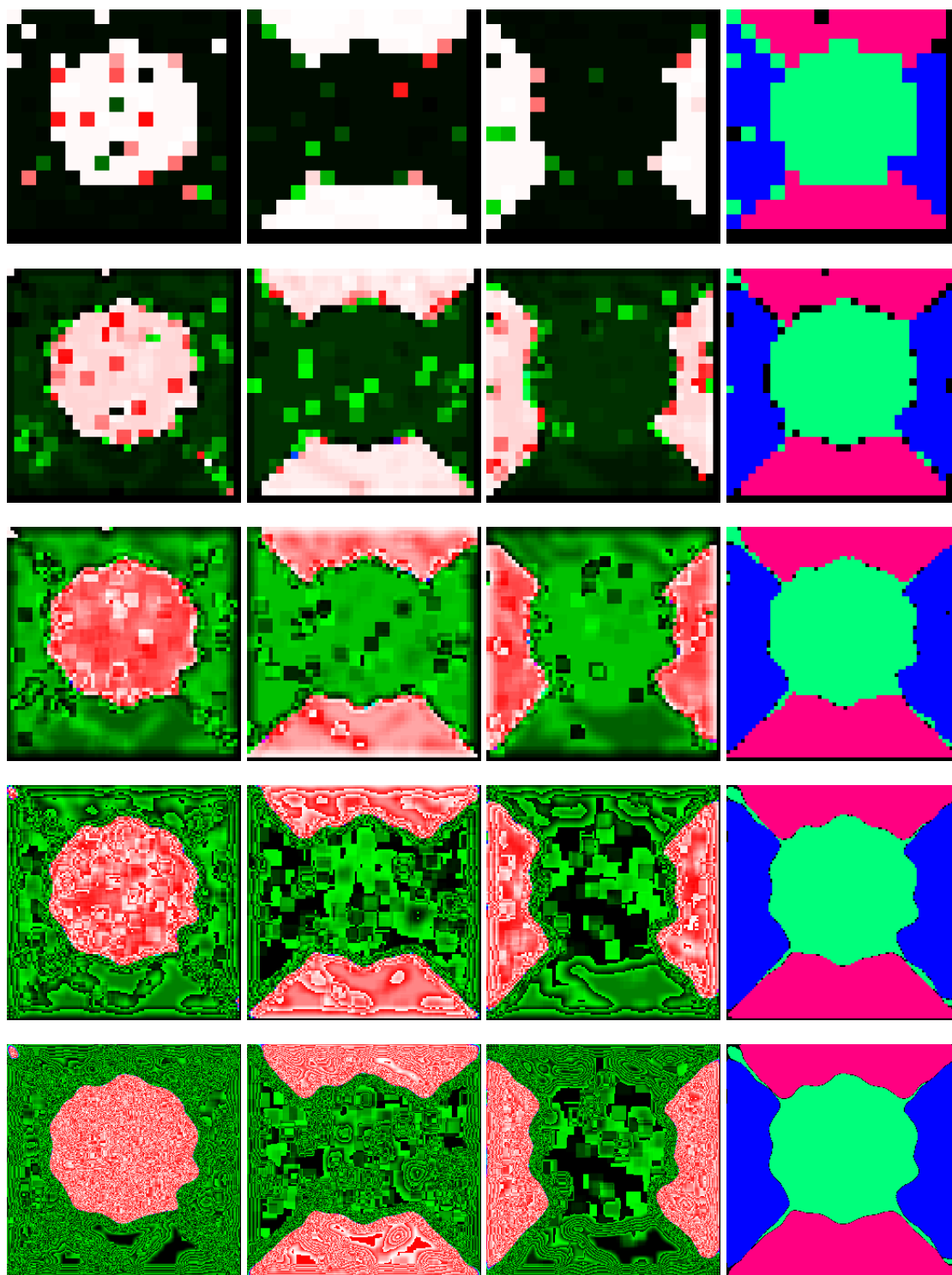
V této kapitole budou prezentovány praktické výsledky segmentace obrazu na reálných obrazových datech. Je zobrazen vždy původní obraz (vlevo) a původní obraz překrytý poloprůhledným segmentovaným obrazem (vpravo) kvůli snazšímu porovnání. Modrá barva značí segment zástavby, červená zemědělské plochy a zelená lesní porosty. Černá, která se objevuje sporadicky, znamená, že pro dané obrazové body nebylo možné jednoznačně rozhodnout, do kterého segmentu patří.

Klasifikace a segmentace je provedena se stejným nastavením jako při testu na umělém obrazu v předchozí kapitole. Vzdálenost překryvu oken je rovna dvěma obrazovým bodům.

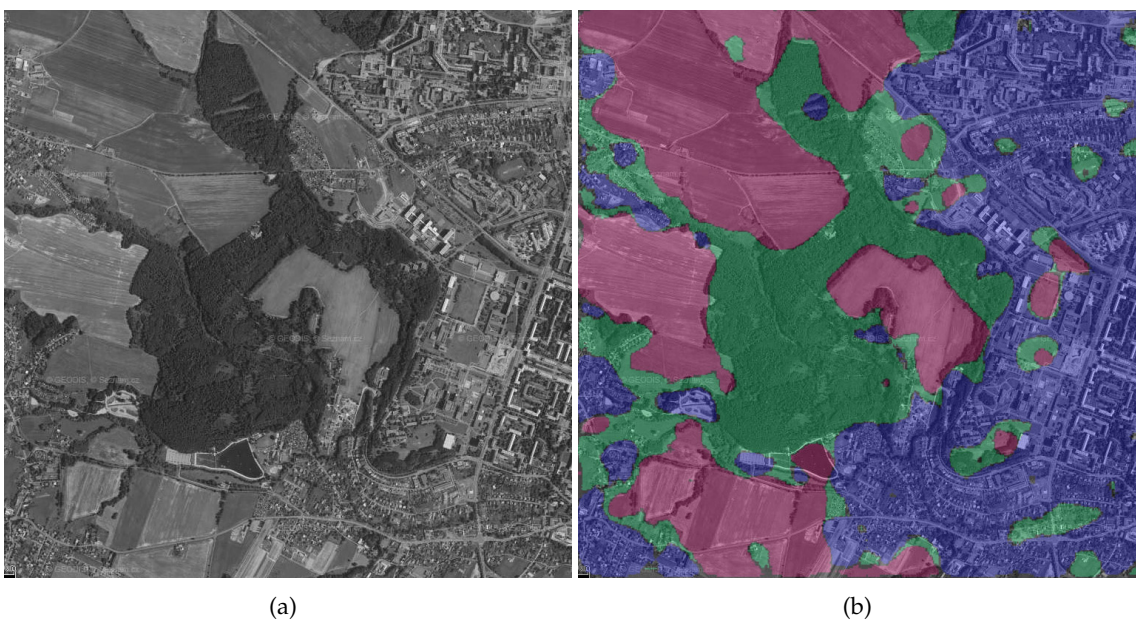
Ukázka segmentace reálného obrazu je na obrázku 6.5, další se nachází v dodatku A. Vidíme, že detekce zástavby je v drtivé většině případů úspěšná. Na obrázku A.3(b) byla jako zástavba klasifikována i oblast obsahující dopravní komunikaci – ovšem to je v pořádku, jelikož dopravní infrastruktura není samostatně rozpoznávána a v trénovacích datech je zahrnuta právě v zástavbě. Oblast s řídkou zástavbou může být také občas klasifikována jako les.

Je vidět, že odlišení lesních porostů a zemědělských ploch je problematické zejména tam, kde textura neobsahuje žádnou “kresbu”. Naopak tam, kde mají zejména zemědělské plochy texturu výraznější (např. na obrázku 6.5(b)), detekce funguje dobře. Na úspěšnost rozpoznání má také vliv velikost rozpoznávaných objektů vzhledem k velikosti okna (konkrétně například meze nebo remízky jsou ve většině příkazů zanedbány).

Celkově detekce funguje velmi dobře tam, kde se v obrazu nalézají dostatečně zřetelná (ale nikoliv z hlediska kontrastu, ale spíše z hlediska přítomnosti jisté struktury) textura. “Prázdnou” texturu můžeme z principu zařadit do trénovacích dat pouze pro jednu skupinu rozpoznávaných objektů, což pak vede k popisovaným záměnám.



Obrázek 6.4: Klasifikace a segmentace umělého testovacího obrazu z obrázku 6.3. V prvních třech sloupcích jsou klasifikace obrazu postupně pro lesní porosty, zemědělské plochy a zástavbu, v posledním sloupci je výsledný segmentovaný obraz. Velikost oken je ve všech případech 32 obrazových bodů a jejich překryv je postupně shora dolů 32, 16, 8, 4 a 2 obrazové body.



Obrázek 6.5: Reálný testovací obraz a výsledek jeho segmentace. Další výsledky segmentace jsou zobrazeny v dodatku A.

Kapitola 7

Závěr

Cílem této práce bylo prozkoumat způsoby klasifikace textur a za pomoci vybrané metody implementovat detektor textur, který bude sloužit k detekci objektů na leteckých snímcích. Tento cíl se podařilo splnit. Těžiště práce je v oblasti počítačového vidění, přičemž problematika klasifikace textur má teoretické východisko v metodách umělé inteligence.

Pro extrakci texturních příznaků byla zvolena metoda *Local binary patterns* a některé její modifikace, které lze na základě praktických výsledků hodnotit jako velmi perspektivní. Dále byly prezentovány dva způsoby klasifikace textur. Přístup založený na neparametrické klasifikaci je výpočetně poměrně nenáročný, i když ne zcela robustní. Velkou výhodou ovšem v tomto případě je, že před vlastním porovnáním není třeba mít žádné informace – proto se tento způsob hodí zejména pro orientační porovnávání textur.

Druhým způsobem je klasifikace pomocí umělé neuronové sítě. Tato metoda se ukazuje jako velmi robustní, i když její úspěšnost přímo závisí na trénovacích datech a parametrech sítě, které je ne vždy snadné zvolit optimálně. Pro klasifikaci každého typu objektů je použita jedna neuronová síť.

Detektor textur byl implementován v jazyce C++ za pomoci toolkitu MDSTk. Byla vytvořena řada modulů, jejichž spojováním lze vytvořit složitější systém. Kromě toho byly do knihovny MDSTk přidány (příslušně dokumentované) třídy a funkce pro práci s LBP, které lze v rámci toolkitu dále využít.

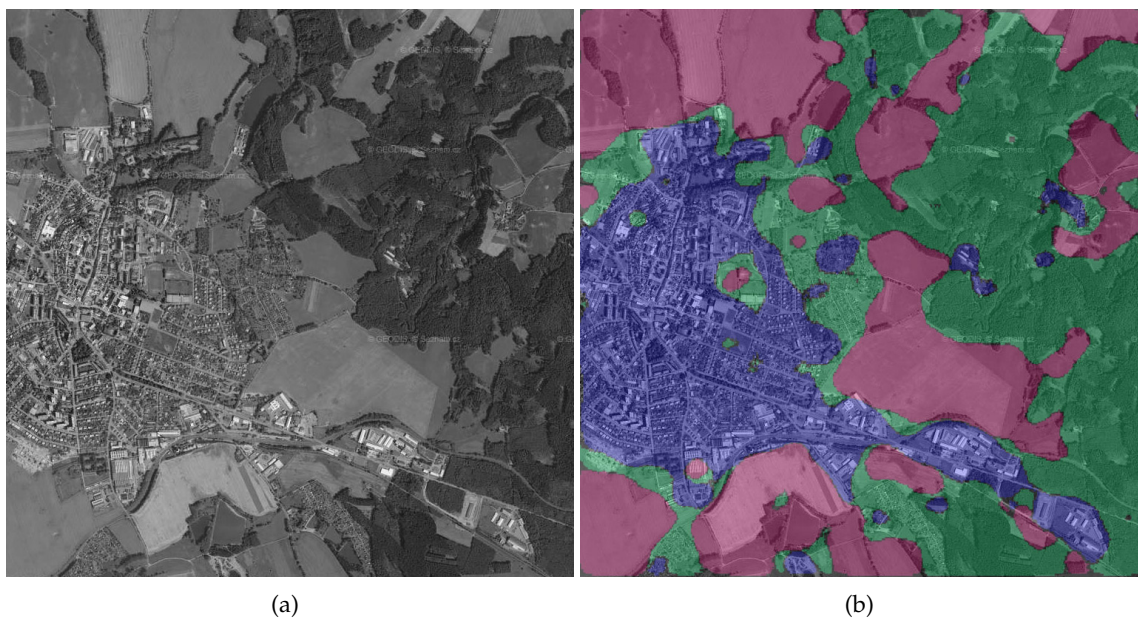
Nakonec byly zvoleny rozpoznávané typy objektů a byly prezentovány praktické výsledky segmentace. Ačkoliv segmentace není absolutně přesná (a vzhledem k téměř neomezené diverzitě textur vyskytujících se v reálu ani absolutně přesná být nemůže), jsou dosažené výsledky více než uspokojivé, zvláště když uvážíme relativní nenáročnost metody.

Možných rozšíření této práce je celá řada – samozřejmě lze snadno změnit specializaci detektoru na jiné textury (pomocí změny trénovacích dat) a stejně jednoduše lze i měnit počet rozpoznávaných objektů s ohledem na segmentaci (to vše na uživatelské úrovni). Co se týče programové části práce, zajímavou oblastí pro další zkoumání jsou modifikace LBP, kterých je celá řada a které mají velmi zajímavé vlastnosti. Dalším tématem k rozšíření práce je vlastní způsob klasifikace textur na základě texturních příznaků.

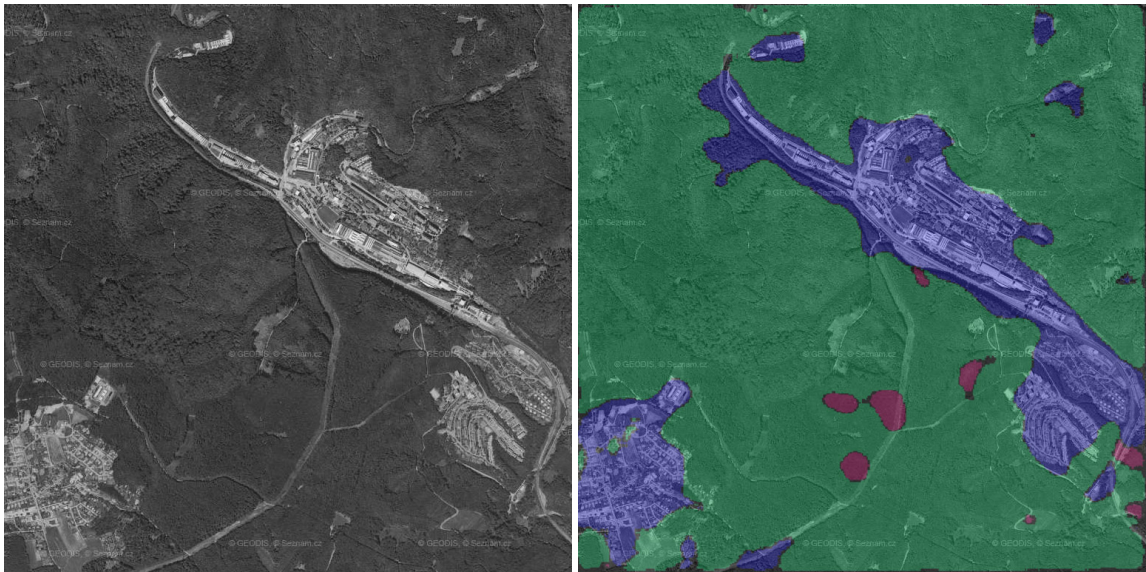
Celkově lze říci, že požadavky zadání bakalářské práce se podařilo splnit úspěšně. Výsledky produkované implementovaným detektorem textur jsou i přes drobné nedostatky dobré, přičemž existuje hned několik cest, jak detektor dále vylepšit. Vedlejším (ale nikoliv zanedbatelným) efektem této práce je navíc rozšíření funkčnosti toolkitu MDSTk (z hlediska běžného uživatele i programátora) o další nástroje.

Dodatek A

Ukázka segmentace reálných obrazových dat



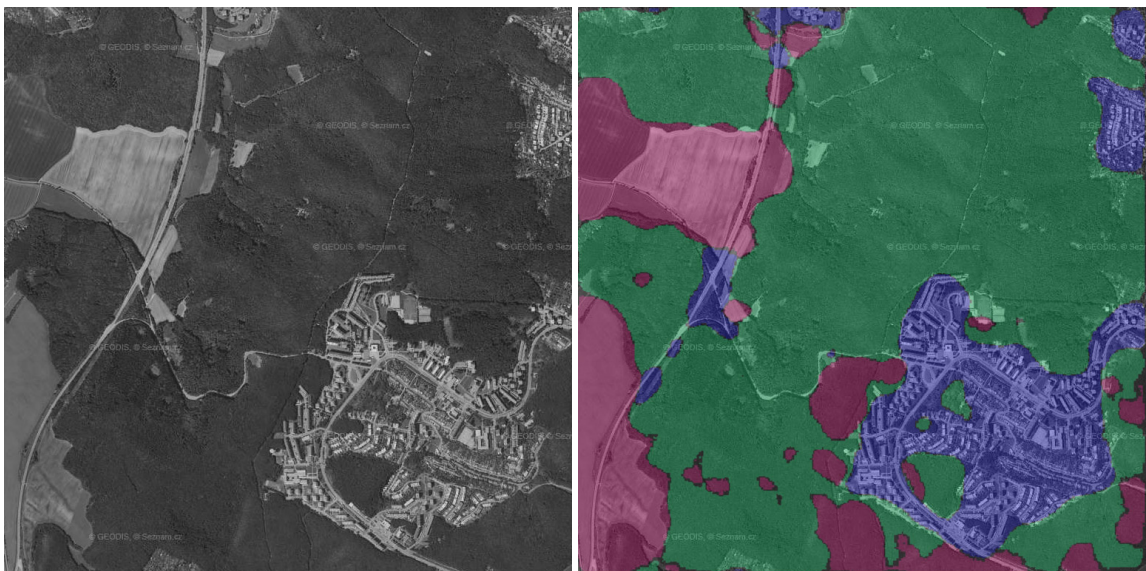
Obrázek A.1: Reálný testovací obraz a výsledek jeho segmentace.



(a)

(b)

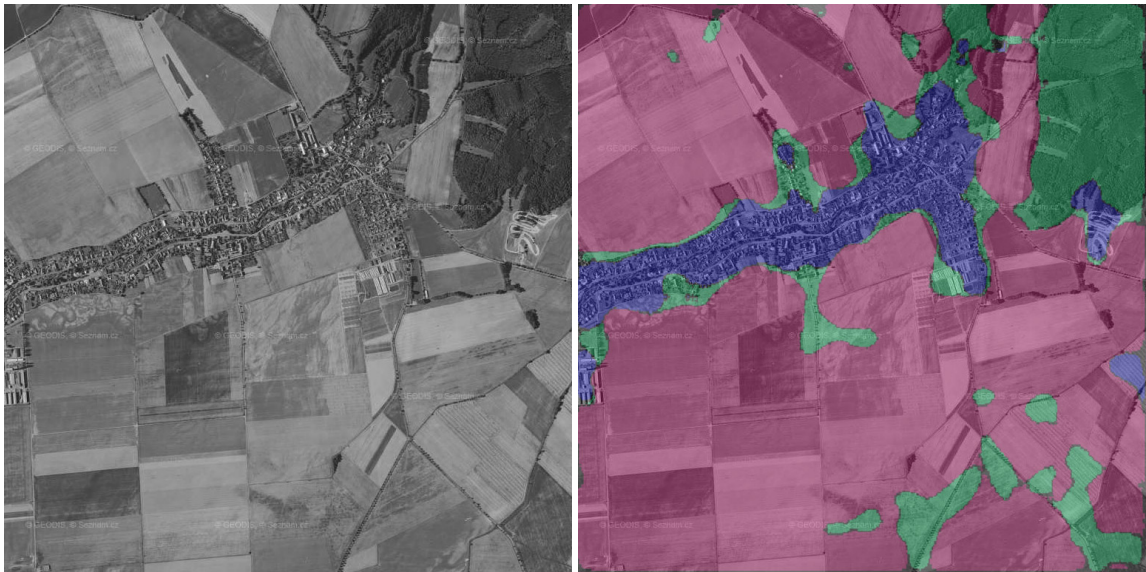
Obrázek A.2: Reálný testovací obraz a výsledek jeho segmentace.



(a)

(b)

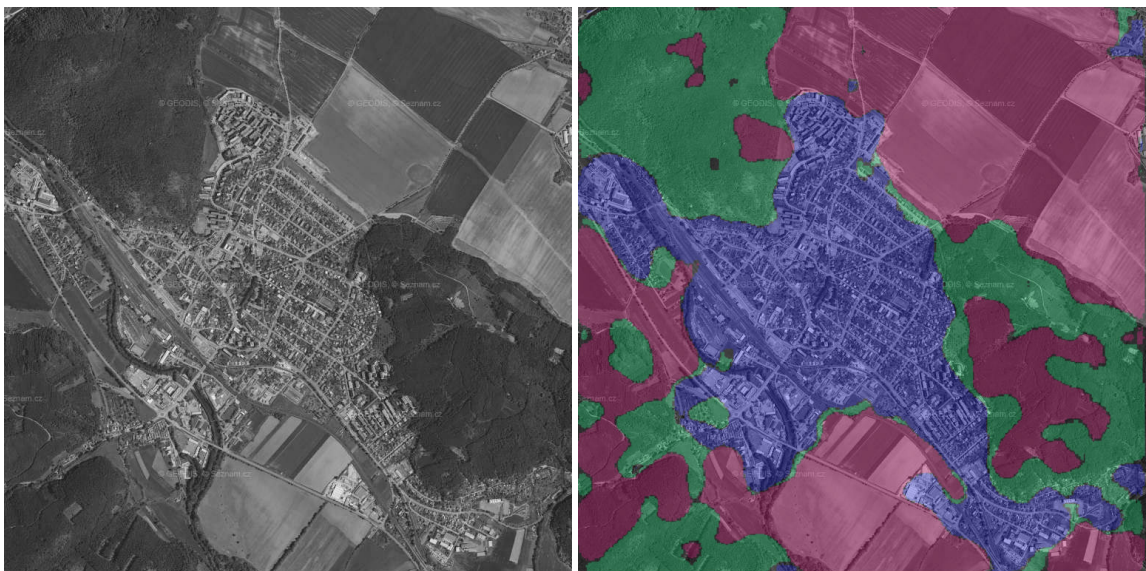
Obrázek A.3: Reálný testovací obraz a výsledek jeho segmentace.



(a)

(b)

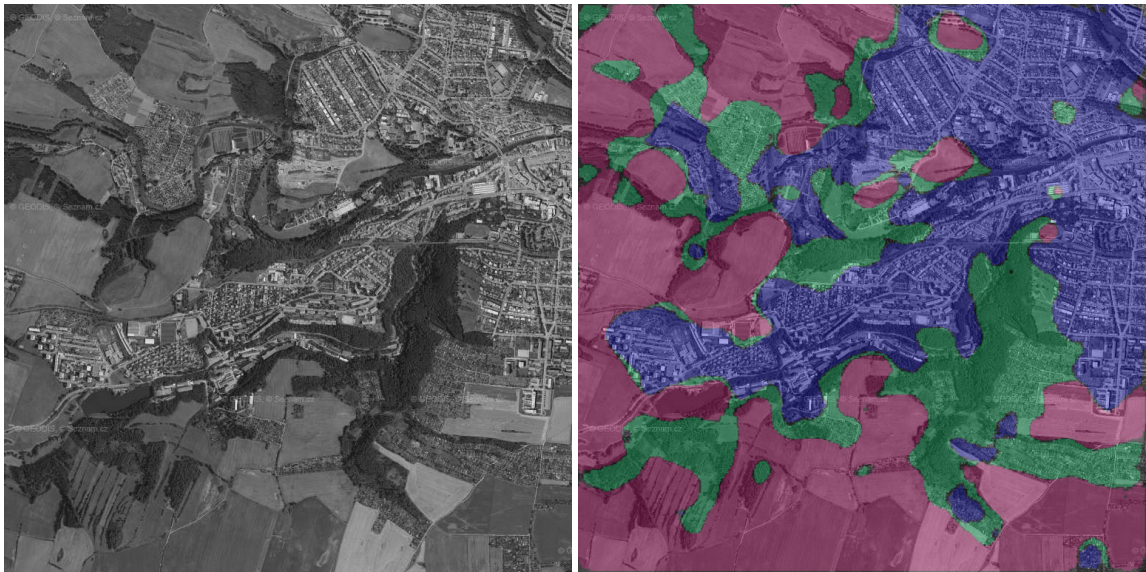
Obrázek A.4: Reálný testovací obraz a výsledek jeho segmentace.



(a)

(b)

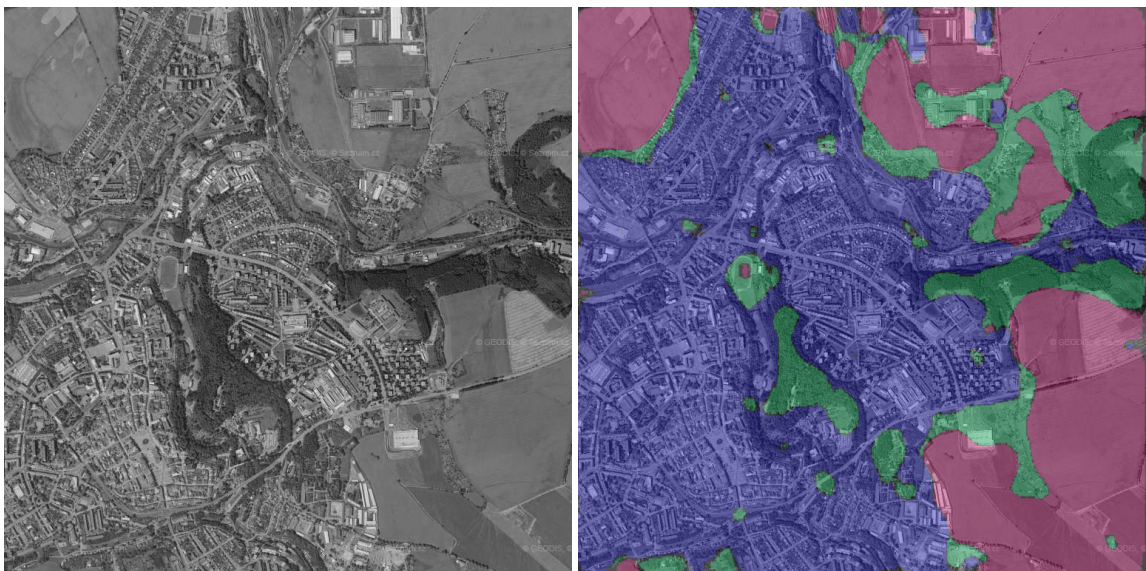
Obrázek A.5: Reálný testovací obraz a výsledek jeho segmentace.



(a)

(b)

Obrázek A.6: Reálný testovací obraz a výsledek jeho segmentace.



(a)

(b)

Obrázek A.7: Reálný testovací obraz a výsledek jeho segmentace.

Dodatek B

Příručka uživatele pro nově implementované moduly

Jelikož všechny nově implementované moduly jsou součástí toolkitu MDSTk, jejich instalace a způsob použití jsou shodné s ostatními moduly toolkitu a lze je tedy nalézt v dokumentaci MDSTk. Zde bude pouze stručně popsán název, účel a parametry každého nového modulu.

`mdsSliceLBPHistogram`

Vstup: Obraz MDSTk typu `slice`.

Výstup: LBP histogram vstupního obrazu dle parametrů modulu.

Parametry:

- `[-m]` Typ LBP mapování. Možné hodnoty:
 - `p` Žádné.
 - `ri` Rotačně-invariantní LBP.
 - `u` Uniformní LBP.
 - `riu2` Rotačně-invariantní uniformní (2) LBP (výchozí).
- `[-s]` Počet bodů v okolí středového obrazového bodu (hodnota P). Výchozí hodnota: 8.
- `[-r]` Vzdálenost okolních bodů od středového obrazového bodu (hodnota R). Výchozí hodnota: 1.0
- `[-p]` Formátování výstupního histogramu. Možné hodnoty:
 - `g` Pseudografické (sloupce kresleny pomocí ASCII znaků). Výchozí.
 - `c` Index sloupce a jeho hodnota ve formátu CSV.

`mdsSliceLBPCompare`

Vstup: Dva obrazy MDSTk typu `slice`.

Výstup: Číselná hodnota vyjadřující podobnost zadaných obrazů na základě jejich textury. Využívá neparametrické metody klasifikace.

Parametry:

- `[-m]` Stejně jako u modulu `mdsSliceLBPHistogram`.
- `[-s]` Stejně jako u modulu `mdsSliceLBPHistogram`.

- [-r] Stejné jako u modulu `mdsSliceLBPHistogram`.
- [-c] Metoda klasifikace. Možné hodnoty:
 - `chi` Metoda χ^2 . Nejpřesnější (výchozí).
 - `intersct` Průnik histogramů.
 - `gstat` *G-statistic*.
 - `all` Všechny předchozí metody s popisem.

`mdsSliceLBPEXtract`

Vstup: Obraz MDSTk typu `slice`.

Výstup: Nový (nebo modifikovaný stávající) datový soubor FANN obsahující trénovací data pro umělou neuronovou síť extrahovaná ze zadaného souboru.

- Parametry:*
- f Soubor, do kterého se budou ukládat trénovací data.
 - c Klasifikace vstupního souboru uživatelem pro učení umělé neuronové sítě (výchozí: 1.0).
 - [-m] Stejné jako u modulu `mdsSliceLBPHistogram`.
 - [-s] Stejné jako u modulu `mdsSliceLBPHistogram`.
 - [-r] Stejné jako u modulu `mdsSliceLBPHistogram`.
 - [-d] Pokud je zadáno a soubor se jménem zadaným pomocí parametru `-f` již existuje, jeho obsah bude přepsán (v opačném případě se nová data přidají na jeho konec).
 - [-w] Pokud je zadáno, jsou ze vstupního obrazu náhodně generována čtvercová okna a každé je zpracováno jako samostatný obraz. Hodnota parametru je šířka takto generovaných oken.
 - [-n] Počet náhodně generovaných oken (výchozí: 16). Bere se v potaz pouze, pokud je nastaven parametr `w`.

`mdsANNTrain`

Vstup: Neení.

Výstup: Neení.

- Parametry:*
- t Soubor s trénovacími daty pro umělou neuronovou síť.
 - [-e] Požadovaná chyba učení. Výchozí: 0.001.
 - [-m] Maximální počet neuronů v síti. Výchozí: 500.
 - [-b] Počet přidávaných neuronů mezi po sobě následujícími výpisy. Výchozí: 1000.

`mdsSliceLBANNClassify`

Vstup: Obraz MDSTk typu `slice`.

Výstup: Obraz MDSTk typu `slice`, ve kterém body svoji intenzitou vyjadřují svoji klasifikaci podle neuronové sítě.

- Parametry:*
- f Soubor obsahující natrénovanou neuronovou síť.

- [-m] Stejně jako u modulu `mdsSliceLBPHistogram`.
- [-s] Stejně jako u modulu `mdsSliceLBPHistogram`.
- [-r] Stejně jako u modulu `mdsSliceLBPHistogram`.
- [-w] Velikost oken, které se klasifikují zvlášť (jako jeden region).
Výchozí: 64 obrazových bodů.
- [-p] Překryv oken. Výchozí: 64 obrazových bodů.

`mdsSliceLBPSegment`

- Vstup:* Alespoň 2 obrázky MDSTk typu `slice`, každý klasifikovaný pro jeden typ objektů.
- Výstup:* Obraz MDSTk typu `slice` segmentovaný na základě vstupních obrazů.
- Parametry:* [-t] Práh segmentace. Pokud se intenzita obrazových bodů na odpovídajících souřadnicích neliší alespoň o práh, je segment označen jako neurčitý. Výchozí: 512.

Literatura

- [1] *Mapy.cz*. [Online], [cit. 30. 3. 2008].
URL <http://www.mapy.cz>
- [2] FAHLMAN, S. E.; LEBIERE, C.: *The Cascade-Correlation Learning Architecture*. School of Computer Science, Carnegie Mellon University, 1991.
- [3] KRŠEK, P.: *Základy počítačové grafiky – Studijní opora*. FIT VUT v Brně, 2007.
- [4] MÄENPÄÄ, T.: *The local binary pattern approach to texture analysis – extensions and applications*. Infotech Oulu and Department of Electrical and Information Engineering, University of Oulu, 2003, ISBN 951-42-7076-2.
- [5] NISSEN, S.: *Fast Artificial Neural Network Library (FANN)*. [Online], [cit. 12. 2. 2008].
URL <http://leenissen.dk/fann/>
- [6] OJALA, T.: *A comparative study of texture measures with classification based on feature distributions*. Pattern Recognition, 1996: s. 51–59.
- [7] OJALA, T.; PIETIKÄINEN, M.: *Nonparametric multichannel texture description with simple spatial operators*. Proc. 14th International Conference on Pattern Recognition, Brisbane, Australia, 1998: s. 1052–1056.
- [8] OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T.: *Multi-resolution gray-scale and rotation invariant texture classification with Local binary patterns*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002: s. 971–987.
- [9] OJALA, T.; PIETIKÄINEN, M.; XU, Z.: *Rotation-invariant texture classification using feature distributions*. Pattern Recognition, 2000: s. 43–52.
- [10] ŠPANĚL, M.: *Počítačové vidění – Analýza a extrakce příznaků z textur (prezentace přednášek)*. FIT VUT v Brně, 2007.
- [11] ŠPANĚL, M.: *Medical Data Segmentation Toolkit*. [Online], [cit. 2. 3. 2008].
URL <http://www.fit.vutbr.cz/~spanel/mdstk/>
- [12] PRECHELT, L.: *Investigation of the CasCor Family of Learning Algorithms*. Fakultät für Informatik, Universität Karlsruhe, 1996.
- [13] SOKAL, R.; ROHLF, F.: *Biometry*. W. H. Freeman, 1969.
- [14] TURPINEN, M.: *Learning and recognizing texture characteristics using local binary patterns*. Faculty of Technology, Department of Electrical and Information Engineering, Infotech Oulu, University of Oulu, 2007, ISBN 978-951-42-8502-8.