

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## PODPORA TVORBY MAP POMOCÍ METOD ZPRACOVÁNÍ OBRAZU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JÁN JAROŠ

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## PODPORA TVORBY MAP POMOCÍ METOD ZPRACOVÁNÍ OBRAZU

SUPPORT OF MAPPING BY IMAGE PROCESSING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JÁN JAROŠ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN VÁŇA

BRNO 2012

## Abstrakt

Tato bakalářská práce se zabývá metodami detekce vybraných objektů ve videu a importováním těchto objektů do centrální databáze OpenStreetMap na základě jejich geografické polohy. Zaměřena je z velké části na rozpoznávání dopravních značek. První část stručně popisuje některé nejpoužívanější metody a samotný projekt OpenStreetMap. V následujících kapitolách je uveden podrobnější přehled použitých metod navrhnutého systému, jeho implementace a testování. Závěr obsahuje zhodnocení celé práce a jsou zde uvedené možné rozšíření.

## Abstract

This bachelor's thesis deals with methods of detection of selected objects in video and with importing these objects into OpenStreetMap central database based on their geographic location. It focuses mainly on recognition of road signs. First section briefly describes some of the most widely used methods and OpenStreetMap project itself. In the following chapters is given a more detailed overview of used methods of proposed system, its implementation and testing. The conclusion contains evaluation of whole work and the possible improvements are listed here.

## Klíčová slova

Detekce dopravních značek, Rozpoznávání dopravních značek, OpenCV, Podpůrné vektory, Histogram orientovaných gradientů, JOSM, OpenStreetMap

## Keywords

Road sign detection, Road sign recognition, OpenCV, Support Vector Machine, Histogram of oriented gradients, JOSM, OpenStreetMap

## Citace

Ján Jaroš: Podpora tvorby map pomocí metod zpracování obrazu, bakalářská práce, Brno, FIT VUT v Brně, 2012

# Podpora tvorby map pomocí metod zpracování obrazu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jana Váňu. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Ján Jaroš  
13. května 2012

## Poděkování

Poděkovať by som sa chcel predovšetkým vedúcemu, pánu Ing. Janu Váňovi za pozitívny prístup a cenné rady pri písaní práce. Ďalej Ing. Davidovi Hermanovi za poskytnutie videozáznamov, Ing. Tomášovi Novotnému za pomoc pri výbere témy bakalárskej práce a môjmu bratovi Braňovi za asistovanie pri testovaní.

© Ján Jaroš, 2012.

*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

<b>Úvod</b>	<b>2</b>
<b>1 Teoretický rozbor</b>	<b>3</b>
1.1 OpenStreetMap . . . . .	3
1.2 Vhodné objekty pre mapovanie . . . . .	3
1.3 Základná metodológia . . . . .	4
1.4 Farebné modely . . . . .	4
1.5 Segmentácia . . . . .	6
1.6 Detekcia tvarov . . . . .	8
1.7 Extrakcia príznakov . . . . .	8
1.8 Rozpoznávanie . . . . .	10
<b>2 Návrh riešenia</b>	<b>12</b>
2.1 Popis prvkov navrhnutého systému . . . . .	12
2.2 Získanie vstupných záznamov . . . . .	13
2.3 Množina detekovaných objektov . . . . .	13
2.4 Návrh Aplikácie . . . . .	14
2.5 Návrh Zásuvného modulu . . . . .	18
<b>3 Implementácia</b>	<b>20</b>
3.1 Aplikácia pre rozpoznávanie dopravných značiek . . . . .	20
3.2 Zásuvný modul . . . . .	22
<b>4 Testovanie</b>	<b>23</b>
4.1 Porovnanie segmentačných metód . . . . .	23
4.2 Zisťovanie optimálnych prahov pre HSV . . . . .	25
4.3 Testovanie aplikácie . . . . .	25
4.4 Zhodnotenie dosiahnutých výsledkov . . . . .	26
4.5 Import a uloženie dát na server OSM . . . . .	28
<b>Záver</b>	<b>30</b>
<b>A Zoznam podporovaných značiek</b>	<b>34</b>
<b>B Výsledky testov</b>	<b>36</b>
<b>C Ukážky aplikácie a pluginu</b>	<b>39</b>
<b>D Obsah priloženého DVD</b>	<b>40</b>

# Úvod

V dnešnej dobe sú technológie natoľko vyspelé, že stroje v niektorých činnostiach úplne nahradili človeka. Súčasné počítače dokážu riešiť vysoko náročné úlohy, pričom ich výkon sa neustále zvyšuje a ich rozmery sa znižujú. Tieto zariadenia nás niekoľko tisícnásobne, ak nie viac, prevyšujú v schopnostiach riešiť zložité matematické a logické operácie, či uchovávať veľké množstvo údajov v pamäti.

Výhodou ľudí však ešte stále je ich vnímanie okolitého sveta, schopnosť myslieť a učiť sa novým veciam. Vďaka našim zmyslom sme schopní zachytávať, spracovávať a vyhodnocovať rôzne zvuky, obrazy, chute, či pachy. Počítače aj napriek svojmu výkonu zaostávajú. Keďže sa však stali neoddeliteľnou súčasťou nášho života, prenikajú krok po kroku aj do týchto oblastí. Aj keď doba, kedy ich naučíme rozoznávať vône a chute je asi ešte ďaleko, pri snahe napodobniť ostatné zmysly tomu tak nie je.

Už dnes existujú inteligentné systémy, ktoré analyzujú teplotu, tlak a predpovedajú nám počasie. Digitálne fotoaparáty dokážu v obraze detekovať tvár človeka, automatické rozpoznávanie ŠPZ automobilov uľahčuje prácu policajným zložkám a niektoré systémy sú schopné analyzovať ľudskú reč a to dokonca v najrôznejších svetových jazykoch.

Oblasť, ktorou sa zaoberám v tejto bakalárskej práci je práve spracovanie obrazu. Cieľom je navrhnúť a vytvoriť systém schopný nájsť vo videu objekty, ktoré by sa dali využiť pri tvorbe máp. Zameral som sa na dopravné značky, pretože majú pomerne veľkú informatívnu hodnotu a pomáhajú zvyšovať bezpečnosť cestnej premávky.

Značky zavedené do mapy môžu byť užitočné aj pri plánovaní trasy, napríklad nákladných automobilov. Tie musia spĺňať isté dopravné obmedzenia ako je maximálna výška, šírka, alebo hmotnosť vozidla kvôli konštrukčným limitom mostov, či tunelov. Keďže mapa do ktorej sa majú značky zaviesť je v digitálnom formáte, môžu tieto dáta slúžiť aj na iné účely, ako je napríklad navigácia.

V práci vychádzam z niekoľkých štúdií formulovaných osobami, ktoré sa touto problematikou už zaoberali a ktorí navrhli mnoho spôsobov rozpoznávania dopravných značiek. Prehľad týchto metód je uvedený v prvej kapitole. Okrem iného sa tu venujem aj projektu OpenStreetMap, ktorý slúži pre zber geografických dát a generovanie rozličných druhov máp. Druhá kapitola obsahuje návrh systému, ktorý som na základe získaných znalostí navrhol a podrobnejšiemu popisu použitých metód. Implementácii tohto systému je vyhradená tretia kapitola. Posledná kapitola sa zaoberá jeho testovaním a zhodnotením dosiahnutých výsledkov. Záver obsahuje zhrnutie celej práce a stručne popisuje možné rozšírenia.

# Kapitola 1

## Teoretický rozbor

V tejto kapitole v krátkosti predstavím projekt OpenStreetMap a uvediem objekty, ktoré sú vhodné pre zavedenie do máp generovaných pomocou tohto projektu. Taktiež opíšem základné princípy a metódy zaoberajúce sa nachádzaním objektov v obraze. Táto problematika spadá do oblasti počítačového videnia, ktoré je v neustálom vývoji. Existuje mnoho špecializovaných metód zameraných na rozpoznávanie konkrétnych typov objektov, ako je písmo, ľudská tvár, dopravné značenia či sledovanie pohybujúcich sa objektov. Z týchto metód stručne uvediem len tie, ktoré sú vhodné pre túto prácu.

### 1.1 OpenStreetMap

OpenStreetMap (OSM) je otvorený projekt pre vytváranie voľne editovateľných a voľne šíriteľných geografických dát. Tieto dáta sa používajú pre vytváranie rôznych druhov máp, či už turistických, cyklistických alebo automáp. Dáta nahraté na server OSM spadajú pod licenciu Creative Commons Attribution-ShareAlike 2.0. Na ich tvorbe sa podieľajú užívatelia z celého sveta a ich počet pomaly narastá. V roku 2011 počet registrovaných užívateľov presiahol hodnotu 500 000. Pre vytváranie dát sa používajú záznamy z GPS prijímačov, prípadne satelitné snímky a iné mapy, ktoré spĺňajú licenčné podmienky [1].

#### Formát dát

Dáta sa na server ukladajú vo formáte xml, pričom sú zložené z nasledujúcich prvkov:

- bod (*node*) - je určený gps súradnicami
- cesta (*way*) - tvorí ju usporiadaný zoznam po sebe idúcich bodov (*nodes*)
- relácia (*relation*) - skupina bodov, ciest alebo iných relácií, ktoré môžeme zoskupiť do jedného objektu (napr. budovy, vodné plochy, lesy)
- atribúty (*tag*) - určujú vlastnosti bodu, cesty alebo relácie, pozostávajú z typu (*key*) a hodnoty (*value*)

### 1.2 Vhodné objekty pre mapovanie

Ako už bolo naznačené, cieľom práce je nájsť vhodné objekty vo vstupnom videozázname získaného z idúceho vozidla a nahrať ich na server OSM. Takýmito objektmi môžu byť

napríklad semaforey, dopravné značky, prechody pre chodcov, prípadne železničné priecestia (výstražná svetelná signalizácia). Najvýznamnejšou množinou vhodnou pre zavedenie do máp sú dopravné značky, pretože majú vysoký informatívny charakter a môžu sa využiť pri tvorbe automáp, v dopravných navigáciách alebo pri plánovaní trasy.

**Typy dopravných značiek** Dopravné značky rôznych štátov sa odlišujú v niektorých detailoch. Zoznam všetkých dopravných značiek platných na území Slovenskej Republiky je daný vyhláškou ministerstva vnútra č. 9/2009 Z. z. Na základe tohto dokumentu ich delíme na:

1. Zvislé dopravné značky
  - (a) Výstražné dopravné značky
  - (b) Značky upravujúce prednosť a dodatkové tabuľky s tvarom križovatky
  - (c) Zákazové značky
  - (d) Príkazové značky
  - (e) Informatívne prevádzkove značky
  - (f) Informatívne smerové značky
  - (g) Informatívne iné značky
2. Vodorovné dopravné značky

Každá skupina je charakteristická svojou farbou a tvarom. Napríklad u výstražných značiek je to červený trojuholník, u zákazových červený kruh a podobne.

### 1.3 Základná metodológia

Uvedené vlastnosti, teda farba a tvar, sú základným záchytným bodom metód zameraných na hľadanie objektov v obraze. Ich podstata je preto rovnaká a celý postup môžeme rozdeliť do niekoľko bodov:

1. Predspracovanie obrazu - odstránenie šumu, konverzia do iného farebného modelu.
2. Segmentácia a detekcia objektu - vymedzenie oblastí, kde sa potencionálny objekt nachádza s využitím informácie o farbe a tvare.
3. Rozpoznanie objektu (klasifikácia) - určenie konkrétneho typu detekovaného objektu, jeho zaradenie do triedy.

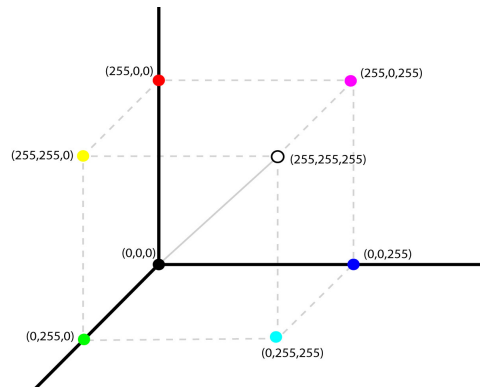
### 1.4 Farebné modely

Farebný model je abstraktný model popisujúci farbu skladajúcu sa z rôznych zložiek, najčastejšie z troch. Výsledná farba pixelu vznikne ich kombináciou. Použitie vhodného farebného modelu môže značne uľahčiť proces segmentácie, a následne aj celkovej detekcie a rozpoznania objektu v obraze. Pretože takmer všetky zariadenia na získavanie obrazu (kamery, fotoaparáty) pracujú implicitne s RGB modelom, použitie iného farebného modelu znamená konverziu z RGB.



## RGB

RGB, základný model používaný v počítačovej grafike si môžeme predstaviť ako kocku v systéme kartézskych súradníc, kde  $x, y, z$  predstavujú farby: červenú, zelenú a modrú (Red, Green, Blue). RGB je aditívny model, kde kombináciou jednotlivých zložiek  $R, G$  a  $B$  dostaneme výslednú farbu. Je však dosť neintuitívny, hlavne pri práci s intenzitou svetla. Preto je obtiažné dosahovať dobré výsledky segmentácie pri rôznych svetelných podmienkach. Výhodou je jeho jednoduchosť a to, že nie je nutná konverzia pôvodného obrazu. Aj keď mnoho autorov doporučuje konverziu RGB do iného farebného modelu, existujú metódy ktoré tento model v upravenej forme používajú [2].



Obr. 1.1: RGB model.

## HSV

HSV model je bližšie k ľudskému vnímaniu farieb. Výslednú farbu získame ako Hue (odtieň) Saturation (sýtosť farby) a Value (úroveň jasu - množstvo bieleho svetla). Konverzia z RGB do HSV, ktorá sa opisuje v [3] prebieha nasledovne:

$$H = \cos^{-1} \left( \frac{0.5(R - G) + (R - B)}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \quad (1.1)$$

$$S = 1 - \left( \frac{3}{R + G + B} \right) \min(R, G, B) \quad (1.2)$$

$$V = \max(R, G, B) \quad (1.3)$$

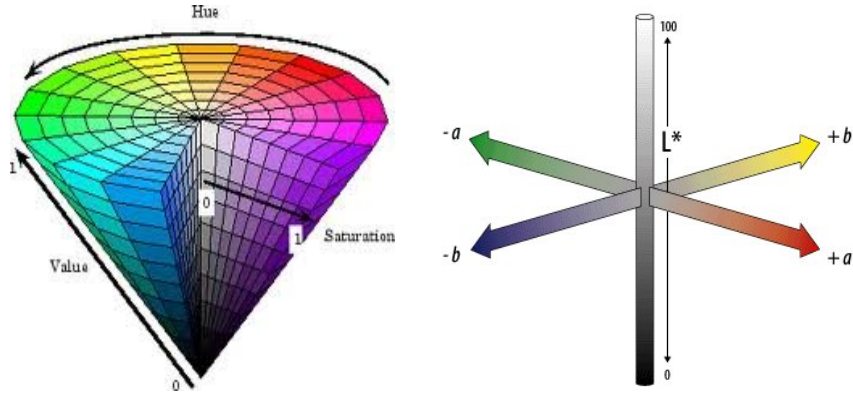
Tento prevod môže byť ešte optimalizovaný pre vylepšenie vlastností modelu. Metóda, ktorá je detailnejšie opísaná v [4], je viac odolná voči zmenám farby a voči tieňom.

## CIElab

Farebný model CIElab, je vylepšenou verziou modelu CIE XYZ [5]. Skladá sa zo zložiek  $L, a, b$ . Zložka  $L$  predstavuje jas(svetlosť), čiže zastúpenie čiernej a bielej farby v rozmedzí  $[0,100]$ . Zložka  $a$  predstavuje zelenú ( $a-$ ) a červenú farbu ( $a+$ ) v rozmedzí  $[-500, 500]$ . Zložka  $b$  predstavuje modrú ( $b-$ ) a žltú farbu ( $b+$ ) z intervalu  $[-200, 200]$ . Odlíšné rozsahy

súvisia s tým, že model je prispôsobený ľudskému vnímaniu farby, ktoré je na červenú a zelenú zložku citlivejšie ako na modrú a žltú.

Konverzia z RGB do CIElab je zložitejšia. Najprv je treba previesť RGB obrázok do CIE XYZ modelu a následne z CIE XYZ do CIE lab. Tento postup je opísaný v [6].



Obr. 1.2: Vľavo HSbV model, vpravo CIElab.

## 1.5 Segmentácia

Prvým krokom k rozpoznaniu objektu v obraze je segmentácia. Jej cieľom je rozčleniť vstupný digitálny obraz do častí, ktoré reprezentujú nejaký predmet reálneho sveta. Výstupom zvyčajne býva binárna mapa, v ktorej rozlišujeme popredie a pozadie. Popredie predstavuje potencionálne hľadaný objekt a v binárnej mape je vyplnené bielou farbou. Pozadie, teda okolie objektu, je vyplnené čiernou farbou. Príklad takejto binárnej mapy je na obrázku 1.3. Vstupný obraz môže byť a vo väčšine prípadov je zaťažený šumom, ktorý zťažuje celý proces segmentácie. Preto sa v praxi používajú kombinácie rôznych metód. V tejto časti uvediem tri základné prístupy k segmentácii. Väčšina informácií pochádza z [7].

### Prahovanie

Segmentácia prahovaním je najjednoduchšia a najrýchlejšia metóda, ktorá berie v úvahu farbu alebo jas pixelov v obraze. Základný princíp transformácie vstupného obrazu na obraz binárny je nasledovný :

$$g(i, j) = \begin{cases} 1 & \text{pre } f(i, j) \geq T \\ 0 & \text{pre } f(i, j) < T \end{cases} \quad (1.4)$$

kde  $f(i, j)$  je vstupný obraz,  $g(i, j)$  je výstupný obraz a  $T$  je vhodne zvolený prah (úroveň jas, farby). Keďže prah  $T$  nás obmedzuje len na jednu hodnotu, čo je v praxi nepoužiteľné, používa sa namiesto prahu  $T$  množina  $D$ . Táto množina predstavuje interval prípustných hodnôt prahov. Prah môžeme určiť ručne (musíme preto poznať množinu prípustných hodnôt), alebo môže byť určený (prípadne len upravený) automaticky pomocou analýzy histogramu.

Pre segmentáciu prahovaním je vhodné prekonvertovať vstupný obraz z RGB formátu do farebného modelu, ktorý umožňuje lepšiu manipuláciu s jasom a farbou zvlášť. Nie je

to však nutné. Metóda prahovania bola použitá v dokumente [8], kde ju autor používa na detekciu dopravných značiek.



Obr. 1.3: Pôvodny RGB obraz a Binárna mapa po RGB prahovaní.

## Detekcia hrán

Metódy detekcie hrán predpokladajú, že hranice oblastí pozostávajú z hrán, teda z miesta kde sa výrazne mení jas, farba prípadne textúra. Pre detekciu hrán sa používajú hranové operátory, najvýznamnejšie z nich sú Sobelov, Laplaceov a Cannyho hranový operátor. Po aplikácii hranového operátora vznikne obraz s množstvom nespojitých hrán. Spojením týchto hrán do reťazcov získame ohraničené oblasti reprezentujúce objekty v obraze. Na spájanie hrán existuje niekoľko metód, ktorých princíp a použitie sú podrobnejšie opísané v [7].

Oproti prahovaniu si tieto metódy poradia lepšie s obrazom, v ktorom chýba informácia o farbe, pretože používajú obraz prevedený do odtieňov šedej. Na druhej strane sú veľmi náchylné na vplyv šumu. Jeho následkom sa v obraze hrán môže vyskytovať množstvo drobných, nežiaducich hrán.

V praxi sa často stretávame s kombináciou viacerých metód. Použitie farebnej segmentácie prahovaním a detekciou hrán opisujú autori v článku [4].

## Narastanie oblastí

Základom segmentácie pomocou narastania oblastí je rozdeliť obraz do homogénnych, maximálne súvislých oblastí. Princíp je jednoduchý: susedné pixely s rovnakými vlastnosťami (napr. úroveň jasů, textúra) sú združované k sebe a tým vytvárajú oblasti. Vzniknuté oblasti sú následne spájané do väčších celkov tak, aby platili vzťahy [7]:

$$\begin{aligned} H(R_i) &= TRUE; i = 1, 2, \dots, S, \\ H(R_i \cup R_j) &= FALSE; i \neq j \wedge R_i \text{ susedis } R_j \end{aligned} \quad (1.5)$$

kde  $S$  je celkový počet oblastí a  $H(R_i)$  je ohodnotenie homogenity oblasti  $R_i$ .

## 1.6 Detekcia tvarov

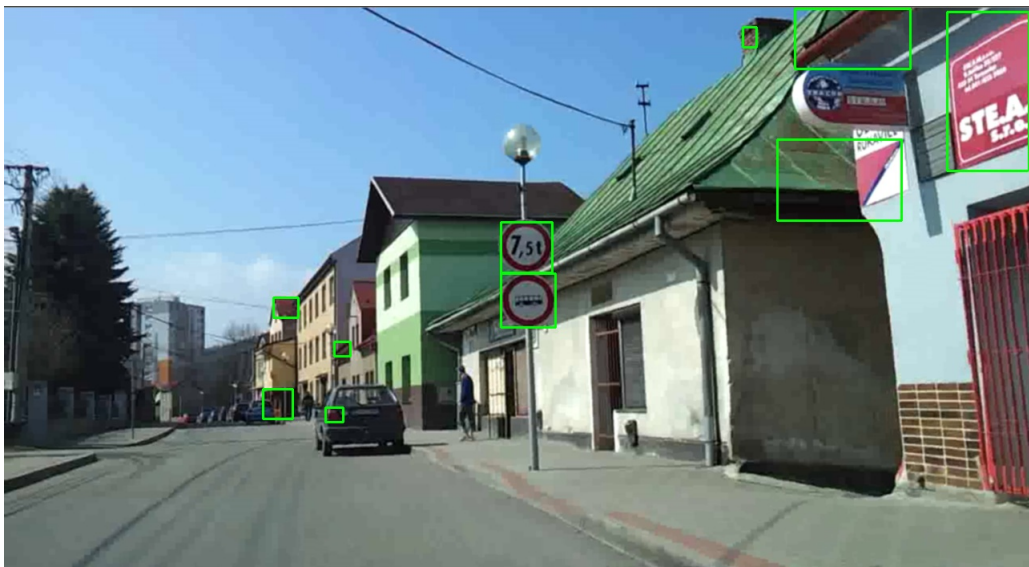
Cieľom detekcie je redukovať prehľadávaný priestor vo vstupnom obraze pre následné rozpoznávanie, čím sa celý proces urýchli a zároveň sa znižuje šanca na chybné rozpoznanie objektu. Vstupom pre detekciu je binárna mapa, v ktorej rozdeľujeme popredie a pozadie. Ďalším krokom je teda v týchto bodoch nájsť súvislosti - či dokopy tvoria nejaký geometrický útvar ako napr. trojuholník, kruh, priamka, atď. Na túto úlohu je vhodná Houghova Transformácia [9].

### Houghova Transformácia

Pôvodne bola navrhnutá na detekciu priamok. Jej zobecnená forma umožňuje vyhľadávať aj iné objekty [7]. Pre hľadanie kruhových oblastí, (čo je jeden z najčastejších tvarov dopravných značiek) je analytický popis, teda rovnica kružnice nasledovná:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (1.6)$$

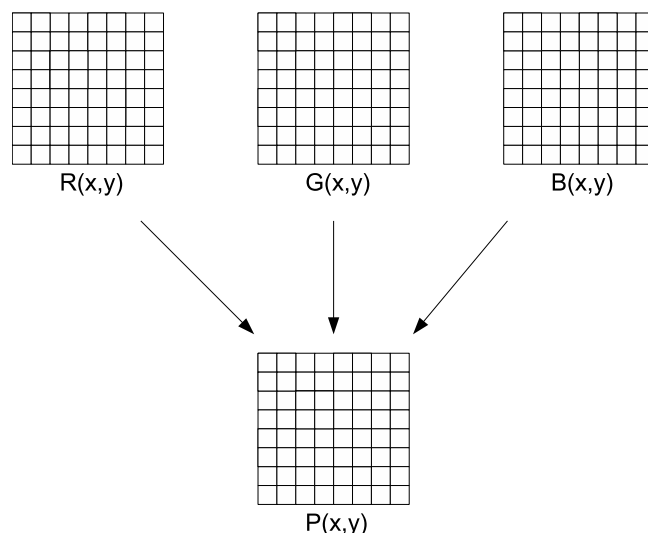
Pomocou takto popísanej kružnice je algoritmus schopný nájsť kružnice bez závislosti na ich veľkosti či polohe. Podobne môže byť použitá aj pre iné objekty ku ktorým existuje analytický popis, no s rastúcim počtom popisujúcich rovníc rastie aj výpočtová náročnosť.



Obr. 1.4: Príklad výstupu detekcie. Nad vyznačenými objektmi prebieha následná klasifikácia.

## 1.7 Extrakcia príznakov

Ak sme predchádzajúcimi postupmi vymedzili plochu s potencionálnym hľadaným objektom, získame obraz tvorený (v závislosti od použitého farebného modelu) niekoľkými maticami bodov. Tieto matice obsahujú hodnoty pixelov v každom farebnom kanáli (R,G,B), ako je znázornené na obrázku 1.5. Pri použití niektorej z metód strojového učenia (NN, SVM) je potrebné z týchto matíc nejakým spôsobom vyextrahovať dôležité informácie, ktoré vhodne popisujú daný obraz, a vytvoriť vektor príznakov.



Obr. 1.5: Transformovanie troch matíc (kanálov) R,G,B do jednej matice P.

### Prevod do odtieňov šedej

Najjednoduchším riešením je z jednotlivých kanálov vytvoriť jednokanálový obraz, čo sa dá doceliť napríklad prevodom do odtieňov šedej. V takto vzniknutej matici tvoria príznaky jednotlivé úrovne jasů pixelov. Nevýhodou je, že takto vytvorený vektor príznakov nepopisuje obraz všeobecne, teda nie je invariantný voči natočeniu, rotácii, výskytu šumu a voči zmenám svetelných podmienok.

### HOG

Histogram orientovaných gradientov (HOG) je metóda extrakcie príznakov, ktorá sa vo veľkom používa pre detekovanie chodcov v obraze. HOG opisuje obraz pomocou niekoľkých lokálnych histogramov. Tieto histogramy uchovávajú výskyty orientácií gradientov [10]. Postup výpočtu lokálnych histogramov je nasledovný:

1. Výpočet gradientov z obrazu.
2. Vytvorenie histogramu orientácii gradientov pre každú bunku.
3. Normalizácia histogramu rozdelením do blokov.

**Výpočet gradientu** prebieha aplikovaním konvolučnej masky v horizontálnom a vertikálnom smere. Príkladom takejto masky je 1.7.

$$\begin{aligned} \textit{horizontal} &: (-1 \ 0 \ 1) \\ \textit{vertical} &: (-1 \ 0 \ 1)^T \end{aligned} \tag{1.7}$$

**Vytvorenie histogramu** spočíva v rozdelení obrazu na malé bunky, ktoré majú preddefinovanú veľkosť. V rámci každej bunky sa spočíta histogram gradientov sčítaním veľkostí gradientov pre každú orientáciu. Týmto z jednej bunky získame jeden príznak, zlúčením všetkých príznakov dostávame vektor príznakov ktorý popisuje obraz.

**Normalizácia histogramu** sa vykonáva za účelom dosiahnutia vyššej presnosti a pomáha odbúrať malé výskyty zmien v jase. Podstatou normalizácie je zohľadniť hodnoty okolitých buniek tým, že výpočet príznaku prebieha v rámci blokov, ktoré sa skladajú z niekoľkých buniek.

## 1.8 Rozpoznávanie

Rozpoznávanie, alebo aj klasifikácia, je posledným krokom, v ktorom sa detekovaný objekt porovnáva s databázou vzorových obrazov. V tejto fáze sa rozhodne, či sa jedná o objekt ktorý hľadáme, v prípade že áno, rozhodne sa o jeho príslušnosti do triedy.

### Template matching

Táto metóda porovnáva šablónu (template) s časťou vstupného obrazu, ktorá bola vymedzená detekciou. Porovnanie prebieha pixel za pixelom a za triedu, do ktorej objekt patrí sa vyberie šablóna s najväčšou zhodou. Pred porovnávaním je potreba oba obrazy normalizovať na rovnakú veľkosť. Autor článku [2] použil túto metódu s celkovou úspešnosťou 78% rozpoznaných dopravných značiek.

### Neurónové siete

Neurónové siete (Neural networks) [11] sú výpočtové modely inšpirované biologickými neurónovými systémami ako je ľudský mozog. Základnou jednotkou sú neuróny, ktoré vzájomnou spoluprácou riešia problémy. Neurón má niekoľko vstupov, buď od iných neurónov alebo z okolia a spravidla jeden výstup.

Činnosť neurónových sietí môžeme v stručnosti rozdeliť do dvoch hlavných fáz. Prvou je fáza učenia, kedy sa na vstup siete privádzajú tréningové dáta. Učenie môže prebiehať dvoma spôsobmi:

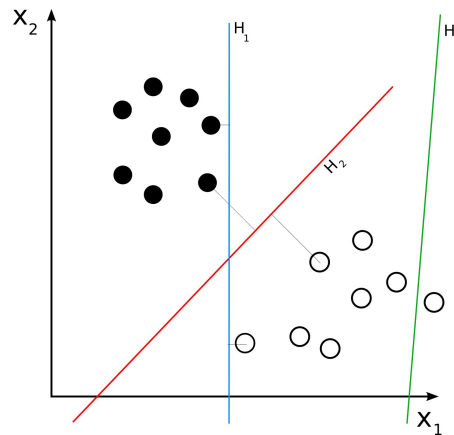
- Kontrolované učenie s učiteľom (supervised learning) spočíva v tom, že sieť má k dispozícii vstupy, ktoré sa snaží čo najlepšie namapovať na vopred známe výstupy upravením váh spojení medzi neurónmi.
- Učenie bez učiteľa (unsupervised learning), kedy sú dané len vstupy, ktoré sa sieť snaží utriediť, no nie je tu žiadna spätná väzba.

Priradzovaním váh jednotlivým neurónom si teda sieť vytvorí model, ktorý následne v druhej fáze využíva pre riešenie problémov. Jedným z najčastejších oblastí pre ktoré sa neurónové siete využívajú je aj klasifikácia objektov. Existuje mnoho typov neurónových sietí, medzi najrozšírenejšie architektúry patrí MLP (Multilayer perceptron). Túto architektúru vo svojej práci použili autori článku [12] na rozpoznávanie dopravných značiek v reálnom čase. V tejto štúdií sa zamerali na minimalizovanie chýb pri klasifikácii.

### SVM

SVM je skratka pre podporné vektory (Support Vector Machine) a označuje metódy strojového učenia s učiteľom. Základnou myšlienkou týchto metód je oddeliť dve triedy vstupných dát (tzv. pozitívnu a negatívnu sadu) pomocou lineárnej hranice. Cieľom je nájsť najlepšiu deliacu hranicu, teda maximalizovať hranice medzi týmito dvoma triedami

(viď obrázok 1.6). Najväčšou výhodou je, že ak nie je možné tieto dve triedy oddeliť, SVM sú schopné previesť vstupný priestor do viacrozmerného priestoru, kde sú triedy lineárne oddeliteľné [13].



Obr. 1.6: SVM - ukážka deliacích rovín, červená je optimálna. Zdroj: [14].

Keďže SVM je navrhnutý ako binárny klasifikátor, je v pôvodnej verzii na klasifikovanie rôznych značiek takmer nepoužiteľný. Existuje však niekoľko jednoduchých metód využívajúcich hierarchickú štruktúru niekoľkých SVM klasifikátorov pre riešenie viactriednych problémov pomocou binárnych klasifikátorov.

Tieto metódy boli predstavené v dokumente [15]:

**One-against-all** Pre problémy skladajúce sa z  $N$  tried, kde  $N > 2$ , je zostrojených  $N$  binárnych klasifikátorov. Trieda  $i$ , kde  $i = 0..N$ , je označená ako pozitívna sada a všetky ostatné triedy tvoria sadu negatívnu. V rozpoznávacej fáze je potom testovacia vzorka predložená každému SVM klasifikátoru a vybraná je tá, pre ktorú sa nájde najlepšie ohodnotenie.

**One-against-one** Táto metóda je podobná predchádzajúcej, no tu proti sebe stoja vždy pozitívna a negatívna tréningová sada zložená len z jednej triedy. Vznikne tak  $N(N - 1)/2$  binárnych klasifikátorov. Výhodou oproti predchádzajúcej metóde je, že obe sady vyžadujú menší počet dát. Na druhej strane, nevýhodou je veľký počet vzniknutých klasifikátorov a s ním súvisiaca náročnosť výpočtu (hlavne pre vysoké  $N$ ).

Support Vector Machine nachádza čoraz častejšie použitie pri rozpoznávaní dopravných značiek. Niekoľko rôznych variácií SVM opisujú autori v [16]. V tejto práci dosiahli na vybraných druhoch značiek úspešnosť klasifikácie až 99%-100% .

# Kapitola 2

## Návrh riešenia

Po zhrnutí základných princípov je treba navrhnuť systém, ktorý bude zvolené objekty detekovať, klasifikovať a nahrávať na server OSM. V tejto kapitole popíšem navrhovaný systém, taktiež uvediem podrobnejší popis použitých metód a niektoré problémy, na ktoré som počas návrhu a implementácie narazil.

### 2.1 Popis prvkov navrhnutého systému

Navrhnutý systém (obrázok 2.1) sa skladá z dvoch samostatných celkov pre rozpoznávanie značiek a pre ich zavedenie do mapy. Komunikáciu medzi nimi zabezpečuje súbor, ktorý je výstupom rozpoznávania.

**Vstup:** Vstupom celého systému bude videozáznam a GPS záznam získaný videokamerou z idúceho vozidla, prípadne z iného dopravného prostriedka. Je dôležité aby sa tieto dva záznamy dali zosynchronizovať, resp. aby bolo možné priradiť súradnice z GPS prístroja k nájdenej dopravnej značke.

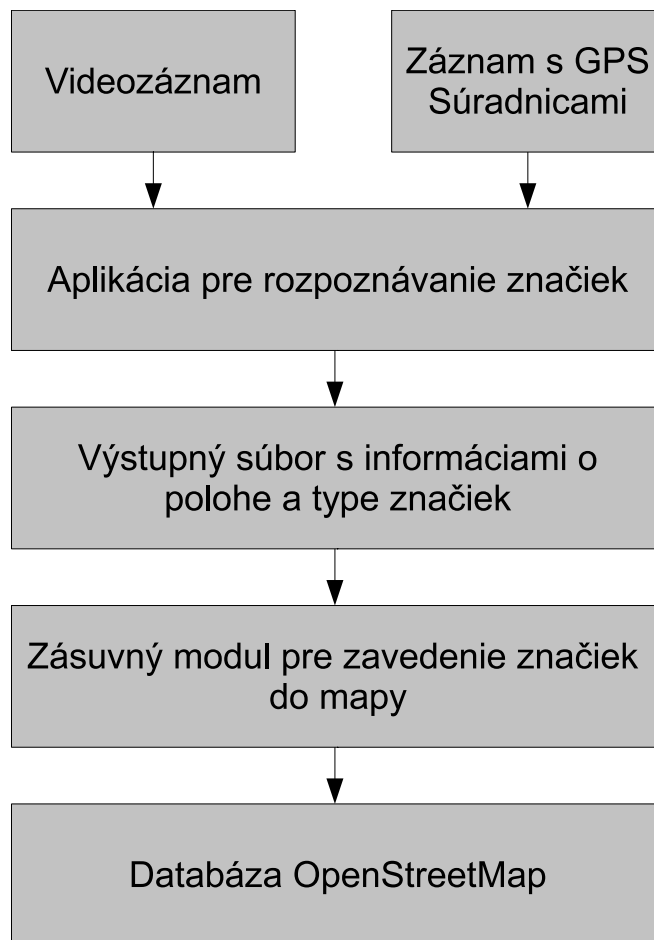
**Aplikácia:** Za predpokladu, že máme požadované vstupy, je na rade aplikácia, ktorá ich spracuje. Z videozáznamu budú istou frekvenciou získavané snímky a na nich bude prevádzaná detekcia a rozpoznávanie.

**Výstupný súbor:** Výstup aplikácie na rozpoznávanie značiek bude súbor vo formáte xml. Tento súbor bude obsahovať dôležité informácie, potrebné pre správne zavedenie do mapy. Slúži ako premostenie medzi aplikáciou na rozpoznávanie značiek a zásuvným modulom. Taktiež každá značka, ktorá sa zapíše do xml súboru bude uložená aj vo formáte jpg.

**Zásuvný modul:** Tento zásuvný modul (plugin) rozširuje vybraný, už existujúci editor pre prácu s geografickými dátami. S databázou OSM bude plugin komunikovať prostredníctvom API editora.

**Databáza OSM:** Po pridaní dát prostredníctvom pluginu sa za pomoci editora dáta nahrajú na server OSM. Tu budú prístupné ostatným užívateľom a bude ich možno využiť ku generovaniu máp.





Obr. 2.1: Prvky navrhnutého systému. Komunikácia je len jednosmerná, každá časť využíva výstupy predošlého prvku.

## 2.2 Získanie vstupných záznamov

Na získanie vstupov je možné použiť takmer akékoľvek zariadenie schopné zaznamenať video vo formáte avi alebo mp4. Čo sa týka GPS záznamu, je nutné aby zariadenie podporovalo formát gpx. Taktiež je vhodné aby toto zariadenie umožňovalo nastaviť frekvenciu zaznamenávania polohy (napr. záznam každých 5 metrov).

Pre účely tejto práce bude na zber dát použitý mobilný telefón Samsung Galaxy S II (obrázok 2.2). Výhodou tohto zariadenia je jeho kompaktnosť, zabudovaný 8Mpx fotoaparát a GPS prijímač.

## 2.3 Množina detekovaných objektov

Spomedzi všetkých objektov som sa zameril výhradne na dopravné značky, ktorých spoločným znakom je červená farba. To zužuje výber na výstražné, zákazové značky a značky upravujúce prednosť v jazde. Zoznam všetkých podporovaných značiek je uvedený v dodatku A.



Obr. 2.2: Použité zariadenie na získanie záznamov. Zdroj:[17].

## 2.4 Návrh Aplikácie

V rámci návrhu aplikácie opíšem detailnejšie niektoré použité metódy, ktoré boli spomenuté v prvej kapitole.

### HSV a farebná segmentácia

V prvom kroku sa obraz prevedie z RGB do HSV priestoru pomocou algoritmu uvedenom v prvej kapitole (1.4). V takto prevedenom obraze sa sekvenčne prejde každý pixel. Ak jeho jednotlivé zložky  $H, S, V$  spadajú do prípustného intervalu prahov, označí sa za objektový pixel a vyplní sa bielou farbou. V opačnom prípade bude predstavovať pozadie a bude vyplnený čiernou farbou.

Tento proces ilustruje nasledovný algoritmus:

```
for all pixels {  
    if (H>H_min && H<H_max && S>S_min && S<S_max && V>V_min && V<V_max) then  
        object pixel;  
    else  
        background pixel;  
}
```

$H, S, V$  sú hodnoty aktuálne spracovávaného pixelu,  $H_{min}, S_{min}, V_{min}$  sú dolné hranice prahu a  $H_{max}, S_{max}, V_{max}$  horné hranice predstavujúce červenú farbu.

Štúdia ohľadom optimálnych prahov vybraných farebných modelov [18], poslúži pre východzie hodnoty prahov. Tieto hodnoty sú kľúčové a preto budú predmetom testovania.

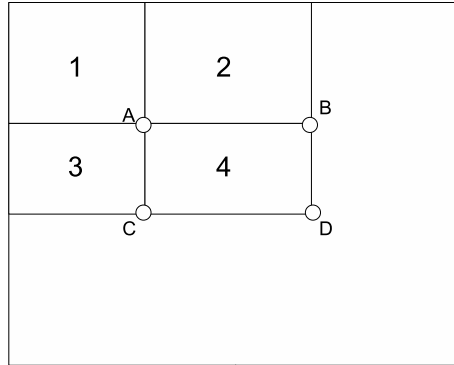
### Extrakcia príznakov pomocou HOG

Pred samotnou extrakciou príznakov pomocou HOG sa obraz upraví na normovanú veľkosť 32x32 pixelov a vypočíta sa z neho integrálny histogram.

Zostavenie integrálneho histogramu predstavil autor článku v [19] a vychádza z princípu integrálneho obrazu [20]. Integrálny obraz možno popísať rovnicou 2.1, kde  $ii$  je integrálny obraz a  $x, y$  sú súradnice pixelu.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.1)$$

Na pozícii  $x, y$  obsahuje súčet hodnôt (intenzít) všetkých predchádzajúcich pixelov vľavo a nahor od pozície  $x, y$  po počiatok súradnicového systému. Hlavnou výhodou integrálneho obrazu je, že súčet intenzít v rámci ľubovoľnej oblasti trvá konštantný čas. Ak chceme sčítať intenzity v oblasti určenej bodmi  $A, B, C, D$ , ako je naznačené na obrázku 2.4, použijeme vzťah  $D + A - B - C$ . To znamená 4 prístupy do poľa a 3 operácie pre akúkoľvek obdĺžnikovú oblasť.



Obr. 2.3: Hodnota v bode A predstavuje súčet všetkých hodnôt v oblasti 1. Súčet hodnôt obdĺžnikov 1 + 2 predstavuje bod B, obdĺžnikov 1 + 3 bod C. Analogicky bod D obsahuje súčet obdĺžnikov 1, 2, 3 a 4. Ak chceme vypočítať obsah len obdĺžnika 4, odčítame od nej zvyšné oblasti 1, 2, 3. Vo výsledku dostávame vzťah  $D + A - B - C$ .

Integrálny histogram namiesto intenzít obsahuje veľkosti gradientov. Jeho vytvorenie opíšeme podrobnejšie a vychádza z [21].

Najprv sa na obraz aplikuje Sobelov operátor pre detekciu vodorovných a zvislých hrán. Z týchto dvoch obrazov sa vypočíta orientácia gradientu (2.2) a jeho veľkosť (2.3).

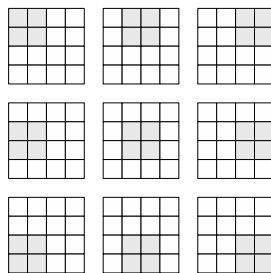
$$\theta = \operatorname{arctg} \left( \frac{G_y}{G_x} \right) \quad (2.2)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.3)$$

Obor hodnôt funkcie  $\operatorname{arctg}$  je  $\langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$ , takže získame orientácie gradientu v tomto intervale<sup>1</sup>. Na základe tohto intervalu ( $180^\circ$ ) sa vytvorí 9 obrazov, rozmerovo rovnakých ako pôvodný obraz, ktoré budú slúžiť pre uchovanie veľkosti gradientov. Každý obraz teda zodpovedá intervalu o veľkosti  $20^\circ$  ( $180^\circ/9$ ). V poslednom kroku sa sčítajú hodnoty v každom z deviatich obrazov, tak ako v integrálnom obraze (2.1). Takto sme vytvorili integrálny histogram, ktorý sa skladá z deviatich obrazov, každý pre inú orientáciu gradientu. Pre každý z nich platí vlastnosť, ktorú má integrálny obraz, čiže výpočet ľubovoľného obsahu štvorca v obraze trvá konštantný čas.

V obraze, ktorý sme na začiatku upravili na rozmery  $32 \times 32$  a pre ktorý sme vytvorili integrálny histogram, vypočítame príznaky. Tento postup sa opisuje v [21]. Rozdelíme si obraz na bloky, každý o rozmeroch  $16 \times 16$ , pričom sa navzájom prekrývajú tak ako naznačuje obrázok 2.4.

<sup>1</sup>Niektoré implementácie funkcie  $\operatorname{arctg}$  vracajú hodnoty v intervale  $\langle -\pi, \pi \rangle$ . V jazyku C je to funkcia  $\operatorname{atan2}$ . Použitím tejto funkcie sa interval dvojnásobne zväčší, čím možno dosiahnuť lepšiu presnosť.



Obr. 2.4: Prekrývajúce sa bloky pre výpočet príznakov.

V každom bloku sú 4 bunky o veľkosti 8x8 pixelov, pre každú z nich sa vypočíta súčet veľkosti gradientov. Využije sa pri tom už spomínaný vzťah  $D + A - B - C$ . Keďže sa suma počíta z deviatich integrálnych histogramov, pre jednu bunku získame 9 príznakov. Pre blok je to teda  $4 * 9 = 36$  príznakov. A pretože máme 9 prekrývajúcich sa blokov, získame celkovo  $9 * 36 = 324$  príznakov pre jeden obraz o veľkosti 32x32 pixelov.

Tento postup sa dá aplikovať na akýkoľvek veľký obraz. Čím väčšie rozmery má, tým viac príznakov z neho získame.

### Trénovacia sada

Trénovaciu sadu môžeme rozdeliť na negatívne dáta a pozitívne dáta. V našom prípade pozitívne dáta obsahujú všetky dopravné značky, ktoré chceme vo videu detekovať a klasifikovať. Na druhej strane negatívne dáta obsahujú všetky obrázky, v ktorých sa značky nenachádzajú. Je dôležité, aby aj negatívne obrázky zodpovedali objektom, ktoré sa vo videu môžu vyskytovať, to znamená, že do tejto sady zahrnieme najmä obrázky áut, semaforov, ľudí, budov a podobne.

Pozitívne dáta obsahujú fotografie dopravných značiek, nafotených ručne za rôznych svetelných podmienok, z rôznych videí a z internetových zdrojov. Všetky dopravné značky sú orezané a upravené na veľkosť 32x32 pixelov. Rovnakým spôsobom je zostavená aj negatívna sada, doplnená o databázu INRIA person dataset [22]. Tieto obrázky sú v rozličných rozlíšeních.

### Klasifikácia pomocou SVM

SVM predpovedá s akou pravdepodobnosťou patrí objekt do jednej z dvoch skupín. Preto, aby sme mohli správne rozlíšiť dopravnú značku, je potrebné vytvoriť niekoľko spolupracujúcich klasifikátorov, ktoré budú radené do kaskády. Na túto úlohu sa využije metóda One-against-all, načrtnutá v predchádzajúcej kapitole (1.8).

Trénovanie SVM prebieha za pomoci pozitívnych a negatívnych dát, z ktorých boli získané vektory príznakov pomocou HOG. Vykonáva sa na niekoľkých úrovniach:

**Filter:** Na prvej úrovni sa klasifikátor učí rozlišovať, či ide o značku alebo nie, čiže trénovacími dátami sú pozitívna sada obsahujúca obrázky všetkých značiek a negatívna sada, v ktorej sú všetky obrázky, ktoré značku neobsahujú.

**Tvar:** Na druhej úrovni sa rozhoduje o tvare značky. Keďže možné tvary sú 4 (trojuholník, obrátený trojuholník, osemuholník a kruh), vytvoria sa 4 rôzne klasifikátory, pričom každý

rozhoduje o jednom tvare. Pre účely tréovania je preto potrebné rozdeliť pozitívne dáta na dve sady. Úlohu pozitívnej sady preberajú značky jedného tvaru, všetky ostatné značky tvoria negatívnu sadu.

**Typ:** Na poslednej úrovni sa rozhoduje o konkrétnom type značky, preto je v rámci každého typu vytvorených toľko binárnych klasifikátorov, koľko druhov značiek obsahuje. Princíp rozdelenia sady na pozitívnu a negatívnu je podobný ako v predchádzajúcom prípade. V prípade, že v rámci jedného tvaru je len jeden druh značky, nie je pre tento tvar potrebné vytvárať klasifikátory tretej úrovne.

Pri predikovaní SVM klasifikátor prvej úrovne vracia hodnotu buď zápornú alebo kladnú. V prípade, že je hodnota kladná, o značku sa nejedná a v rozhodovaní sa nepokračuje. V opačnom prípade sa pokračuje v klasifikovaní tvaru. Pri klasifikovaní tvaru a typu opäť vracia buď zápornú alebo kladnú hodnotu, hľadá sa teda minimum z predikovaných hodnôt.

Postup ako prebieha klasifikovanie je znázornený na obrázku 2.5.

## Sledovanie značky

Predchádzajúce postupy slúžia pre detekovanie a klasifikovanie dopravných značiek. Pretože ich chceme zaviesť do mapy a vstupný obraz je získaný z idúceho vozidla, nastáva niekoľko problémov. Tým najhlavnejším je ako určiť, že značka na snímku je tou istou značkou, ktorá bola nájdená na predchádzajúcom snímku. Pretože automobil sa nepohybuje konštantnou rýchlosťou a môže ľubovoľne meniť smer, či dokonca cúvať, nie je to jednoduché určiť. Na túto úlohu je možné použiť niektorý z algoritmov na sledovanie pohybu. Rovnako ako autor článku [23] som sa rozhodol pre Lucas-Kanadeho algoritmus.

Akonáhle sa v obraze rozpozna dopravná značka, zaregistruje sa nový bod, ktorý sa umiestni do jej stredu a začne sa sledovať pomocou Lucas-Kanadeho algoritmu. Spolu s bodom sa uložia rozmery rozpoznanej značky, jej typ a pozícia. Po načítaní nového snímku sa najprv vyrátajú nové pozície sledovaných bodov. Na základe uložených rozmerov značky, sa vypočítajú súradnice oblasti, v ktorej by sa mala značka v tomto snímku nachádzať.

Ak sa tieto nové súradnice oblasti určia mimo obraz alebo sa k oblasti žiadna značka nepriradí, sledovanie bodu sa preruší. V tej chvíli sa značka pripraví na zápis do súboru. V prípade, že sa v snímku značka nájde a vmestí sa do oblasti, aktualizujú sa hodnoty ako rozmery a typ značky a pokračuje sa v sledovaní bodu.

Sledovanie optického toku okrem toho, že predchádza duplikovaniu značiek na výstupe, pomáha zvýšiť presnosť klasifikácie. Počas sledovania sa uchovávaajú všetky detekované typy a pri zápise sa vyberie ten, ktorý sa vyskytuje najčastejšie.

## Výpočet súradníc a výstupný súbor

Posledným krokom je zapísať do výstupného xml súboru informácie o klasifikovanej značke. Zapísané budú údaje o type, geografických súradniciach (*latitude*, *longitude*) a cesta k obrázku rozpoznanej značky z videa. Z celkového počtu snímok videa a počtu gpx záznamov sa určí koľko snímok prislúcha jednému záznamu. Prepočet prebehne podľa nasledujúceho algoritmu:

```

N=total_frames/total_gpx_entries;

if (current_frame > ((N*active_coord_index)-(N/2))) then
    set activity to next coord;

```

kde `total_frames` je počet snímkov videa, `total_gpx_entries` počet záznamov GPS súradníc, `current_frame` číslo aktuálne spracovávaného snímku a `active_coord_index` je index v rámci poľa, v ktorom sú uložené súradnice. Nastavenie aktivity na ďalší záznam znamená inkrementovanie tohto indexu.

## 2.5 Návrh Zásuvného modulu

Existuje niekoľko editorov pre prácu s geografickými dátami OpenStreetMap. Najvhodnejším z nich je editor JOSM, ktorý má mnohé výhody:

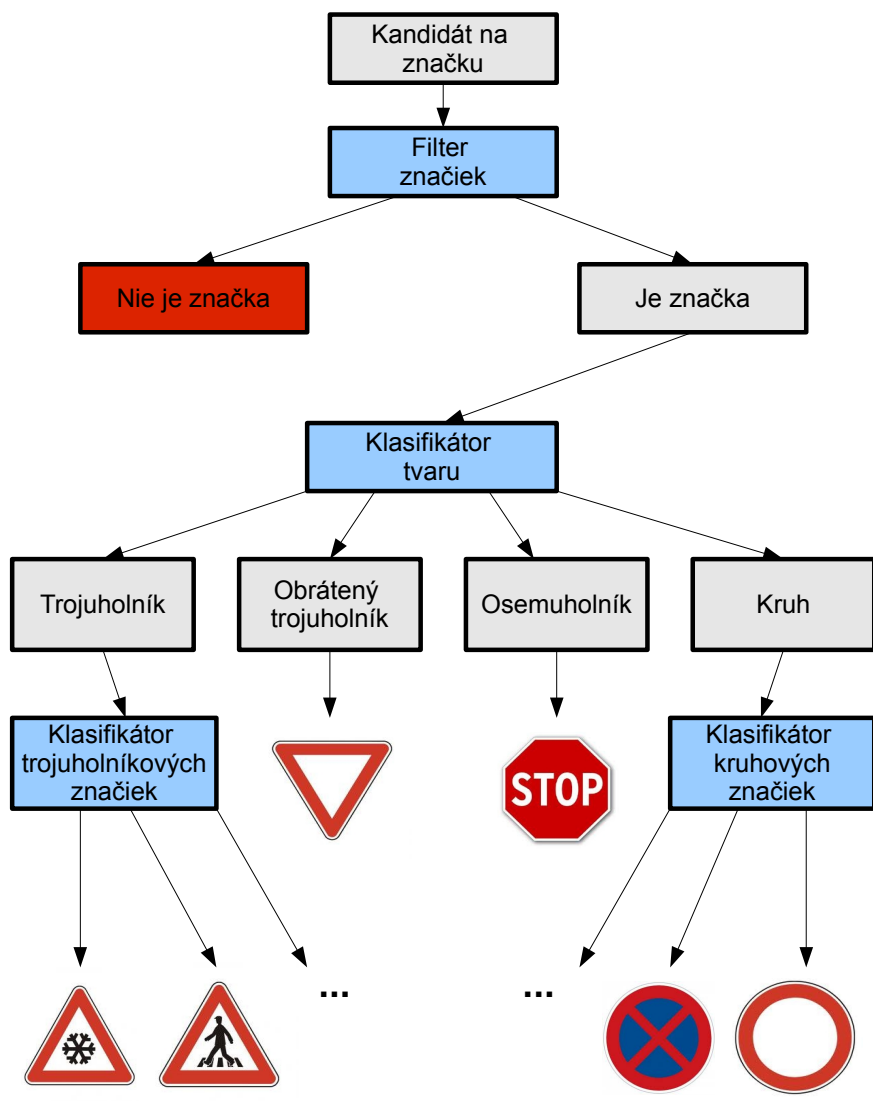
- je v desktopovej verzii, umožňuje prácu offline
- je napísaný v Jave, má voľne prístupné zdrojové kódy
- je ľahko rozšíriteľný pomocou pluginov

Zásuvný modul (plugin) bude pracovať s xml súborom a s obrázkami značiek, rozpoznanými aplikáciou. Plugin bude sekvenčne prechádzať všetkými záznamami a zobrazovať v okne danú značku. Na základe tohto obrázku bude možné určiť, či bola značka rozpoznaná správne.

### Tagy

Pri zavedení značky do mapy sa vytvorí nový bod, ktorému sa pridelia súradnice a typ značky. Typ značky sa uloží ako tag vytvoreného bodu s hodnotami `<key, value>`. Najväčším problémom zavádzania značiek do OSM je pomerne veľká nejednoznačnosť pri určovaní práve `key` a `value`. Existuje niekoľko návrhov na to, ako dopravné značky označovať, no v čase písania tejto práce stále neboli schválené komunitou a v konečnom dôsledku samotné návrhy nepokrývajú všetky značky. Oficiálne sú teda podporované len niektoré značky ako sú Stop, Daj prednosť v jazde, Obmedzenia hmotnosti, výšky, šírky a maximálnej rýchlosti [24].

Tie dopravné značky, ktoré sú oficiálne podporované, budú tagované tak, ako je uvedené na oficiálnych stránkach, pre všetky ostatné bude `key=traffic_sign` a `value` bude rovné názvu značky v angličtine.



Obr. 2.5: Kaskáda binárnych SVM klasifikátorov. Klasifikátor tvaru, trojuholníkových značiek a kruhových značiek pomocou metódy one-against-all vyberá najvhodnejšieho kandidáta. Okrem značky Stop žiadna iná značka nemá tvar osemuholníka, preto nie je potrebné ďalšie delenie. To isté platí u značky Daj prednosť v jazde.

## Kapitola 3

# Implementácia

Aplikácia pre rozpoznávanie dopravných značiek a zásuvný modul boli vyvíjané v odlišnom programovacom jazyku a tvoria samostatné celky. Implementácia a testovanie prebehlo pod operačným systémom Windows 7 Professional SP1 64-bit.

### 3.1 Aplikácia pre rozpoznávanie dopravných značiek

Program pracuje ako konzolová aplikácia, má grafický výstup, no do priebehu výpočtu užívateľ nemôže zasahovať. Pre implementáciu som si zvolil programovací jazyk C/C++. Aplikácia bola vyvíjaná vo vývojovom prostredí Visual Studio 2010.

#### Použité knižnice

Hlavným dôvodom pre zvolenie jazyka C/C++ je, že v tomto jazyku je napísaná multiplatformová knižnica OpenCV, ktorá obsahuje vyše 2500 optimalizovaných algoritmov pre prácu s obrazom a počítačové videnie [25]. Knižnica je šírená pod BSD licenciou a je v neustálom vývoji. Pri implementácii bola využitá verzia 2.2, aj keď v čase písania práce už bola dostupná verzia 2.4 beta.

Pre prácu s xml súbormi som použil C++ knižnicu tinyxml vo verzii 2.6.2. Poskytuje základné operácie ako je čítanie, parsovanie a zápis xml súborov, ktoré pre účely práce postačujú.

#### Popis hlavných tried

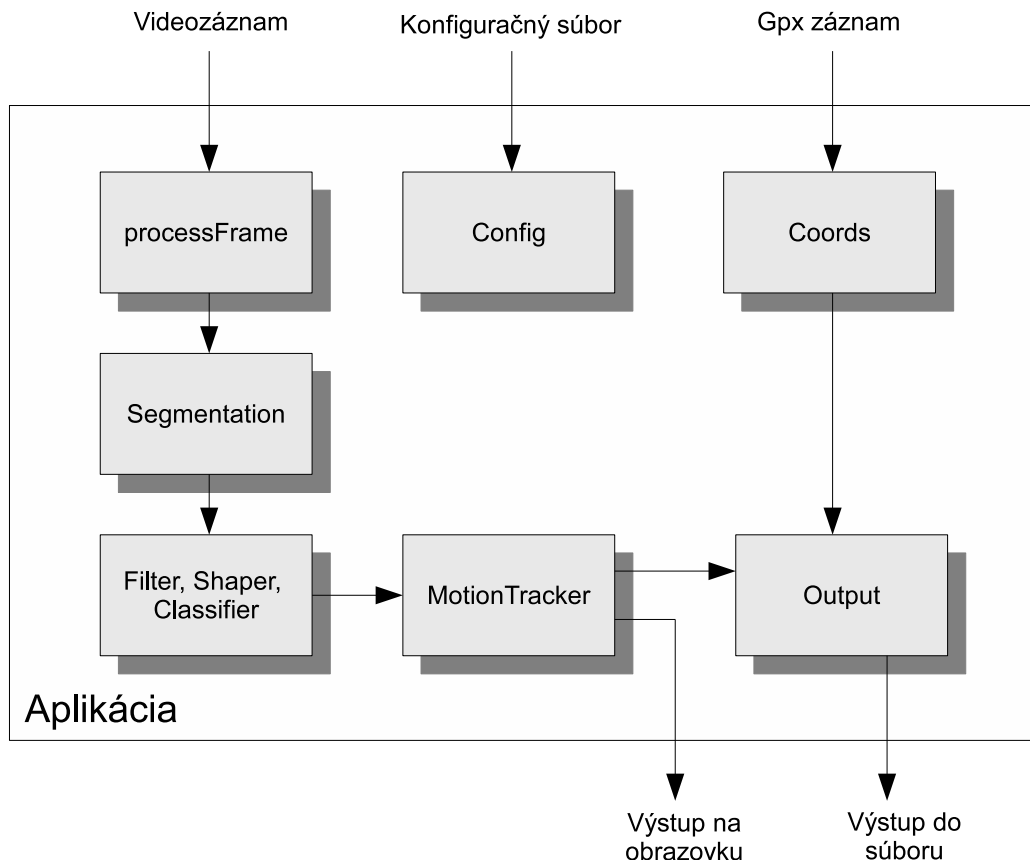
Pre detailnejší prehľad implementácie slúži vygenerovaná projektová dokumentácia. Tu uvediem len hlavné triedy ktoré zabezpečujú chod celého programu.

Trieda `Config` obsahuje konfiguráciu, ako sú prahy pre HSV segmentáciu, požadované fps, cesty k natrénovaným modelom pre SVM a cesty k tréningovej sade. Tieto hodnoty sa pri štarte načítajú z externého konfiguračného súboru. Pre rozšírenie tréningovej sady stačí editovať tento súbor a opäť spustiť tréningovanie. Nie je potreba zasahovať do zdrojového kódu.

Pre načítanie súradníc z gpx súboru slúži trieda `Coords`. Obsahuje taktiež metódy pre prepočet pozície vzhľadom k práve spracovávanému snímku.

Z tela globálnej funkcie `processFrame()`, ktorá je volaná pre každý spracovávaný snímok, sú postupne volané metódy tried zabezpečujúcu rozpoznávanie značiek v rámci snímku. Vytvorenie binárneho obrazu zabezpečuje trieda `Segmentation`. Obsahuje implementáciu HSV





Obr. 3.1: Hlavné triedy aplikácie pre rozpoznávanie značiek.

a RGB prahovania a jej rozhranie je navrhnuté tak aby bolo možné bez problémov pridať prípadnú inú segmentačnú metódu.

Na vytvorenú binárnu mapu sa aplikuje funkcia knižnice OpenCV - `cvFindContours()`, ktorá rozčlení obraz na zoznam potencionálnych objektov pomocou hľadania kontúr (obrysov). Nad týmto zoznamom prebieha klasifikácia, ktorú zabezpečujú triedy `Filter`, `Shaper` a `Classifier`. Tieto triedy sú SVM klasifikátory a v danom poradí odpovedajú kaskádovému návrhu uvedenom v predchádzajúcej kapitole (obrázok 2.5).

O sledovanie optického toku, ukladanie značiek a zápis do výstupného súboru sa stará trieda `MotionTracker`. Taktiež sa zabezpečuje vykresľovanie a zobrazovanie výstupu na obrazovku.

Poslednou triedou je trieda `Output`. Uchováva obsah výstupného súboru, ktorý sa vytvorí po spracovaní celého videa.

### Trénovacia sada

Trénovacia sada v konečnej verzii určená pre tréovanie SVM obsahuje 873 pozitívnych obrázkov (44 druhov značiek) a 1445 negatívnych obrázkov.

## 3.2 Zásuvný modul

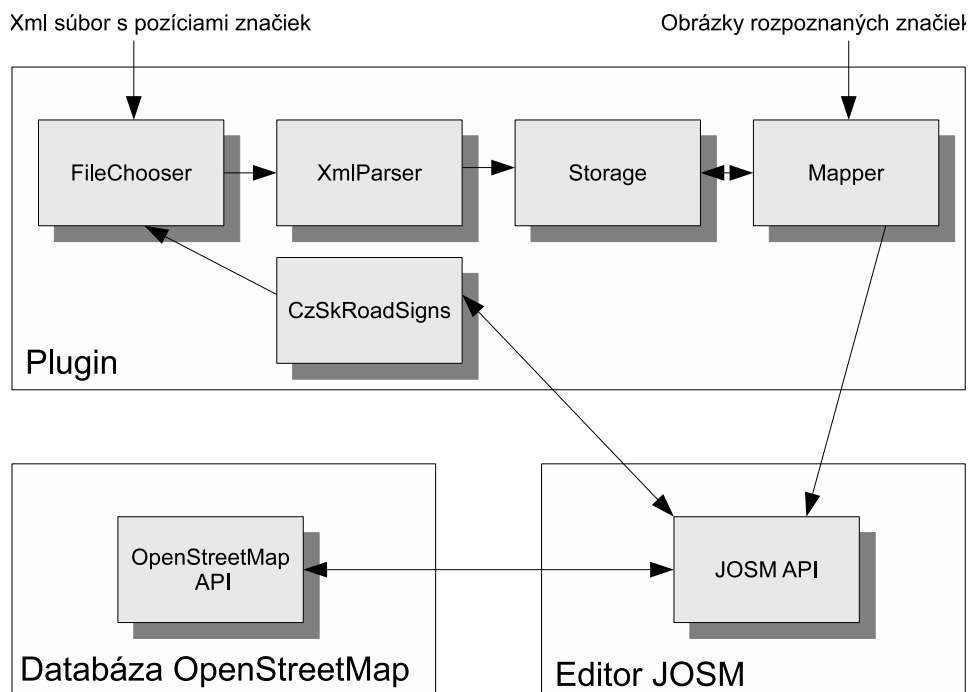
Zásuvný modul bol vyvíjaný v rovnakom jazyku v akom je napísaný editor JOSM, v Jave. Vývoj a testovanie prebehlo vo vývojovom prostredí Eclipse, za použitia editora JOSM vo verzii 5047. Pre správny beh pluginu je teda potrebná minimálne táto verzia. Pri implementácii bolo použité rozhranie editora, žiadne ďalšie knižnice neboli potrebné.

### Popis hlavných tried

Štartovacou triedou pluginu je trieda `CzSkRoadSigns`, ktorej konštruktor je invokovaný samotným editorom JOSM pri načítaní pluginov [26]. Trieda zabezpečuje zaregistrovanie pluginu do hlavného menu programu.

Trieda `FileChooser` sa stará o načítanie xml súboru, získaného ako výstup aplikácie pre rozpoznávanie značiek. Obsah súboru sa tu skontroluje a načíta sa do úložiska, do objektu typu `Storage`. Pri načítaní sa odstránia duplicitné záznamy, čiže výskyt rovnakej značky na rovnakých súradniciach bude uložený len raz. Kontrolu súboru a jeho parsovanie vykonáva trieda `XmlParser`.

Po načítaní a spracovaní potrebných dát je volaná trieda `Mapper`. Táto trieda prostredníctvom API editora vytvorí novú vrstvu do ktorej sa budú vkladať body reprezentujúce značky. Po jej vytvorení trieda sekvenčne prechádza záznamami uloženými v `Storage` a vykresľuje informácie o značke do okna pluginu. Ovládacie prvky umožňujú zmeniť typ značky, zadať doplňujúce informácie (napríklad hodnotu v km/h pri značke Najvyššia povolená rýchlosť) a pridať ju do mapy, prípadne preskočiť na ďalší záznam.



Obr. 3.2: Komunikácia pluginu s editorom a jeho hlavné triedy.

## Kapitola 4

# Testovanie

Testovanie prebehlo celkovo na jedenástich videách získaných v okolí Žiliny, Čadce, Turzovky a Brna. K desiatim z nich bol dostupný aj gpx záznam. Jedenásty videozáznam, kvôli absencii záznamu o geografickej polohe, nebol zahrnutý do posledného testu finálnej aplikácie. Všetky videá boli získané za dňa, no za odlišných svetelných podmienok. Testy boli vykonané na stroji s procesorom Intel Core2 Duo T5870 @ 2.00GHz a 4GB RAM.

### 4.1 Porovnanie segmentačných metód

Cieľom prvých testov bolo porovnať úspešnosť farebnej segmentácie v HSV a RGB priestore. V oboch prípadoch boli predmetom skúmania rovnaké vlastnosti: počet správne/chybné detekovaných značiek a počet správne/chybné klasifikovaných značiek. Prehľadávané boli rôzne intervaly prahov, pričom sa hľadala ich najoptimálnejšia kombinácia. Kvôli pomerne veľkej časovej náročnosti boli tieto testy vykonané s pomocou upraveného programu, odľahčenom o prácu s geografickými súradnicami, sledovaním optického toku a bez práce so samotným videom.

Spracovávané boli jednotlivé snímky získané zo všetkých dostupných videí. Každá značka, ktorá sa vo videozázname nachádzala, bola najprv zaznamenaná na dvoch snímkoch. Na jednom z väčšej vzdialenosti, na druhom z blízka, tak ako je znázornené na obrázku 4.1. Pre každú značku bol taktiež ručne zapísaný jej typ a pozícia v rámci obrazu do textového súboru. Celkovo bolo zozbieraných 353 snímkov, 212 so značkami a 141 snímkov, na ktorých sa značka nenachádzala. Úlohou programu bolo vykonávať nad snímkami detekciu a klasifikáciu a porovnávať výsledky so súborom obsahujúcim ich pozície a typy.



Obr. 4.1: Príklad rozpoznávaných značiek - z väčšej a z kratšej vzdialenosti.

## HSV

V prvom teste sa zisťovali rôzne hodnoty prahov, ktoré boli dané intervalom prípustných hodnôt. Horné prahy boli konštantne nastavené na maximálnu hodnotu. Dolné prahy sa pohybovali vo vymedzených intervaloch, pričom sa testovali všetky možné kombinácie. Pretože intervaly sú pomerne veľké, hodnoty sa zvyšovali vždy po istom kroku. Skúmané intervaly sú uvedené v tabuľke 4.1.

Tabuľka 4.1: Hodnoty prahov v teste č.1.

	<b>H_min</b>	<b>H_max</b>	<b>S_min</b>	<b>S_max</b>	<b>V_min</b>	<b>V_max</b>
<b>Interval</b>	< 160, 200 >	255	< 20, 60 >	255	< 4, 34 >	255
<b>Krok</b>	3	-	3	-	3	-

V tomto teste bolo preskúmaných 2744 rôznych kombinácií prahov na všetkých 352 snímkoch. Najlepšia kombinácia prahov, pri ktorých bola dosiahnutá najvyššia úspešnosť detekcie a klasifikácie, je uvedená v tabuľke 4.2.

Tabuľka 4.2: Výsledky testu č.1.

	<b>H_min</b>	<b>H_max</b>	<b>S_min</b>	<b>S_max</b>	<b>V_min</b>	<b>V_max</b>
<b>Hodnota</b>	178	255	35	255	7	255
<b>Úspešnosť detekcie</b>				58.22%		
<b>Úspešnosť klasifikácie</b>				86.30%		

## RGB

Pri RGB priestore som vyskúšal dva rôzne prístupy k segmentácii. Prvý spôsob určuje intervaly prípustných hodnôt pre jednotlivé farebné zložky, rovnako ako predchádzajúci test. V tomto prípade boli konštantné horná hranica  $R$  a dolné hranice  $G$  a  $B$ . Skúmané intervaly sú uvedené v tabuľke 4.3.

Tabuľka 4.3: Hodnoty prahov v teste č.2.

	<b>R_min</b>	<b>R_max</b>	<b>G_min</b>	<b>G_max</b>	<b>B_min</b>	<b>B_max</b>
<b>Interval</b>	< 50, 90 >	255	0	< 110, 130 >	0	< 70, 90 >
<b>Krok</b>	5	-	5	-	5	-

V tomto teste bolo preskúmaných 130 rôznych kombinácií prahov, opäť na všetkých snímkoch. Najlepšia kombinácia prahov, pri ktorých bola dosiahnutá najvyššia úspešnosť detekcie a klasifikácie, je uvedená v tabuľke 4.4.

Druhý spôsob segmentácie v RGB priestore využíva fakt, že ak má byť značka červená, musí byť hodnota červenej zložky väčšia ako hodnota zelenej a modrej zložky. Hodnota o akú má byť väčšia, čiže aký veľký má byť prah, bola predmetom skúmania. Všetkých 354 snímkov bolo podrobených testu pre každú hodnotu prahu z intervalu < 0, 100 >. Prah, pre ktorý bola dosiahnutá najvyššia úspešnosť detekcie a klasifikácie je uvedený v tabuľke 4.5.

Tabuľka 4.4: Výsledky testu č.2.

	R_min	R_max	G_min	G_max	B_min	B_max
<b>Hodnota</b>	55	255	0	110	0	135
<b>Úspešnosť detekcie</b>				33.80%		
<b>Úspešnosť klasifikácie</b>				72.22%		

Tabuľka 4.5: Výsledky testu č.3

<b>Hodnota prahu</b>	11
<b>Úspešnosť detekcie</b>	42.25%
<b>Úspešnosť klasifikácie</b>	86.67%

## 4.2 Zisťovanie optimálnych prahov pre HSV

Ako som už na začiatku kapitoly spomenul, každý videozáznam bol získaný za iných svetelných podmienok, tým pádom obsahuje rozličné hodnoty jasu. Preto bolo cieľom tohto testu zistiť ako moc sa líšia, teda zistiť najlepšie kombinácie prahov pre každý z 11-tich videozáznamov osobitne. Test prebiehal podobným spôsobom ako predchádzajúci. Za použitia už získaných výsledkov bola zostavená množina testovaných prahov. Horné prahy boli opäť konštantne nastavené na hodnotu 255. Množina dolných prahov hodnôt prahov je v tabuľke 4.6.

Tabuľka 4.6: Hodnoty prahov v teste č.4.

	H_min	S_min	V_min
<b>Množina hodnôt</b>	{168, 173, 178}	{30, 35, 40, 48}	{0, 7, 13}

Celkovo bolo v rámci 198 testov vyskúšaných 18 kombinácií pre každú sadu snímok zodpovedajúcu jednotlivým videozáznamom. Kombinácie pri, ktorých boli dosiahnuté najlepšie výsledky pre každý videozáznam zvlášť sú uvedené v tabuľke 4.7.

## 4.3 Testovanie aplikácie

V nasledujúcom teste som použil výsledky predošlého testu a to tak, že pre každý videozáznam boli nastavené prahy HSV segmentácie na hodnoty, pri ktorých boli dosiahnuté najlepšie výsledky. Skúmanou vlastnosťou bola okrem úspešnosti aj časová náročnosť programu potrebná k spracovaniu videa.

Videa boli pomocou finálnej verzie aplikácie na rozpoznávanie značiek podrobené testom v závislosti na počtu spracovávaných snímokoch za sekundu (fps). Testy prebehli pre 7 fps, 9 fps, 15 fps, 29 fps, čo znamená, že sa spracovával každý piaty, štvrtý, každý druhý respektíve každý snímok. Celková dĺžka desiatich videozáznamov, na ktorých bolo vykonávané testovanie dosahuje 45 minút a 23 sekúnd.

V tabuľke 4.8 sú priemerované výsledky úspešnosti a potrebný čas pre spracovanie všetkých desiatich videozáznamov. Dodatok B obsahuje tabuľky s úspešnosťou detekcie

Tabuľka 4.7: Najoptimálnejšie hodnoty prahov a úspešnosť pre test č.4.

	H_min	S_min	V_min	Úspešnosť detekcie	Úspešnosť klasifikácie
video1	178	30	7	66.67%	92.86%
video2	178	35	7	50.00%	80.00%
video3	178	30	7	42.86%	88.89%
video4	168	48	13	63.63%	92.86%
video5	178	48	13	100.00%	100.00%
video6	168	35	7	73.68%	100.00%
video7	168	30	7	41.67%	60.00%
video8	178	48	13	68.75%	77.27%
video9	178	35	7	69.77%	83.33%
video10	178	35	7	46.15%	100.00%
video11	178	40	7	75.00%	91.67%
<b>priemer</b>				<b>63.47%</b>	<b>87.90%</b>

a klasifikácie pre každý videozáznam zvlášť.

Tabuľka 4.8: Celková úspešnosť pre jednotlivé fps v teste č.5.

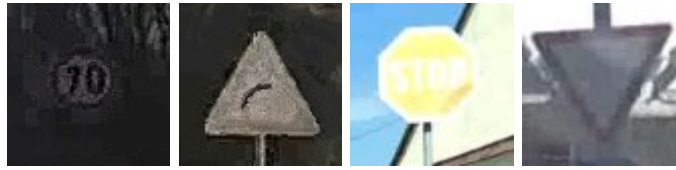
	7fps	9fps	15fps	29fps
<b>Celkovo značiek</b>	121	121	121	121
<b>Správne detekovaných značiek</b>	54	63	84	94
<b>Úspešnosť detekcie</b>	44.63%	52.07%	69.42%	77.69%
<b>Objekty nesprávne detekované ako značky</b>	16	20	34	58
<b>Správne klasifikovaných značiek</b>	50	60	80	91
<b>Úspešnosť klasifikácie</b>	92.59%	95.24%	95.24%	96.80%
<b>Potrebný čas</b>	1h 23m	1h 34m	2h 55m	5h 36m

## 4.4 Zhodnotenie dosiahnutých výsledkov

Počas testovania bola dosiahnutá najvyššia priemerná úspešnosť detekcie 77.69%. V niektorých prípadoch bola však táto hodnota veľmi nízka, v jednom testovacom videozázname (pri 7 fps) dokonca nebola detekovaná žiadna značka. Úspešnosť detekcie ovplyvňuje viacero faktorov. Testovaním bolo dokázané, že pri záznamoch, ktoré boli snímame oproti slnku boli dosiahnuté výrazne horšie výsledky.

V prvom rade je to teda kvalita videozáznamu a svetelné podmienky, pri ktorých bol získaný. Značky sú buď priveľmi osvetlené alebo sú osvetlené nedostatočne na to, aby ich farba spadala do prípustných rozsahov prahov (obrázok 4.2). Problematické je taktiež určenie optimálnych prahov. Ako ukázal test č.4, pre každý videozáznam je vhodná iná kombinácia.

V niektorých prípadoch nie je problémom priamo farba značky, ale farba jej pozadia. Ak má objekt za značkou červenú farbu (napríklad strecha domu), splynú v jeden objekt (obrázok 4.3).



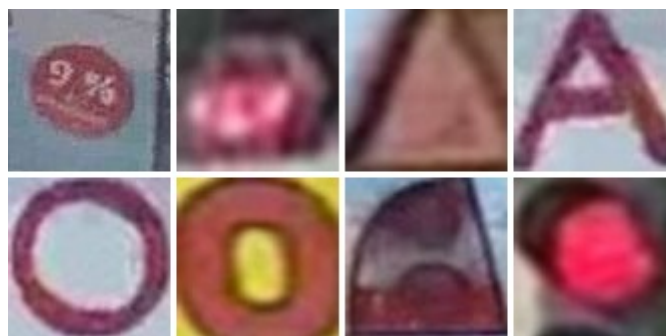
Obr. 4.2: Príklady nedetekovaných značiek.



Obr. 4.3: Splynutie značky s pozadím a vytvorenie binárnej mapy.

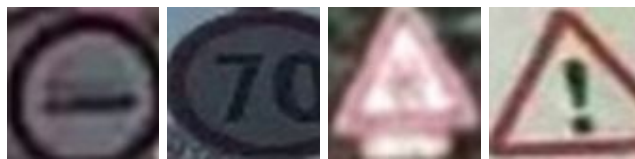
Posledným faktorom, ktorý sa do značnej miery podieľa na úspešnosti detekcie je sledovanie optického toku. Kvôli veľkému počtu objektov, ktoré sa vyskytovali na výstupe a neboli značkami som pridal podmienku, že značka sa zapíše len ak sa nájde na dvoch za sebou idúcich snímkoch. Pri vyšších rýchlostiach vozidla je však často predpovedaná poloha značky odlišná od jej reálneho umiestnenia v obraze a preto sa sledovanie optického toku preruší. To je aj hlavný dôvod, prečo sa so zvyšujúcim fps výrazne zvyšuje úspešnosť detekcie.

Čo sa klasifikácie týka, najväčším problémom sú objekty veľmi sa podobajúce na značky. Hoci takéto objekty boli zaradené do trénovacej sady, ich výskyt je stále pomerne vysoký. Na obrázku 4.4 sú príklady takýchto objektov.



Obr. 4.4: Objekty chybne klasifikované ako značky.

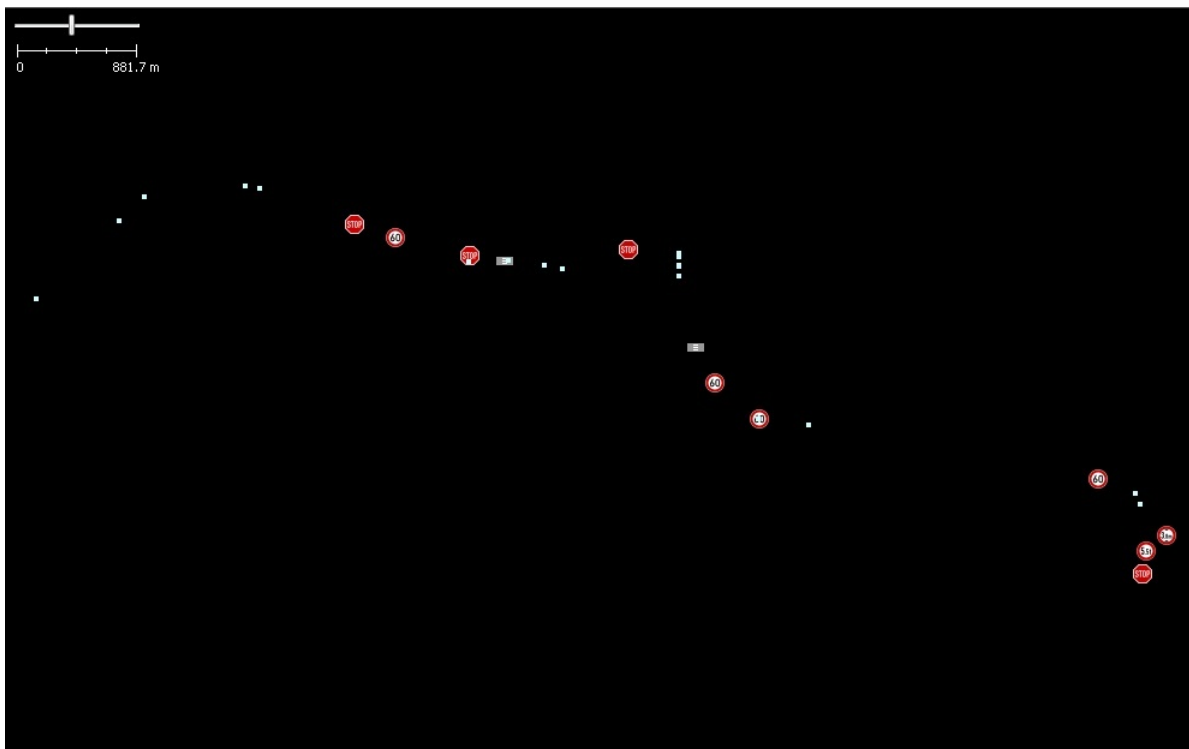
Okrem zamieňania objektov SVM klasifikátory v niektorých prípadoch určujú chybné aj typ značky. Dôvodmi sú veľká podobnosť niektorých značiek medzi sebou, prekrytie značky iným objektom alebo zlá kvalita obrázku spôsobená normalizovaním značky na 32x32 pixelov. Príklady chybné klasifikovaných značiek možno vidieť na obrázku 4.5



Obr. 4.5: Zľava doprava: Zákaz vjazdu autobusov chybné klasifikovaný ako Zákaz vjazdu bicyklov, Najvyššia povolená rýchlosť klasifikovaná ako Zákaz vjazdu vozidiel ktorých hmotnosť prekračuje danú hranicu, Sneh alebo poľadovica klasifikovaná ako Nebezpečenstvo šmyku a značka Iné nebezpečenstvo klasifikovaná ako Zúžená vozovka.

## 4.5 Import a uloženie dát na server OSM

Prostredníctvom pluginu bolo všetkých 94 značiek, ktoré boli rozpoznané aplikáciou pri 29 fps, importovaných do editora JOSM (obrázok 4.6). V editore je možnosť pracovať s viacerými vrstvami súčasne, pričom je možné použiť mapové podklady z už existujúcich dát umiestených na serveri OSM (obrázok 4.7). Všetky body boli ručne skontrolované a ich poloha bola skorigovaná vzhľadom k mapovým podkladom.

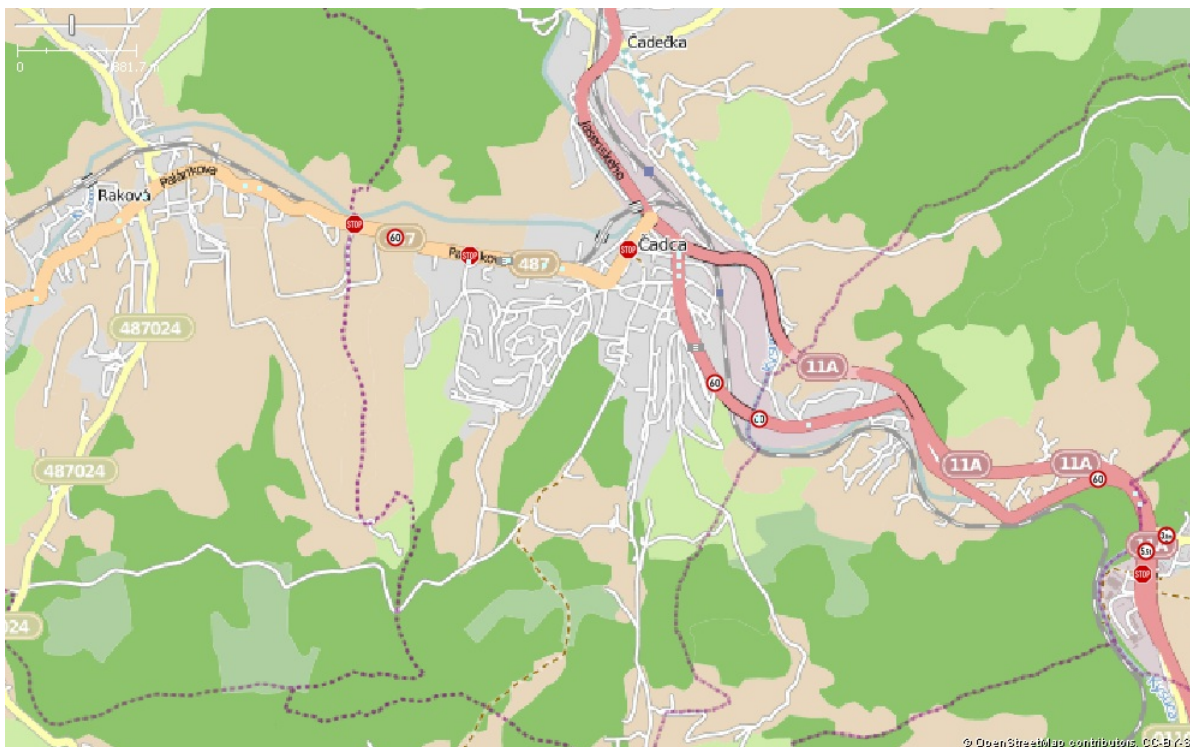


Obr. 4.6: Časť naimportovaných značiek v editore JOSM.

### Možné problémy pri ukladaní dát na server

Pred samotným nahrávaním dát na server vykonáva JOSM automatickú validáciu tagov. Pri tejto kontrole môžu vzniknúť problémy ktoré musia byť odstránené (**Errors**) alebo





Obr. 4.7: Časť naimportovaných značiek v editore JOSM aj s mapovými podkladmi Mapnik.

varovania (**warnings**), ktoré opravené byť nemusia. V priebehu testovania som narazil na niekoľko druhov varovaní, ktoré sa vyskytli pri pokuse o nahranie dát na server:

1. Dva rozličné body majú rovnaké súradnice (**duplicate nodes**).
2. Značka najvyššia povolená rýchlosť nie je priradená k ceste (**maxspeed without highway**).
3. Značka okružná križovatka nie je priradená k ceste (**junction without highway**).
4. Značka zákaz vjazdu nie je priradená k ceste (**oneway without highway**).
5. Značka zákaz vjazdu je priradená k bodu (**oneway tag on a node**).

Pretože plugin slúži len na pridávanie bodov, je odstránenie týchto problémov len na užívateľovi. Prvý problém je možné vyriešiť ručnou korekciou polohy. Zvyšné problémy sú spôsobené tým, že hoci je tieto značky možné vytvárať ako samostatné body, odporúča sa priradiť ich konkrétnej ceste (**way**). Tým sa platnosť obmedzení aplikuje na celú cestu. Na vyriešenie tohto problému je potrebné stiahnuť existujúce dáta zo servera OSM a ručne priradiť tieto body k už vytvoreným cestám.

# Záver

V tejto práci som sa zaoberal detekciou a rozpoznávaním dopravných značiek vo videu a ich zavedením do mapy. Počas jej tvorby som sa zoznámil s viacerými metódami pre spracovanie obrazu a projektom OpenStreetMap. Na základe týchto metód bola navrhnutá a implementovaná aplikácia pre detekciu a rozpoznávanie dopravných značiek. Navrhnutý a implementovaný bol taktiež zásuvný modul rozširujúci editor JOSM. Pomocou pluginu boli detekované značky zavedené do centrálnej databáze projektu OpenStreetMap.

Počas testovania som porovnal niekoľko spôsobov segmentácie v RGB a HSV farebnom priestore. Od pôvodného úmyslu vyskúšať aj iné spôsoby klasifikácie som upustil, pretože použitá metóda už od začiatku vykazovala pomerne dobré výsledky. Preto som sa zameral hlavne na rozšírenie trénovacej sady o viaceré typy značiek. V jej konečnej podobe, 44 druhov, pokrýva množinu všetkých červených dopravných značiek, ktoré sa v testovacích záznamoch vyskytli. Vďaka tomu boli aplikácia a plugin dôkladne otestované.

Na testovacej sade pozostávajúcej z desiatich videozáznamov bola dosiahnutá najvyššia priemerná úspešnosť detekcie 77.69% a úspešnosť klasifikácie 96.80%. Nevýhodou aplikácie je však čas potrebný na spracovanie videozáznamu a veľké množstvo chybných detekcií, ktoré sa zvyšuje s rastúcou úspešnosťou detekcie. Na server OpenStreetMap bolo za pomoci pluginu vložených celkovo 94 značiek.

Dosiahnuté výsledky a hlavné príčiny nedetekovania, prípadne chybného detekovania značiek a problémy spojené s nahrávaním dát na server som zhrnul v rámci kapitoly o testovaní.

Čo sa týka možných rozšírení, tých sa ponúka viacero. Ak by mal byť takto navrhnutý systém použiteľný v praxi, bolo by potrebné rozšíriť množinu detekovateľných objektov na všetky značky ktoré sú platné v danom štáte. Taktiež by musel spracovávať značky, ktoré rušia prechádzajúce zákazy. To znamená, že by bolo nutné vykonávať farebnú segmentáciu aj pre modrú a žltú, prípadne zelenú a šedú farbu. Okrem toho by mala byť aplikácia schopná rozpoznávať značky aj v noci. Keďže v takomto prípade sú väčšinou značky osvetlené a majú opäť rozličné úrovne jasů, znamenalo by to ďalšiu úpravu prahov. Použitie dokonalejšieho algoritmu pre sledovanie značky by mohlo odbúrať závislosť na vysokom fps, čím by sa výrazne zredukoval potrebný čas na spracovanie videa bez toho aby klesla úspešnosť detekcie. Pre širšie využitie by bolo taktiež vhodné zjednotiť aplikáciu tak, by obsahovala aj funkcionality pluginu, pretože práca s výstupnými súbormi je v istom smere nepraktická. V tomto prípade by však aplikácia musela zabezpečovať aj komunikáciu so serverom, ktorú v tejto práci pokrýva editor JOSM.

# Literatúra

- [1] Domovská stránka projektu *OpenStreetMap* [online]. 2012 [cit. 21. ledna 2012].  
Dostupné na: <<http://www.openstreetmap.org/>>.
- [2] SHNEIER, M. *Road Sign Detection and Recognition* [online]. 2005 [cit. 21. ledna 2012].  
Dostupné na:  
<[http://www.isd.mel.nist.gov/documents/shneier/Road\\_Sign\\_Detection.pdf](http://www.isd.mel.nist.gov/documents/shneier/Road_Sign_Detection.pdf)>.
- [3] ZAKIR, U., LEONCE, A. N. J. a EDIRISINGHE, E. A. Road Sign Detection and Recognition by using Local Energy Based Histogram (LESH). *International Journal of Image Processing*. 2010, roč. 4. S. 567–583. ISSN 1985-2304.
- [4] ESTEVEZ, L. a KEHTARNAVAZ, N. A real-time histogrammic approach to road sign recognition. In *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*. 1996. S. 95–100. ISBN 0780332008.
- [5] HOFFMANN, G. *CIE Lab Color Space* [online]. 2009 [cit. 21. ledna 2012]. Dostupné na: <<http://www.fho-emden.de/~hoffmann/cielab03022003.pdf>>.
- [6] FLEYEH, H. *Traffic and Road Sign Recognition*. Edinburgh: Napier University, 2008. Diplomová práce.
- [7] HLAVÁČ, V. a ŠONKA, M. *Počítačové vidění*. Praha 1, Dlouhá 39: Grada, a.s., 1992. 272 s. ISBN 80-85424-67-3.
- [8] GHICA, D., LU, S. W. a YUAN, X. Recognition of traffic signs by artificial neural network. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*. Nov/dec 1995. S. 1444–1449.
- [9] LONG, C., QINGQUAN, L., MING, L. et al. Traffic Sign Detection and Recognition for Intelligent Vehicle. In *IEEE Intelligent Vehicles Symposium (IV)*. 2011. S. 908–913. ISBN 1931-0587.
- [10] SUARD, F., RAKOTOMAMONJY, A., BENSRAHAI, A. et al. Pedestrian Detection using Infrared images and Histograms of Oriented Gradients. In *Intelligent Vehicles Symposium, 2006 IEEE*. 2006. S. 206–212.
- [11] STERGIU, C. a SIGANOS, D. Neural Networks. *SURPRISE 96*. 1996, roč. 4.  
Dostupné na:  
<[http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html)>.
- [12] VICEN BUENO, R., GIL PITA, R., ROSA ZURERA, M. et al. Multilayer Perceptrons Applied to Traffic Sign Recognition Tasks. In *Computational Intelligence and*

- Bioinspired Systems*. 2005. S. 865–872. Lecture Notes in Computer Science, sv. 3512. ISBN 978-3-540-26208-4.
- [13] ŽIŽKA, J. *Vybrané slajdy k predmetu Strojové učení* [online]. 2005 [cit. 24. ledna 2012]. Faculty of Information Technology Masaryk University Brno. Dostupné na: <[http://is.muni.cz/el/1433/podzim2006/PA034/09\\_SVM.pdf](http://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf)>.
- [14] *Wikipedia: The Free Encyclopedia* [online]. 2012 [cit. 23. ledna 2012]. Dostupné na: <[http://en.wikipedia.org/wiki/File:Svm\\_separating\\_hyperplanes.png](http://en.wikipedia.org/wiki/File:Svm_separating_hyperplanes.png)>.
- [15] MADZAROV, G., GJORGJEVIKJ, D. a CHORBEV, I. A Multi-class SVM Classifier Utilizing Binary Decision Tree. *Informatika*. 2008, roč. 33, č. 2. S. 233–241. ISSN 1854-3871.
- [16] SHI, M., WU, H. a FLEYEH, H. Support vector machines for traffic signs recognition. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence)*. *IEEE International Joint Conference on*. 2008. S. 3820–3827.
- [17] *Spravodajský portál idnes: Samsung Galaxy S II* [online]. 2012 [cit. 25. dubna 2012]. Dostupné na: <[http://i.idnes.cz/11/033/c15/V0K39e8e5\\_Samsung\\_Galaxy\\_S.II\\_01.JPG](http://i.idnes.cz/11/033/c15/V0K39e8e5_Samsung_Galaxy_S.II_01.JPG)>.
- [18] ZAKIR, U., LEONCE, A. N. a EDIRISINGHE, E. A. Road Sign Segmentation Based On Colour Spaces: A Comparative Study. In *11th IASTED International Conference Computer Graphics and Imaging (CGIM 2010)*. 2010. ISBN 978-0-88986-836-6.
- [19] PORIKLI, F. Integral histogram: a fast way to extract histograms in Cartesian spaces. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. 2005. ISSN 1063-6919.
- [20] VIOLA, P. a JONES, M. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. 2001. ISSN 1063-6919.
- [21] YILDIZ, C. *An Implementation on Histogram of Oriented Gradients for Human Detection* [online]. [cit. 1. května 2012]. Dostupné na: <<http://www.cs.bilkent.edu.tr/~cansin/projects/cs554-vision/pedestrian-detection/pedestrian-detection-paper.pdf>>.
- [22] *INRIA Person Dataset* [online]. [cit. 9. května 2012]. Dostupné na: <<http://pascal.inrialpes.fr/data/human>>.
- [23] TOTH, Š. Rozpoznávání dopravných značek a ich použitie v mapových aplikáciách. In *Sborník - Sympozium GIS Ostrava 2011*. 2011.
- [24] *Zoznam používaných tagov pre OpenStreetMap* [online]. 2012 [cit. 11. května 2012]. Dostupné na: <[http://wiki.openstreetmap.org/wiki/Map\\_features](http://wiki.openstreetmap.org/wiki/Map_features)>.
- [25] *Domovská stránka knižnice OpenCV* [online]. 2012 [cit. 30. dubna 2012]. Dostupné na: <<http://opencv.willowgarage.com/wiki/>>.

- [26] *Príručka pre vytváranie pluginov pre JOSM* [online]. 2012 [cit. 30. dubna 2012].  
Dostupné na:  
<<http://josm.openstreetmap.de/wiki/DevelopersGuide/DevelopingPlugins>>.

## Dodatok A

# Zoznam podporovaných značiek

### Výstražné značky

1. Dvojitá zákruta, prvá vľavo
2. Dvojitá zákruta, prvá vpravo
3. Iné nebezpečenstvo
4. Nebezpečenstvo šmyku
5. Nebezpečné klesanie
6. Nebezpečné stúpanie
7. Nerovnosť vozovky
8. Obojsmerná premávka
9. Pozor deti
10. Priechod pre chodcov
11. Sneh alebo poľadovica
12. Spomaľovací prah
13. Svetelné signály
14. Zákruta vľavo
15. Zákruta vpravo
16. Železničné priecestie bez závor
17. Zúžená vozovka sprava
18. Zúžená vozovka zľava
19. Zúžená vozovka z oboch strán

### Značky upravujúce prednosť v jazde

1. Daj prednosť v jazde
2. Stop
3. Okružná križovatka
4. Križovatka
5. Križovatka s vedľajšou cestou
6. Prednosť protiídúcich vozidiel

## Zákazové značky

1. Iný zákaz
2. Najvyššia povolená rýchlosť
3. Zákaz odbočenia vľavo
4. Zákaz odbočenia vpravo
5. Zákaz otáčania
6. Zákaz predchádzania
7. Zákaz predchádzania pre nákladné automobily
8. Zákaz státia
9. Zákaz vjazdu
10. Zákaz vjazdu v oboch smeroch
11. Zákaz vjazdu autobusov
12. Zákaz vjazdu bicyklov
13. Zákaz vjazdu nákladných automobilov
14. Zákaz vjazdu vozidiel, ktorých šírka presahuje vyznačenú hranicu
15. Zákaz vjazdu vozidiel, ktorých výška presahuje vyznačenú hranicu
16. Zákaz vjazdu vozidiel, ktorých okamžitá hmotnosť presahuje vyznačenú hranicu
17. Zákaz vjazdu všetkých motorových vozidiel
18. Zákaz vstupu chodcov
19. Zákaz zastavenia

## Dodatok B

# Výsledky testov

Pomenovanie videozáznamov v tabuľkách odpovedá ich názvom v priloženom DVD.

### Výsledky testu č.5

Tabuľka B.1: Výsledky testu č.5 pre video1.

	<b>7fps</b>	<b>9fps</b>	<b>15fps</b>	<b>29fps</b>
<b>Značiek vo videozázname</b>	11	11	11	11
<b>Správne detekovaných značiek</b>	6	6	9	9
<b>Správne klasifikovaných značiek</b>	4	6	9	9
<b>Objekty nesprávne detekované ako značky</b>	1	1	2	6
<b>Úspešnosť detekcie</b>	54.55%	54.55%	81.82%	81.82%
<b>Úspešnosť klasifikácie</b>	66.67%	100.00%	100.00%	100.00%

Tabuľka B.2: Výsledky testu č.5 pre video2.

	<b>7fps</b>	<b>9fps</b>	<b>15fps</b>	<b>29fps</b>
<b>Značiek vo videozázname</b>	5	5	5	5
<b>Správne detekovaných značiek</b>	2	3	3	5
<b>Správne klasifikovaných značiek</b>	2	3	3	5
<b>Objekty nesprávne detekované ako značky</b>	0	0	0	2
<b>Úspešnosť detekcie</b>	40.00%	60.00%	60.00%	100.00%
<b>Úspešnosť klasifikácie</b>	100.00%	100.00%	100.00%	100.00%



Tabuľka B.3: Výsledky testu č.5 pre video3.

	7fps	9fps	15fps	29fps
Značiek vo videozázname	12	12	12	12
Správne detekovaných značiek	2	5	10	11
Správne klasifikovaných značiek	2	4	9	10
Objekty nesprávne detekované ako značky	0	0	0	1
Úspešnosť detekcie	16.67%	41.67%	83.33%	91.67%
Úspešnosť klasifikácie	100.00%	80.00%	90.00%	90.91%

Tabuľka B.4: Výsledky testu č.5 pre video4.

	7fps	9fps	15fps	29fps
Značiek vo videozázname	13	13	13	13
Správne detekovaných značiek	6	6	8	9
Správne klasifikovaných značiek	6	6	7	8
Objekty nesprávne detekované ako značky	1	2	5	7
Úspešnosť detekcie	46.15%	46.15%	61.54%	69.23%
Úspešnosť klasifikácie	100.00%	100.00%	87.50%	88.89%

Tabuľka B.5: Výsledky testu č.5 pre video5.

	7fps	9fps	15fps	29fps
Značiek vo videozázname	3	3	3	3
Správne detekovaných značiek	2	3	3	3
Správne klasifikovaných značiek	2	3	3	3
Objekty nesprávne detekované ako značky	0	0	1	1
Úspešnosť detekcie	66.67%	100.00%	100.00%	100.00%
Úspešnosť klasifikácie	66.67%	100.00%	100.00%	100.00%

Tabuľka B.6: Výsledky testu č.5 pre video6.

	7fps	9fps	15fps	29fps
Značiek vo videozázname	13	13	13	13
Správne detekovaných značiek	7	8	9	10
Správne klasifikovaných značiek	7	8	9	10
Objekty nesprávne detekované ako značky	2	2	6	6
Úspešnosť detekcie	53.85%	61.54%	69.23%	76.92%
Úspešnosť klasifikácie	100.00%	100.00%	100.00%	100.00%

Tabuľka B.7: Výsledky testu č.5 pre video7.

	7fps	9fps	15fps	29fps
Značiek vo videozázname	9	9	9	9
Správne detekovaných značiek	0	2	3	4
Správne klasifikovaných značiek	0	2	2	3
Objekty nesprávne detekované ako značky	0	0	0	1
Úspešnosť detekcie	0.00%	22.22%	33.33%	44.44%
Úspešnosť klasifikácie	-	100.00%	66.67%	75.00%

Tabuľka B.8: Výsledky testu č.5 pre video8.

	7fps	9fps	15fps	29fps
Značiek vo videozázname	20	20	20	20
Správne detekovaných značiek	14	15	17	17
Správne klasifikovaných značiek	13	14	16	16
Objekty nesprávne detekované ako značky	4	5	9	16
Úspešnosť detekcie	70.00%	75.00%	85.00%	85.00%
Úspešnosť klasifikácie	92.86%	93.33%	94.12%	94.12%

Tabuľka B.9: Výsledky testu č.5 pre video9.

	7fps	9fps	15fps	29fps
Značiek vo videozázname	27	27	27	27
Správne detekovaných značiek	11	11	17	21
Správne klasifikovaných značiek	10	10	17	21
Objekty nesprávne detekované ako značky	1	0	3	4
Úspešnosť detekcie	40.74%	40.74%	62.96%	77.78%
Úspešnosť klasifikácie	90.90%	90.90%	100.00%	100.00%

Tabuľka B.10: Výsledky testu č.5 pre video10.

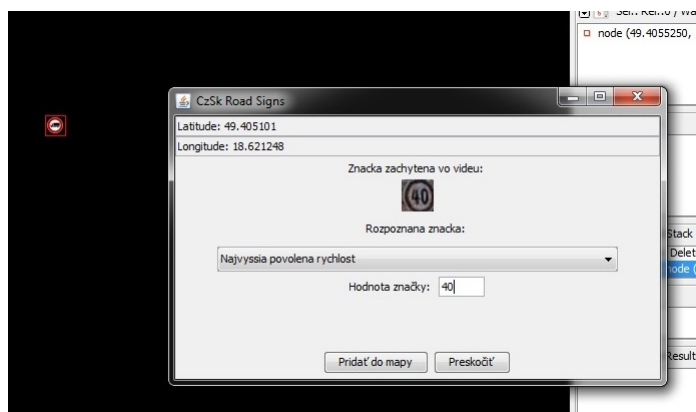
	7fps	9fps	15fps	29fps
Značiek vo videozázname	8	8	8	8
Správne detekovaných značiek	4	4	5	5
Správne klasifikovaných značiek	4	4	5	5
Objekty nesprávne detekované ako značky	7	10	12	14
Úspešnosť detekcie	50.00%	50.00%	62.50%	62.50%
Úspešnosť klasifikácie	100.00%	100.00%	100.00%	100.00%

## Dodatok C

# Ukážky aplikácie a pluginu



Obr. C.1: Ukážka grafického výstupu aplikácie. Zelený štvorec vymedzuje oblasť, v ktorej sa musí značka nachádzať, aby sa pokračovalo v jej sledovaní.



Obr. C.2: Okno pluginu pre import bodov do editora JOSM.

## Dodatok D

# Obsah priloženého DVD

- **Application** - adresár s aplikáciou
  - **bin** - vlastná aplikácia a potrebné knižnice pre platformu Windows
  - **doc** - programová dokumentácia
  - **src** - zdrojové kódy aplikácie
  - **Config** - konfiguračný súbor
  - **Images** - trénovacie dáta
  - **Output** - výstupy testovania
  - **SVM** - natrénované modely pre SVM
  - **Video** - testovacie videá
- **Latex** - zdrojové súbory technickej správy v latexu
- **Maps** - ukážky máp z prostredia editora JOSM
- **Plugin** - adresár s pluginom
  - **doc** - programová dokumentácia
  - **jar** - skompilovaný plugin
  - **src** - zdrojové kódy pluginu
- **technicka-sprava.pdf** - technická správa vo formáte PDF