

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

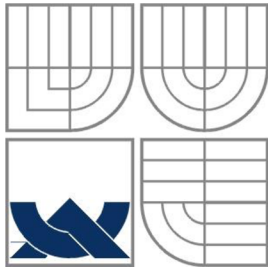
**MAPOVÁNÍ DOPRAVNÍHO ZNAČENÍ ZA POMOCI
METOD ZPRACOVÁNÍ OBRAZU**

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

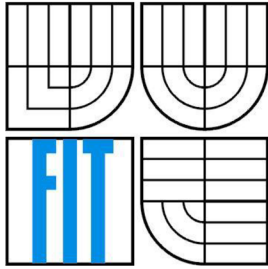
AUTOR PRÁCE
AUTHOR

VLADIMÍR LEŠKO

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

MAPOVÁNÍ DOPRAVNÍHO ZNAČENÍ ZA POMOCI METOD ZPRACOVÁNÍ OBRAZU

MAPPING OF ROAD SIGNS USING IMAGE PROCESSING METHODS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VLADIMÍR LEŠKO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ NOVOTNÝ

BRNO 2015

Abstrakt

Bakalárska práca sa zaoberá rozpoznávaním dopravných značiek vo videozázname, získavaním GPS záznamov a vložением dopravných značiek s odpovedajúcimi GPS súradnicami do centrálnej databázy OSM. V jednotlivých kapitolách sú postupne predstavené metódy detekcie a klasifikácie obrazu, oboznámenie s projektom OSM a jeho zásuvným modulom JOSM, návrh a samotná implementácia výslednej aplikácie a zásuvného modulu. V poslednej kapitole sú predstavené výsledky testovania aplikácie na 10 videozáznamoch. Záver obsahuje zhodnotenie práce s možnými rozšíreniami..

Abstract

The following Bachelor's Thesis deals with traffic sign recognition in video clips, obtaining GPS records and inserting traffic signs with accurate GPS co-ordinates into OSM central database. In each of Thesis' chapters are in sequence introduced methods of image detection and classification, consecutive familiarization with OSM project and its JOSM plug-in. Afterwards there will be presented an application and an applicative plug-in design and implementation process. The final chapter is represented as the results of testing the application in 10 video clips. The conclusion contains an evaluation of the Thesis with possible extensions..

Klíčová slova

Počítačové vidění, Rozpoznávání dopravního značení, Barevná segmentace, Detekce a klasifikace objektu z obrazu, HOG, SVM, OpenCV, JOSM, OpenStreetMap

Keywords

Computer vision, Traffic sign recognition, Color segmentation, Object detection and object classification, HOG, SVM, OpenCV, JOSM, OpenStreetMap

Citace

Vladimír Leško: Mapování dopravního značení za pomoci metod zpracování obrazu, bakalářská práce, Brno, FIT VUT v Brně, 2015

Mapování dopravního značení za pomoci metod zpracování obrazu

Prohlášení

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

.....
Vladimír Leško
18.května.2015

Poděkování

Poděkovat by som sa chcel v prvom rade svojmu vedúcemu práce , pánu Ing. Tomášovi Novotnému za poskytnutie možnosti pracovať na tejto bakalárskej práce, za jeho užitočné a cenné rady počas celého semestra. Ďalej by som sa chcel poďakovať svojmu otcovi za poskytnutie dostatočného počtu videozáznamov a GPS záznamov.

© Vladimír Leško, 2015

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	Teoretická časť	3
2.1	Open Street Map (OSM)	3
2.2	Java Open Street Map Editor (JOSM).....	3
2.3	Farebné modely:.....	5
2.4	Predspracovanie obrazu.....	8
2.5	Segmentácia obrazu a metódy segmentácie	10
2.6	Morfologické operácie	13
2.7	Extrakcia príznakov.....	14
2.8	Detekcia a sledovanie objektov v obraze	16
2.9	Rozpoznávanie (klasifikácia) obrazu	17
3	Návrh riešenia.....	20
3.1	Schéma aplikácie slúžiaca pre spracovanie obrazu	20
3.2	Schéma zásuvného modulu	21
3.3	Získanie vstupných dát.....	22
3.4	Návrh aplikácie pre spracovanie obrazu	23
4	Implementácia	27
4.1	Aplikácia pre rozpoznávanie dopravných značiek z videa.....	27
4.2	Zásuvný modul (Plugin) pre editor JOSM	31
4.3	Testovanie	32
4.4	Výsledky testovania	33
4.5	Zhodnotenie výsledkov testovania	35
5	Záver.....	36
Príloha A	Zoznam rozpoznávaných značiek :	40
Príloha B	Výsledky testov	42
Príloha C	Obsah DVD	46

1 Úvod

V dnešnej dobe patrí automobilová doprava medzi najbežnejšie formy dopravy. Stretávame sa s ňou dennodenné v rôznych formách, či už je to cestou do práce, do školy alebo pri iných každodenných činnostiach. Každý účastník cestnej premávky (vodič) musí dodržiavať pravidlá cestnej premávky štátu v ktorom sa momentálne nachádza. Je to hlavne z dôvodu vlastnej bezpečnosti a bezpečnosti voči ostatným účastníkom cestnej premávky.

Týmto sa dostávame k dopravným značkám, ktoré musí poznať každý vodič aby sa vedel podľa nich orientovať v cestnej premávke a aby dodržiaval zákon o cestnej premávke. Napriek znalosti dopravných značiek vodičom sa môže stať, že vodič môže niekedy prehliadnuť nejakú dopravnú značku a zareagovať nesprávne v danej situácii. Čím môže nielenže obmedziť a ohroziť účastníkov cestnej premávky ale v najhoršom prípade aj spôsobiť dopravnú nehodu. To bol jeden z hlavných dôvodov prečo začali rôzne automobilky vyvíjať inteligentné systémy, ktoré by boli schopné rozpoznať dopravnú značku a vo vhodný čas na ňu upozorniť vodiča a tým pádom predísť vzniku nebezpečnej dopravnej situácie. Tí čo sledujú dianie v automobilovom priemysle, iste vidia snahu výrobcov áut (BMW) ale aj firiem z IT sektoru (GOOGLE) vyvinúť autonómne riadené vozidlo. Jednou z najdôležitejších súčasti takéhoto informačného systému je vizuálne spracovanie obrazu.

To je vlastne oblasť, ktorou sa zaoberám vo svojej bakalárskej práci. Cieľom je navrhnuť aplikáciu, ktorá je schopná vo vstupnom videozázname nájsť a rozpoznať dopravné značky, a následne zavedenie GPS pozície takéhoto značenia do mapy. V mojej bakalárskej práci by som sa chcel hlavne zamerať na identifikáciu týchto typov značiek: výstražné, zákazové, príkazové a niektoré informatívne dopravné značky. Po nájdení a zidentifikovaní dopravnej značky, bude potrebné určiť jej GPS pozíciu a následne bude môcť užívateľ prostredníctvom užívateľského rozhrania rozhodnúť o zavedení dopravnej značky do mapy.

2 Teoretická časť

V tejto časti predstavím projekt Open Street Map, ktorý slúži pre vkládanie a editovanie geografických dát spolu s editorom Java Open Street Map Editorom (JOSM). V ďalších kapitolách predstavím základné farebné modely, ktoré sa používajú pri práci s obrazom, niektoré metódy predspracovania obrazu, ktoré sú schopné potlačiť neželané deformácie v obraze. Takisto tu budú spomenuté aj základné segmentačné metódy, s dôrazom na metódu prahovania a detekciu hrán, ktoré sa najčastejšie používajú pri segmentácii dopravného značenia.

Budem sa tu venovať aj morfológickým operáciám a metódam extrakcie príznakov, ktoré slúžia ako vstup pre klasifikátorov, ako aj metódam sledovania objektov v obraze, ktoré budú v praxi použité na to aby sa rovnaká dopravná značka rozpoznala v sérii snímok len raz. V poslednej kapitole predstavím najpoužívanejšie klasifikačné metódy.

2.1 Open Street Map (OSM)

Open Street Map (OSM) je projekt slúžiaci pre vkladanie a editovanie geografických dát. Za zakladateľa sa považuje Steve Coast, ktorý prišiel s danou myšlienkou v roku 2004. V roku 2014 je počet registrovaných užívateľov vyše 1,9 milióna [1]. Používatelia môžu podporiť projekt získavaním dát nasledujúcimi spôsobmi:

- Pomocou rôznych GPS zariadení
- Satelitnými snímkami
- Vlastnými audio záznamami
- Importu z voľne dostupných databáz

Tieto dáta sú následne uložené v centrálnej databáze, ktorá spadá pod licenciu Open Database Licence. K týmto údajom majú prístup všetci užívatelia využívajúci OSM. Získane dáta sa následne využívajú k tvorbe rôznych typov máp. Dáta slúžiace pre tvorbu máp sa na server vkladajú viacerými spôsobmi. Najčastejšie sa na to využíva editor JOSM.

2.2 Java Open Street Map Editor (JOSM)

Java Open Street Map Editor (JOSM) je editor slúžiaci pre prácu s geografickými údajmi, ktorý svojimi pluginmi ponúka veľa nových možností pre prácu s projektom OSM. Aktuálne podporuje prácu s GPX a NMEA záznamami z ciest, či s GPX dátami z databáze OSM, stiahnutím a úpravou existujúcich prvkov mapy:

- Bodov
- Ciest
- Metadát
- Relácií

Tento editor bol vytvorený Immanuelom Scholzom a jeho popularita na vzostupe. Je to dôsledok mnohých faktorov, kde medzi najdôležitejší patrí určite možnosť spúšťania JOSM editoru na rôznych platformách ako sú (MS Windows, Linux, MAC OS X). Je napísaný v jazyku JAVA, má intuitívne grafické prostredie a zvládne všetko, čo môže začínajúci ale aj pokročilý mapper potrebovať [2].

2.2.1 Vlastnosti editoru JOSM

JOSM dokáže pracovať v *off-line* režime. To znamená, že všetky zmeny sa vykonávajú len lokálne a až do potvrdenia používateľom sa na OSM serveri neprejavujú. Táto vlastnosť robí z editora JOSM ideálny nástroj na pochopenie celej filozofie mapovania a zároveň ponúka možnosť učiť sa mapovať skúšaním na skutočnej mape bez hrozby poškodenia či zničenia cudzej práce (samozrejme za predpokladu, že testovacie dáta používateľ omylom alebo schválne nepotvrdí na OSM server) [3].

Jeho ďalšou výhodou je široká škála pluginov. Keďže je JOSM editor napísaný v jazyku Java, tak táto skutočnosť patrí medzi jeho hlavné nevýhody pretože JOSM môžu využívať len užívatelia, ktorí majú nainštalovanú Javu. JOSM momentálne existuje v 2 verziách:

- *Tested*: je veľmi dobre otestovaná, takže by nemala obsahovať chyby, nevýhodou je, že neobsahuje najnovšie funkcie.
- *Latest*: obsahuje najnovšie funkcie (v podstate je aktualizovaná takmer dennodenne), ale občas obsahuje chybu a vtedy treba použiť staršiu verziu, prípadne verziu **tested**[4].

2.2.2 JOSM API

Globálne objekty prístupné z hlavnej triedy Main v JOSM API

- *Main.parent*
- *Main.pref*:
- *Main.proj*
- *Main.map.mapView*

Za zmienku stoja takisto aj niektoré JOSM udalosti, ktoré reagujú na zmeny vyvolané samotným JOSM a používajú sa pri tvorbe pluginov sú to:

- *Layer change events*
- *Selection change listener*
- *Data Change events*
- *Preference change listener*

2.2.3 Práca s editorom JOSM a formát dát

Prvým krokom pri práci s editorom JOSM je spôsob pridávania alebo editácie uzlov (*node*) a ciest (*way*) v OSM. Uzol (*node*) je určený GPS súradnicami a je reprezentovaný jedným bodom v OSM. JOSM nám umožňuje vložiť samostatné uzly ,tie sa využívajú tam kde je križovatka alebo kde dochádza k zmene organizácie dopravy, alebo vložiť uzly za sebou ktoré vytvoria cestu (*way*). Cesta (*way*) je usporiadaný zoznam za sebou nasledujúcich uzlov.

Uzly a cesty samé o sebe nemajú príliš veľké využitie pokiaľ im nie sú pridané vlastnosti. Vlastnosti jednotlivých uzlov a ciest sa pridávajú pomocou atribútov (*tagov*).

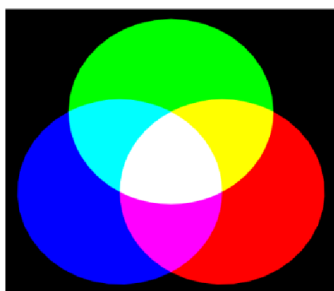
Atribúty pozostávajú z 2 zložiek

- Kľúč (*key*)
- Hodnota (*value*)

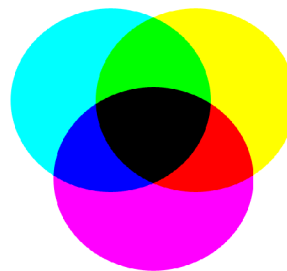
2.3 Farebné modely:

Farebný model [5] opisuje základné farby a schémy miešania týchto základných farieb do výslednej farby. Matematicky môžeme farebnému modelu definovať jeho antropomorfný farebný priestor (alebo aspoň jeho aproximáciu), ktorý je absolútne merateľný v reálnych fyzikálnych jednotkách. Farebné modely potom popisujú v rámci daného farebného priestoru predovšetkým relatívne pomery jednotlivých bodov snímky medzi sebou vzájomne. Farebné modely sa síce snažia napodobniť farby čo najvernejšie ale v praxi sa však používajú modely u ktorých je vybraný vhodný kompromis medzi presnosťou podania farebného dojmu a zložitosti, obtiažnosti konkrétneho modelu. Poznáme 2 metódy miešania farieb:

- **Aditívne miešanie farieb:**
Jednotlivé farebné zložky farieb sa sčítajú a výsledok je svetlo väčšej intenzity. Aditívne skladanie farieb pracuje s tromi základnými farbami červenou, zelenou, modrou . Ich zmiešaním vznikne biela farba.
- **Subtraktívne miešanie farieb:**
S každou pridanou farbou sa odoberá časť pôvodného svetla. Svetlo prechádza jednotlivými farebnými vrstvami a je stále viac a viac pohlcované. Základné farby sú azúrova, purpúrova, žltá. Ich zmiešaním vznikne čierna farba



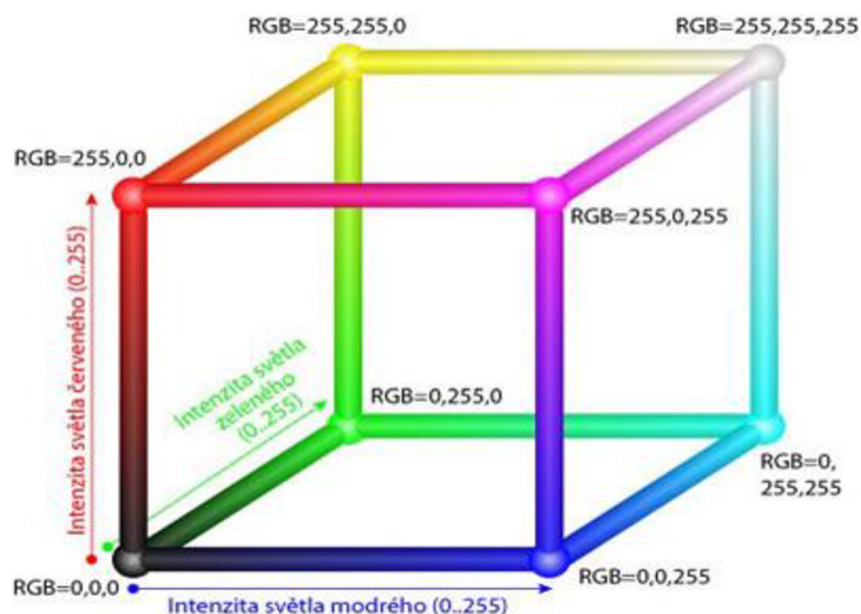
Obr.1: Aditívne miešanie. Zdroj [5].



Obr.2: Subtraktívne miešanie. Zdroj [5].

2.3.1 RGB

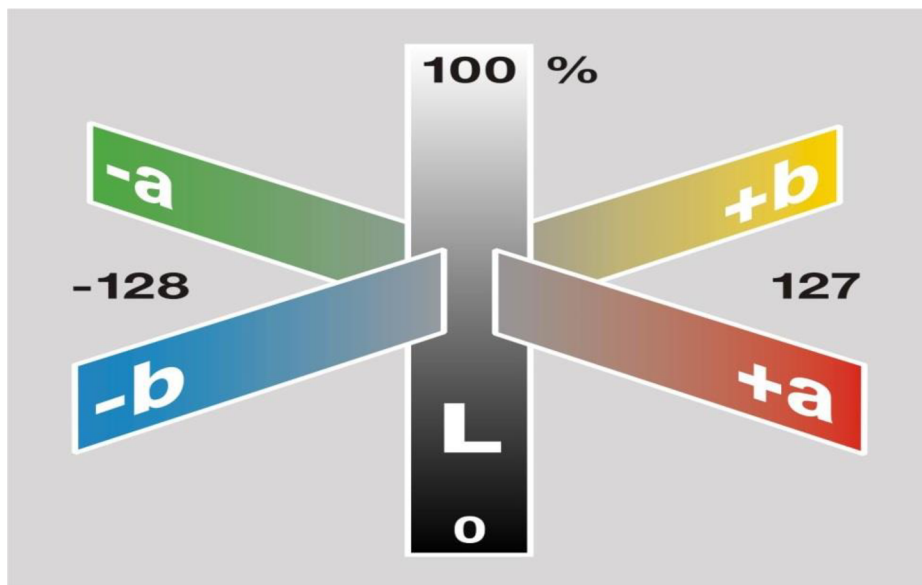
RGB model je základný model v počítačovej grafike. Za skratku RGB sa skrýva označenie troch základných farieb R - *Red* (červená), G - *Green* (zelená), B - *Blue* (modrá). Spôsob použitia jednotlivých zložiek RGB vychádza z aditívneho miešania farieb. V prípade modelu RGB nejde o miešanie fyzických farieb, ale o miešanie troch svetiel. Zlúčením všetkých troch farieb v plnej intenzite (255, 255, 255) vzniká farba biela, pre nulové hodnoty (0, 0, 0) vzniká farba čierna. RGB model ide skvele zobrazit' ako kocku, kde jednotlivé hodnoty x, y, z osy odpovedajú modrému, červenému a zelenému svetlu. Na uhlopriečke kocky je potom stav, kedy všetky tri svetla svietia na maximum svojej intenzity (viď obrázok 3). Tento farebný model však nie je úplne vhodný pri rozpoznávaní objektov z obrazu kvôli tomu, že je náročné určiť presnú farbu objektu. To je zapríčinené, tým že RGB model je dosť variantný čo sa týka intenzity svetla [6].



Obr. 3: RGB kocka. Zdroj: [6].

2.3.2 CIELAB

CIELAB farebný model je farebný model, ktorý bol vytvorený v roku 1976. Tento farebný model vychádza z ľudského vnímania farieb a opiera sa o hodnotu svetlosti a o hodnoty dvoch farebných zložiek, pomocou ktorých sú popísané všetky farby viditeľné pre ľudské oko. Farebné zložky sú vlastne zakódované koordináty v CIE XYZ. Farebné zložky sú akoby dve osi, pričom každá má na jednom konci jednu farbu a na druhom opačnom konci opačnú (komplementárnu) farbu [7].



Obr. 4: CIELAB kocka. Zdroj: [8].

2.3.3 HSL

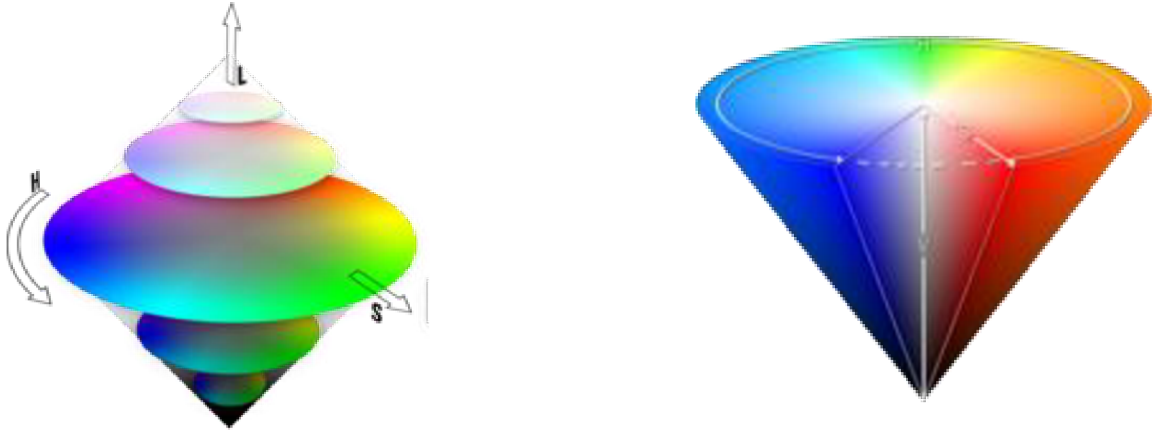
HSL - je farebný model, ktorý nie je tak intuitívny ako to je u RGB či CMYK farebného modelu, taktiež aj nevychádza z ľudského vnímania farieb ako to bolo u CIELAB. Jeho základne zložky sú:

- Hue (Hĺbka) : jedná sa o jedinečný odtieň
- Saturation (Sýtosť) : prebieha od plne nasýtenej farby po sivú s danou svetlosťou
- Lightness (Svetlosť) : prebieha od čiernej cez farbu zvolenú odtieňom a sýtosťou až k bielej

2.3.4 HSV

HSV - je farebný model, ktorý nie je tak intuitívny ako to je u RGB či CMYK farebného modelu, taktiež aj nevychádza z ľudského vnímania farieb ako to bolo u CIELAB. Jeho základne zložky sú:

- Hue (Hĺbka) : jedná sa o jedinečný odtieň
- Saturation (Sýtosť) : určuje vzdialenosť od osi modelu
- Value (Hodnota) : vzdialenosť od vrcholu modelu



Obr. 5: HLS a HSV model. Zdroj: [7].

2.4 Predspracovanie obrazu

Predspracovanie obrazu [9] (*Image preprocessing*) je prvým krokom pri spracovaní obrazu. Jeho cieľom je zlepšenie kvality obrazových dát, ktorá potlačí neželané deformácie a skreslenia alebo zlepši niektoré črty obrazu, ktoré sú dôležité z hľadiska ďalšieho spracovania. V podstate sa jedná o operácie s obrazmi na najnižšej úrovni abstrakcie. Existujú tri základné typy metód predspracovania obrazu:

- Jasové transformácie: transformácia závisí od vlastnosti samotného pixela
- Geometrické transformácie:
- Lokálne predspracovanie: transformácie závisia od samotného pixela a jeho okolia

2.4.1 Jasové transformácie

Poznáme dve skupiny jasových transformácií:

- **Jasové korekcie:** modifikujú jas pixela, kde berú do úvahy jeho hodnotu a polohu v obraze:
 - $f(i, j) = e(i, j) g(i, j)$, kde $e(i, j)$ je multiplikatívny koeficient chyby
 - ak nameriame obraz s konštantným jasom c , potom:

$$g(i, j) = \frac{c \cdot f(i, j)}{f_c(i, j)} \quad (1)$$

- **Šedotónové transformácie:** menia jasovú úroveň bez ohľadu na pozíciu v obraze

$$q = T(p) \quad (2)$$

Najčastejšie používané šedotónové transformácie sú:

- Roztiahnutie jasového intervalu : $q = (q_k - q_0)(p - a)/(b - a) + q_0$ (3)
- Ekvalizácia histogramu
- Pseudofarebné transformácie: jasovým úrovňam sú priradené farby, ľudské oko ich vníma senzitívnejšie ako jasové zmeny

2.4.2 Ekvalizácia histogramu

Histogram vyjadruje početnosť jednotlivých jasových úrovní v obrázku. Umožňuje vyčítať z obrázka, či je obrázok príliš svetlý alebo tmavý. To je najdôležitejšia vlastnosť prečo sa histogram v podstatnej miere používa pri spracovaní obrazu. Na histogram sa môžeme pozerat' ako na diskretnú funkciu hustoty pravdepodobnosti, ktorá je v nasledujúcom tvare :

$$\sum_{i=0}^k G(q_i) = \sum_{i=0}^k H(p_i). \quad (4)$$

Histogram môžeme vyrovnávať čo sa nazýva tzv. **ekvalizácia histogramu**. Cieľom ekvilizácie histogramu je vytvoriť obraz s rovnako rozloženými jasovými úrovňami v celej jeho jasovej škále. Ekvalizovaný histogram $G(q)$ zodpovedá uniformnej funkcii hustoty pravdepodobnosti f , ktorej funkčné hodnoty sú konštantné [9]:

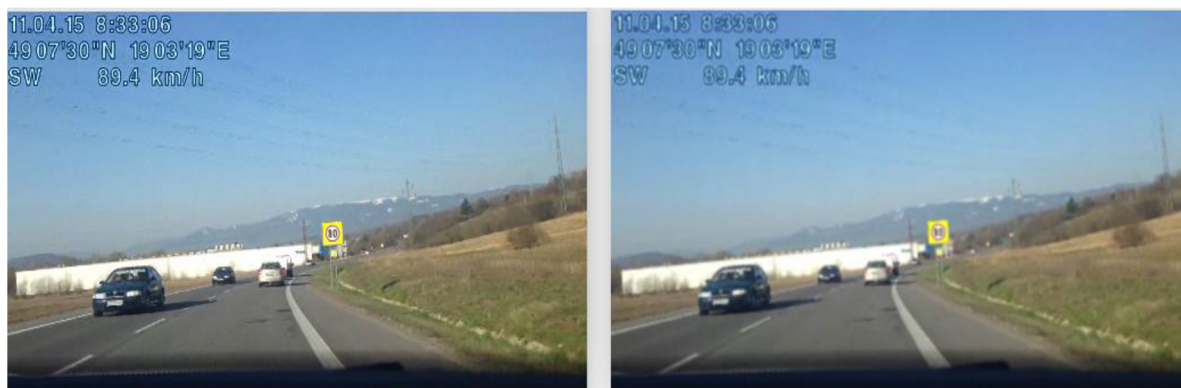
$$f = \frac{N^2}{q_k - q_0} \quad (5)$$

Gaussovo rozmazanie (Gaussian blur)

Gaussovo rozmazanie je jedna z metód, ktorá sa používa pri predspracovaní obrázka pre redukciu šumu v obraze. Je výsledkom rozmazania (*blur*) resp. vyhladenia (*smooth*) obrázku za použitia Gaussovej funkcie, ktorá je v nasledujúcom tvare [10] :

$$f(x) = a \exp\left(-\frac{(x - b)^2}{2c^2}\right) \quad (6)$$

Vyhladenie obrázku sa realizuje na základe masky, ktorá sa označuje ako konvolučné jadro (kernel). Maska sa posúva po celom obrázku, pričom sa pri každom posunutí vykoná súčin každého koeficientu v maske s hodnotou jasu v obraze „pod“ maskou a spočíta sa suma týchto súčinov. Výsledná suma je novou hodnotou pixela obrázka [11].



Obr. 6: Vľavo pôvodný obrázok, vpravo obrázok po použití Gausového rozmazania.

2.5 Segmentácia obrazu a metódy segmentácie

Segmentácia obrazu, sa v odbore počítačového videnia dá chápať ako proces rozdeľovania digitálneho obrazu na viaceré časti (skupina pixelov tzv. superpixels). Preto cieľom segmentácie obrazu je rozdeliť digitálny obraz na časti, ktoré reprezentujú nejaký objekt alebo predmet reálneho sveta. Segmentácia je vlastne prvým krokom pri spracovaní obrazu. Segmentácia môže byť:

- Úplná segmentácia: Rozdelenie obrazu na disjunktné oblasti, ktoré zodpovedajú objektom.
- Čiastočná segmentácia: Jej cieľom je rozdeliť obraz na časti, ktoré sú homogénne z hľadiska vybranej vlastnosti ako sú jas, farba, odrazivosť, textúra apod. [6].

2.5.1 Prahovanie (*Thresholding*)

Prahovanie je najjednoduchšia a najrýchlejšia segmentačná metóda, ktorá je založená na hodnote úrovne jasu každého pixelu. Princípom prahovania je nájdenie takej hodnoty prahu v obraze, pre ktorú platí, že všetky hodnoty jasu nižšie ako prah odpovedajú pozadiu, zatiaľ čo všetky hodnoty vyššie ako prah odpovedajú poprediu. Je to vlastne transformácia, ktorá zobrazuje vstupný obraz $f(i, j)$ na výstupný obraz $g(i, j)$ pri prahu T nasledovne[9]:

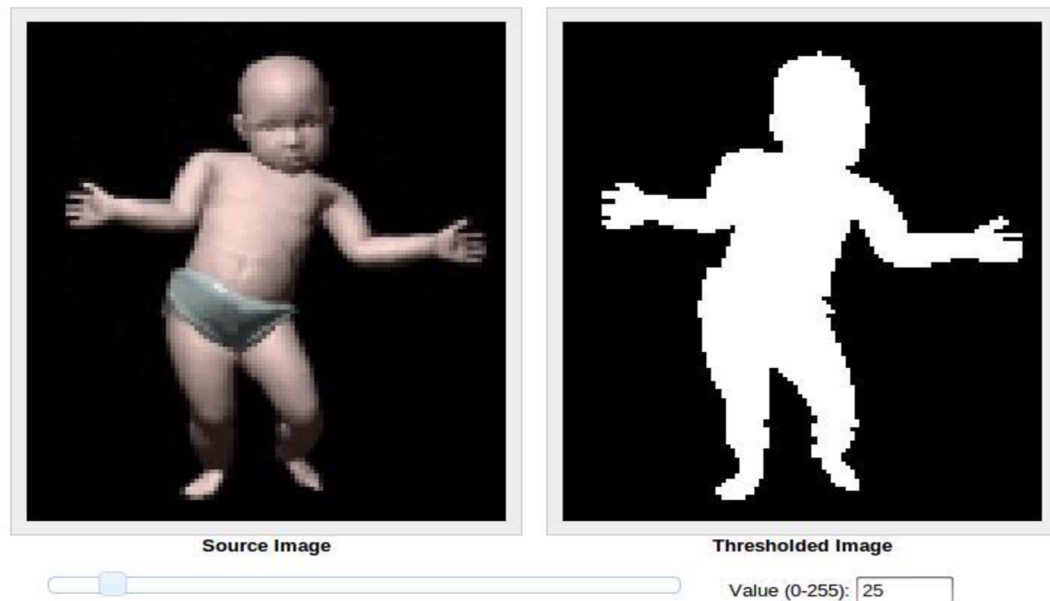
$$g(i, j) = 1 \quad \text{ak} \quad f(i, j) \geq T \quad (7)$$

$$g(i, j) = 0 \quad \text{ak} \quad f(i, j) < T \quad (8)$$

V praxi nie je možné použiť jeden prah na celej ploche obrazu. Je to v dôsledku zmeny jasu objektov a pozadia, zapríčinenými nerovnosťami osvetlenia v obraze. V tejto situácii môže pomôcť prahovanie s premenným prahom, kde hodnota prahu je určovaná podľa lokálnych vlastností obrazu. Existuje viacero modifikácií základného prahovania: Intervalové prahovanie, poloprahovanie,

prahovanie s viacerými prahmi. Pri prahovaní je nevýhodou vôbec zistiť a definovať správnu hodnotu prahu. Hodnota prahu sa môže zadať ručne alebo automaticky. Pre automatické určenie prahu sa používajú nasledujúce metódy[9]:

- Metóda p-podielu
- Metóda analýzy tvaru histogramu
- Optimálne prahovanie



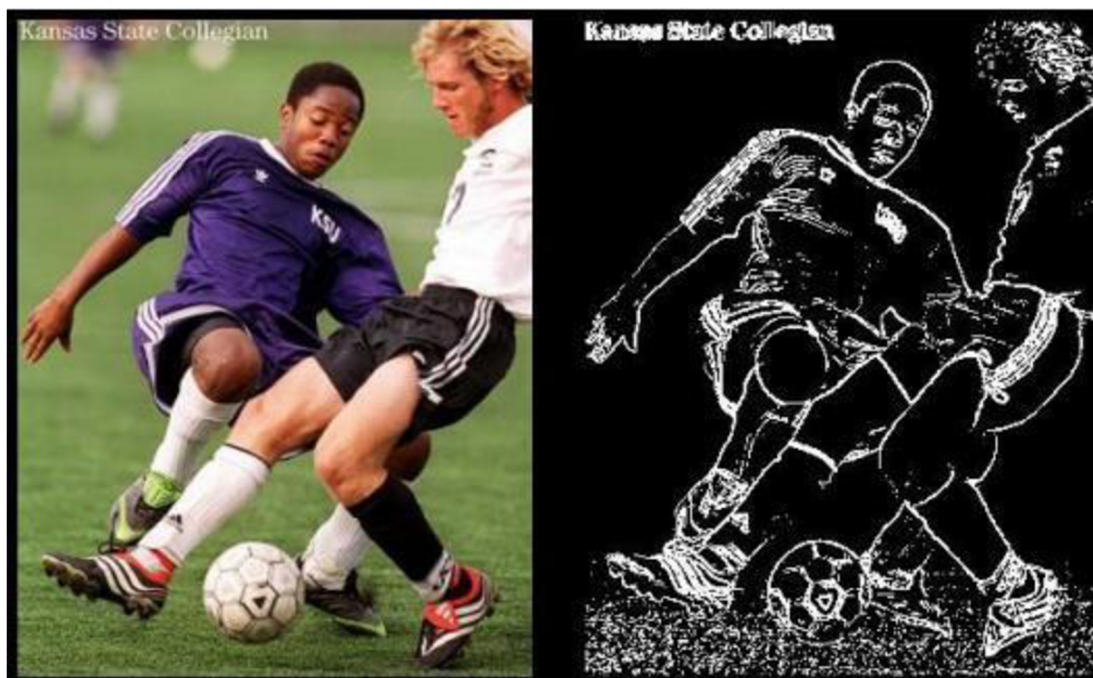
Obr. 7. : Segmentácia obrazu založená na prahovaní. Zdroj: [12].

2.5.2 Detekcia hrán (*Edge detection*)

Detekcia hrán je segmentačná metóda založená na princípe detekcie hrany v obraze (boundary-based). Vychádza zo skutočnosti, že hranice oblastí obrazu sú tvorené hranami, ktoré môžeme nájsť použitím niektorého z hranových operátorov:

- Cannyho hranový detektor
- Sobelov filter
- Robertsov operátor
- Operátor Prewittovej

Výstupom gradientného operátora je obraz, kde sú sice všetky hrany zvýraznené ale obraz môže obsahovať aj nechcené artefakty. K odstráneniu nechcených artefaktov v obraze sa obvykle používa prahovanie. Detektory hrán môžu vyvárať prerušované hranice aj falošne hranice pri segmentácii. Obraz hrán je preto potrebné ďalej spracovať. Hrany sa v obraze vyskytujú v miestach, kde dochádza k náhlej zmene jasů[13].



Obr. 8: Segmentácia obrazu založená na detekcii hrán. Zdroj: [13].

2.5.3 Regionálne metódy (*Region-based methods*)

Regionálne metódy sú segmentačné metódy, ktoré pre vyvárание segmentov(oblastí) využívajú pixely s podobnými vlastnosťami. Touto vlastnosťou môže byť napríklad jas, textúra a pod. Podstatou regionálnych metód je, že segment sa vytvára zdola nahor čiže od jedného pixela po celý segment. Najprv sú určitým algoritmom v obraze rozmiestnené inicializačné (*semienkové angl. seed*) pixely buď rovnomerne alebo náhodne. Celý segment potom vzniká iteratívnym rozrastaním okolo daného inicializačného pixela. Pri daných metódach nie je zaručené, že pri rôznom počte a rozmiestnení inicializačných pixelov bude výsledok segmentácie identický. Ich výhodou je, že dokážu segmentovať obraz, obsahujúci veľké množstvo šumu. Medzi regionálne metódy patria [13]:

Spájanie oblastí

Spájanie oblastí funguje na princípe vyvárания nových oblastí spájaním malých homogénnych oblastí. Na začiatku sa definuje počiatkové rozdelenie obrazu do veľkého množstva malých oblastí. Následne sa definujú kritéria pre spojovanie dvoch susedných oblastí. Susedné oblasti, ktoré vyhovujú vybranému kritériu sa budú spájať dokiaľ sa v obraze nebudú nachádzať dve oblasti, ktoré nie je možné spojiť [13].

Štiepenie oblastí

Štiepenie oblasti má opačný postup ako je to pri metóde spájania oblasti. Na začiatku sa vezme celý obraz a ten sa delí tak dlho, dokiaľ podoblasti neodpovedajú zadanému kritériu. Postupy štiepenia oblastí pracujú s rovnakými kritériami ako postupy spájania oblastí [13].

Štiepenie a spájanie oblastí

Štiepenie a spájanie oblasti využíva pyramidálnu reprezentáciu obrazu, kde je štiepenie a spojovanie realizované v rámci štvorcových oblastí pyramidálnej dátovej štruktúry. Pokiaľ je oblasť v danej úrovni pyramídy nehomogénna je rozštiepená na štyri podoblasti. Ak sú štyri oblasti navzájom homogénne a je možnosť ich spojiť tak sa spoja do vyššej úrovni pyramídy [13].

2.5.4 Watershed

Pri tejto metóde segmentácie sa vychádza z geografie. Obraz je chápaný ako terén alebo topografický reliéf, kde jas vstupného obrázku určuje výšku terénu –čierna je najnižšie a biela najvyššie. Princíp je založený na postupnom zaplavovaní terénu vodou alebo stúpaním hladiny. Výsledkom je obraz rozdelený do jednotlivých povodí (regiónov) oddelených hrádzami. Všetky body daného povodia sú označené takým istým unikátnym indexom. Pre obrazy obsahujúce šum vyvára táto metóda príliš veľa oblastí. To sa dá eliminovať vhodnou predprípravou obrázku alebo “zaplavovať” obrázok až od určitej prahovej výšky [13].

2.6 Morfológické operácie

Matematická morfológia zdôrazňuje úlohu tvaru v predspracovaní, segmentácii a popise obrazu. Vytvára množinu nástrojov, ktoré majú matematický základ a vedú k rýchlym algoritmom. Hlavným pojmom je bodová množina. Morfológia pracuje s transformáciami, ktoré sú opísané v relatívne jednoduchej nelineárnej algebre.

Rozdeľuje sa na binárnu matematickú morfológiu, ktorá pracuje na binárnych obrazoch s 2D bodovými množinami a na šedotónovú matematickú morfológiu, ktorá pracuje na šedotónových obrazoch s 3D bodovými množinami. Používa sa pri predspracovaní (odstránení šumu, zjednodušení tvaru), pri vytvorení kostry, stenčovaní a zhrubnutí obrazu, vytvorení konvexného obalu, pri segmentácii a pri získavaní kvantitatívnych popisov).

Základnými operáciami matematickej morfológie sú : **Erózia** a **Dilátácia**. Ich kombináciou vznikajú nové operácie ako sú morfológická operácia **Otvorenie** a morfológická operácia **Uzavretie**.

Morfológická operácia je daná ako relácia medzi dvomi bodovými množinami. Prvou množinou je bodová množina X . Druhou množinou je malá bodová množina B , nazvaná štruktúrálny prvok. Štruktúrálny prvok je tiež bodová množina, ktorá sa skladá z menšieho počtu bodov. Obsahuje jeden vyznačený bod O , ktorý sa nazýva reprezentatívny bod. [9]

2.6.1 Dilatácia

Dilatácia rozšahuje objekt o najbližšie pixely z jeho okolia, zväčšuje objekt, zaplňa malé diery a úzke zálivy. Vlastnosti dilatácie sú: komutatívnosť, asociatívnosť, invariantnosť voči posunutiu. Je založená na Minkowského súčte.

$$X \oplus B = \{p \in \mathcal{E}^2 : p = x + b, x \in X \text{ a } b \in B\}. \quad (9)$$

2.6.2 Erózia

Erózia zjednodušuje štruktúru objektu, stenčuje objekt, pričom príliš malé a tenké objekty sa stratia. Vlastnosti Erózie sú: nie je komutatívna, asociatívnosť, je prienikom všetkých posunutých množín, invariantnosť voči posunutiu. Je založená na Minkowského rozdieli a je duálna ku dilatácii (nie však inverzná).

$$X - B = \{p \in \mathcal{E}^2 : p + b \in X \text{ a } b \in B\}. \quad (10)$$

2.6.3 Otvorenie

Otvorenie je morfológická operácia, ktorú získame spojením erózie a dilatácie. Otvorenie množiny X štruktúrálnym prvkom B sa označuje $X \circ B$.

$$X \circ B = (X \ominus B) \oplus B \quad (11)$$

Hovoríme, že obraz X je otvorený vzhľadom na B , ak sa nezmení po otvorení množinou B . Otvorenie zjednodušuje objekt, lebo oddeľuje objekty spojené úzkou čiarou.

2.6.4 Uzavretie

Uzatvorenie je morfológická operácia, ktorú získame spojením dilatácie a erózie. Uzatvorenie množiny X štruktúrálnym prvkom B sa označuje $X \cdot B$.

$$X \cdot B = (X \oplus B) \ominus B \quad (12)$$

Hovoríme, že obraz X je uzatvorený vzhľadom na B , ak sa X nezmení po uzatvorení množinou B . Uzatvorenie zjednodušuje objekt, zaplňa malé diery a zálivy, spája objekty, ktoré sú dosť blízko seba

2.7 Extrakcia príznakov

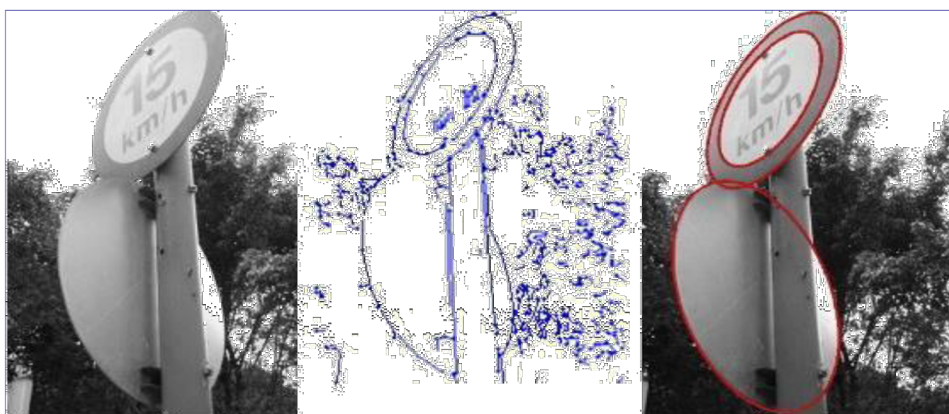
Príznak vo všeobecnosti predstavuje akúkoľvek informáciu, ktorú zvolený bod nesie. Môže sa jednať napríklad o farbu pixelu, priemernú farbu okolia pixelu, autokovarianciu a ďalšie hodnoty. Vo

väčšine prípadov je pre každý bod výsledkom výpočtu jediné číslo. Tieto čísla, spoločne s pozíciou v obraze sú uložené pre ďalšie spracovanie. Pri získavaní príznakov je snaha použiť ich čo najmenej z dôvodu zníženia dimenzie dát. Preto sa vyberajú také príznaky, ktoré sú efektívne a majú dobré rozlišovacie vlastnosti [14].

Extrakcia príznakov je základným krokom pri , na ktorý nadväzuje ďalšie spracovanie. Jednotlivé príznaky môžu byť získané buď z celého obrazu alebo len z nejakej jeho časti. Tieto časti majú tvar štvorcov či obdĺžnikov.

2.7.1 Houghova transformácia

Houghova transformácia je technika extrakcie príznakov ,ktorá slúži pre nájdenie parametrického popisu objektov v obraze. Pri hľadaní je potrebné vedieť analyticky popis hľadaného objektu. Preto je táto metóda používaná pre detekciu jednoduchých objektov v obraze ako sú priamky, kružnice, elipsy atď. Houghova transformácia sa predovšetkým používa pri segmentácii objektov, ktorých hranice sa dajú popísať jednoduchými krivkami. Cieľom tejto metódy je nájsť nedokonale inštancie objektov v rámci určitej triedy tvaru hlasovaním. Toto hlasovanie sa vykonáva v priestore parametrov, z ktorého sú kandidáti na objekt získaní ako lokálne maxima v tzv. akumuláčnom priestore, ktorý je explicitne vytvorený algoritmom pre výpočet Houghovej transformácie [15].



Obr. 9: Houghová transformácia. Zdroj:[16].

2.7.2 Histogram Orientovaných gradientov (HOG)

HOG je technika extrakcie príznakov, ktorá sa používa pri spracovaní obrazu za účelom detekcie objektu. Základnou myšlienkou je , že táto technika dokáže popísať vzhľad a tvar objektu v obraze distribúciou intenzity jednotlivých gradientov. Implementácia týchto príznakov je realizovaná rozdelením obrázku do malých koncentrovaných oblastí, ktoré sa nazývajú bunky (*cells*), kde pre každú takúto bunku sa vypočíta histogram smeru gradientov pre každý pixel v danej bunke.

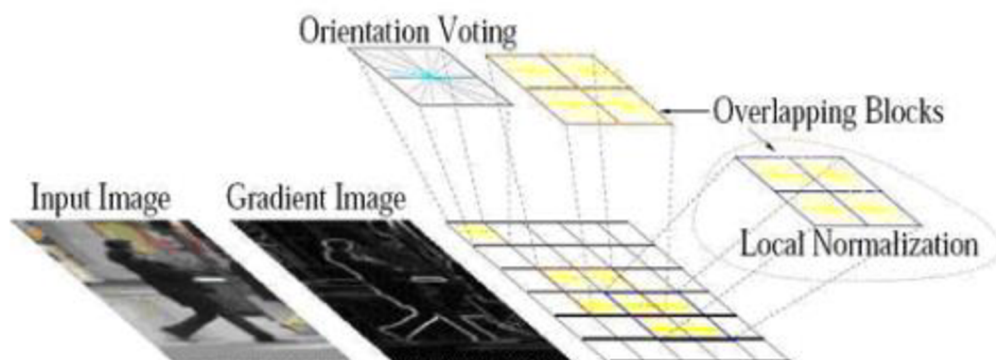
Pre efektívnejšie využitie tejto techniky je vhodné normalizovať lokálne histogramy vypočítaním veľkosti intenzity vo väčších častiach obrázku, ktoré sa nazývajú bloky a následne sa

táto vypočítaná hodnota použije pre normalizáciu všetkých buniek v bloku. Táto normalizácia vedie k menším náchylnostiam na zmeny v osvetlení a tieňov v obraze. Postup pri výpočte lokálnych histogramov je nasledujúci [17]:

1. **Vypočítanie hodnot gradientov:** Najčastejšou metódou pre vypočítanie hodnôt gradientov je použitie konvolučnej masky v horizontálnom a vertikálnom smere. Príklad takejto masky je nasledujúci:

$$D_x = [-1 \ 0 \ 1] \text{ and } D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad (13)$$

2. **Vytvorenie histogramu pre každú bunku:** V každej bunke sa spočíta histogram gradientov sčítaním veľkosti gradientov pre každú bunku.
3. **Normalizácia histogramu:** sa vykonáva za účelom dosiahnutia vyššej presnosti a nižšej náchylnosti na jasové zmeny v obraze. To sa dosahuje spájaním buniek do navzájom prepojených blokov. HOG príznak je potom daný ako vektor buniek v jednotlivých blokoch.



Obr. 10: Normalizácia histogramu. Zdroj:[17].

2.8 Detekcia a sledovanie objektov v obraze

Obsah každého obrazu, či už je to samostatný statický obrázok alebo snímka vo videosekvencií, možno rozdeliť na niekoľko hierarchicky usporiadaných úrovní abstrakcie. Prvú, najnižšiu úroveň abstrakcie tvoria pixely, základ každého obrazu, ktoré predstavujú informáciu o jase alebo farbe. Ďalšia úroveň abstrakcie sa zaoberá znakmi alebo vlastnosťami, ako sú:

- Hrany
- Rohy
- Čiary
- Krivky
- Farebné oblasti v rámci obrazu

Vyššia úroveň abstrakcie kombinuje a interpretuje tieto znaky ako objekty a k nim prislúchajúce atribúty. Na najvyššej úrovni sa preto využívajú koncepty spracovania a abstrakcie obrazu podobné ľudskému vnímaniu, ktoré spájajú jeden alebo viac objektov a definujú vzťahy medzi týmito objektmi. Detekcia objektu vo videosekvencii zahŕňa určenie výskytu daného objektu v postupnosti snímok, ako aj čo najpresnejšie určenie jeho polohy pre ďalší proces rozpoznávania. Princípom sledovania objektu je potom detekcia zmien jeho polohy, veľkosti a tvaru v priestore a čase [18].

Lucas - Kanade metóda (LK)

Lucas - Kanade metóda je jedna z mnohých metód, ktorá slúži na sledovanie objektov v obraze. Za jeho tvorcov sa považujú Bruce D. Lucas a Takeo Kanade. Táto metóda predpokladá, že optický tok je konštantný v lokálnom okolí uvažovaného pixelu. Rieši základne rovnice optického toku pre všetky pixely v tomto okolí pomocou kritéria najmenších štvorcov. Kombináciou informácií z niekoľkých blízkych pixelov, metóda Lucas – Kanade poskytuje riešenie nejednoznačnosti optického toku v danom obrázku.[19].

Oproti point-wise metodám [20] je Lucas-Kanade metóda menej citlivá na šum v obraze. Táto metóda najčastejšie využíva corner-like príznaky pre vypočítanie optického toku. LK rovnica pre vypočítanie optického toku je nasledujúca:

$$O = I_t(p_i) + \nabla I(p_i) \cdot [u \ v] \quad (14)$$

$I_t(p_i)$ - časový gradient

$\nabla I(p_i)$ - priestorový gradient

$[u \ v]$ - flow vektor

V praxi sa táto metóda využíva pre sledovanie objektu vo videosekvenciách, kde na základe porovnania predošlej a aktuálnej snímky táto metóda rozhodne o tom, či zdetegovaný objekt vo videu je ten istý objekt, ktorý bol zdetegovaný na predošlej snímke alebo sa už jedná o úplne nový objekt.

2.9 Rozpoznávanie (klasifikácia) obrazu

Proces triedenia pixelov do konečného počtu tried na základe ich údajových hodnôt. Je úzko spätá s pojmom rozpoznávanie vzoru (*Pattern Recognition*). Pri klasifikácii obrazu ide vlastne o hľadanie zmysluplných vzorov v údajoch, ktoré môžeme extrahovať klasifikáciou.

Napr. technika OCR (Optical Character Recognition). V praxi ide o porovnávanie detekovaných objektov s databázou vzorov.

Delenie klasifikácie [21]:

- Klasifikácia založená na pixeloch
- Klasifikácia založená na objektoch/oblastiach
- Podľa spôsobu priradovania príslušnosti pixelov/objektov do tried na:
 - Tvrďú klasifikáciu
 - Mäkkú klasifikáciu
- Podľa stupňa zasahovania užívateľa do klasifikačného procesu na:
 - Kontrolovanú klasifikáciu
 - Nekontrolovanú klasifikáciu

2.9.1 Klasifikačné metódy

Neurónové siete

Neurónové siete sú jeden z výpočetných modelov, ktoré sa používajú hlavne v oblasti umelej inteligencie. Medzi ďalšie oblasti ich využitia patrí **rozpoznávanie** ,kompresia obrazu a zvuku, predpovedanie vývoja burzových indexov, v lekárstve , k filtrovaniu spamu.

Boli vytvorené na základe biologických neurónových štruktúr. Základným stavebným kameňom neurónových sietí je umelý neurón, ktorý je vytvorený na základe biologického neurónu. Neuróny sú následne vzájomne prepojené a predávajú si signály, ktoré následne transformujú pomocou určitých prenosových funkcií. Neurón má ľubovoľný počet vstupov ale iba jeden výstup.

Cieľom umelých neurónových sietí je distribuované paralelné spracovanie dát. Na základe tohto faktu je ich hlavné využitie v oblasti počítačového videnia pri klasifikácii objektov v **reálnom čase**. Cieľom učenia neurónovej siete je nastaviť neurónovú sieť tak aby dávala presné výsledky. V biologických sieťach sú informácie uložené v dendridoch. V umelých neurónových sieťach sú informácie uložené v ich matematickom ekvivalente vo váhach. Učenie neurónovej siete rozlišujeme na učenie s učiteľom a učenie bez učiteľa. Fáza učenia neurónovej siete sa nazýva adaptívna. Po naučení sa táto fáza nazýva vybavovacia[22].

- **Učenie s učiteľom:** Podobné ako v biologických sieťach aj v umelých je využitá spätná väzba. Neurónovej sieti je predložený vzor. Na základe aktuálneho nastavenia je zistený aktuálny výsledok. Ten porovnáme s vyžadovaným výsledkom a určíme chybu. Následne spočítame potrebnú korekciu (záleží od typu neurónovej siete) a upravíme hodnoty váh alebo prahu aby sme znížili hodnotu chyby. Toto opakujeme až do dosiahnutia nami stanovenej minimálnej chyby. Potom je sieť adaptovaná.
- **Učenie bez učiteľa:** Pri učení bez učiteľa nevyhodnocujeme výstup. Pri tomto učení nám nie je známy. Sieť dostáva na vstup sadu vzorov, ktoré si sama triedi. Buď si vzory triedi do skupín a reaguje na typického zástupcu alebo si prispôsobí topológiu vlastnostiam vstupu.

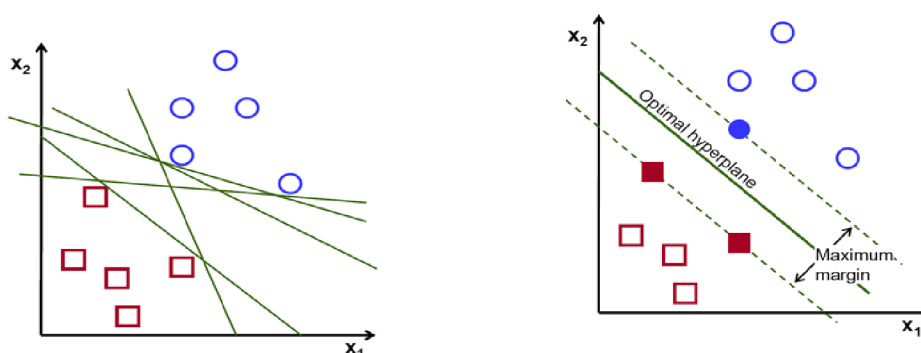
Template Matching

Klasifikačná metóda Template Matching je vysoko úrovňová metóda počítačového videnia, ktorá funguje na princípe porovnávania jednotlivých častí obrázku s preddefinovanou šablónou. Umožňuje porovnávať obrázky so šablónou bez ohľadu na ich otočenie a hodnotu jas. Daná technológia je veľmi flexibilná a relatívne jednoducho sa používa. To zapríčinilo, že daná metóda je jednou z najpopulárnejších metód používaných pri klasifikácii objektov. Jej nevýhodou je vysoká časová náročnosť pri porovnávaní veľa zložitých vzorov s preddefinovanou šablónou [23].

Support Vector Machine (SVM)

SVM je jedna metóda strojového učenia, ktorá funguje na princípe učenia učiteľom. Využíva sa pre rozpoznávanie vzorov, ktoré sa následne používajú pri klasifikácii a regresnej analýze. V úlohe klasifikácie SVM hľadá nadrovinu, ktorá v priestore príznačkov (*feature space*) optimálne rozdeľuje tréningové dáta. Tréningové dáta sa skladajú z pozitívnej tréningovej sady, ktorá obsahuje obrázky s objektami, ktoré chceme rozpoznať a negatívnej tréningovej sady, ktorá obsahuje obrázky kde nie sú objekty, ktoré chceme rozpoznať. Optimálna nadrovina je taká, ktorej body ležia v opačných polopriestoroch a hodnota minima vzdialenosti bodov od roviny je čo najväčšia. Inými slovami okolo nadroviny je na oboch stranách, čo najširší pruh bez bodov, maximálny margin (*maximal margin*). Na popis nadroviny stačia iba najbližšie body, ktorých je obvykle málo. Tieto body sa nazývajú podporné vektory (*support vectors*). Rozdeľujúca nadrovina je lineárna funkcia v priestore príznačkov.

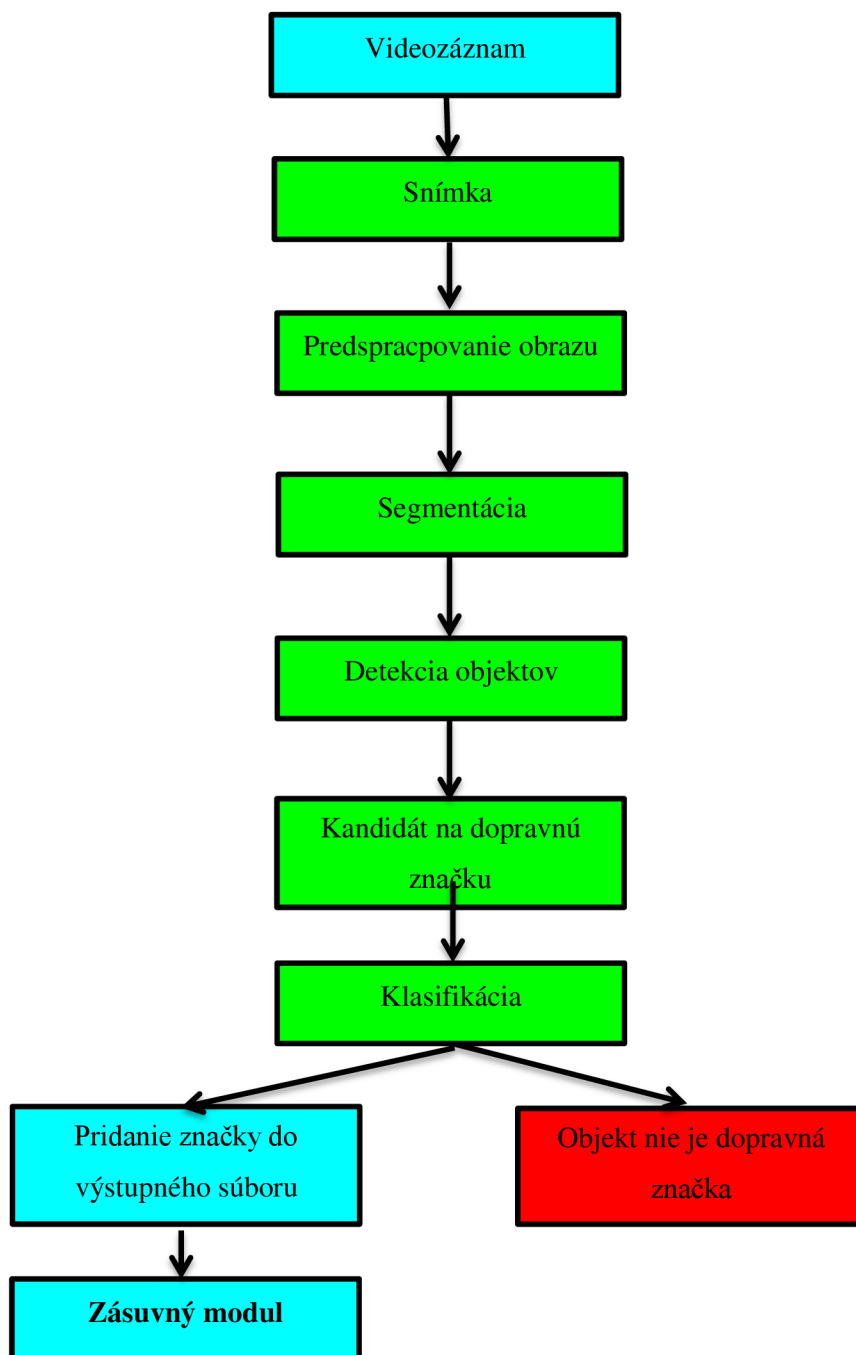
Dôležitou súčasťou SVM je jadrova transformácia (*kernel transformatio*). Táto jadrova transformácia umožňuje previesť pôvodne lineárne neseparovateľné dáta na dáta lineárne separovateľné. Používajú sa rôzne jadrové transformácie. Výhodou jadrovej transformácie je, že transformácia sa dá definovať pre rôzne typy objektov nie iba pre body v R^n . Napr. pre grafy, stromy, atď. [24]



Obr. 11: Vľavo deliace roviny, Vpravo optimálna deliaca rovina. Zdroj [25]

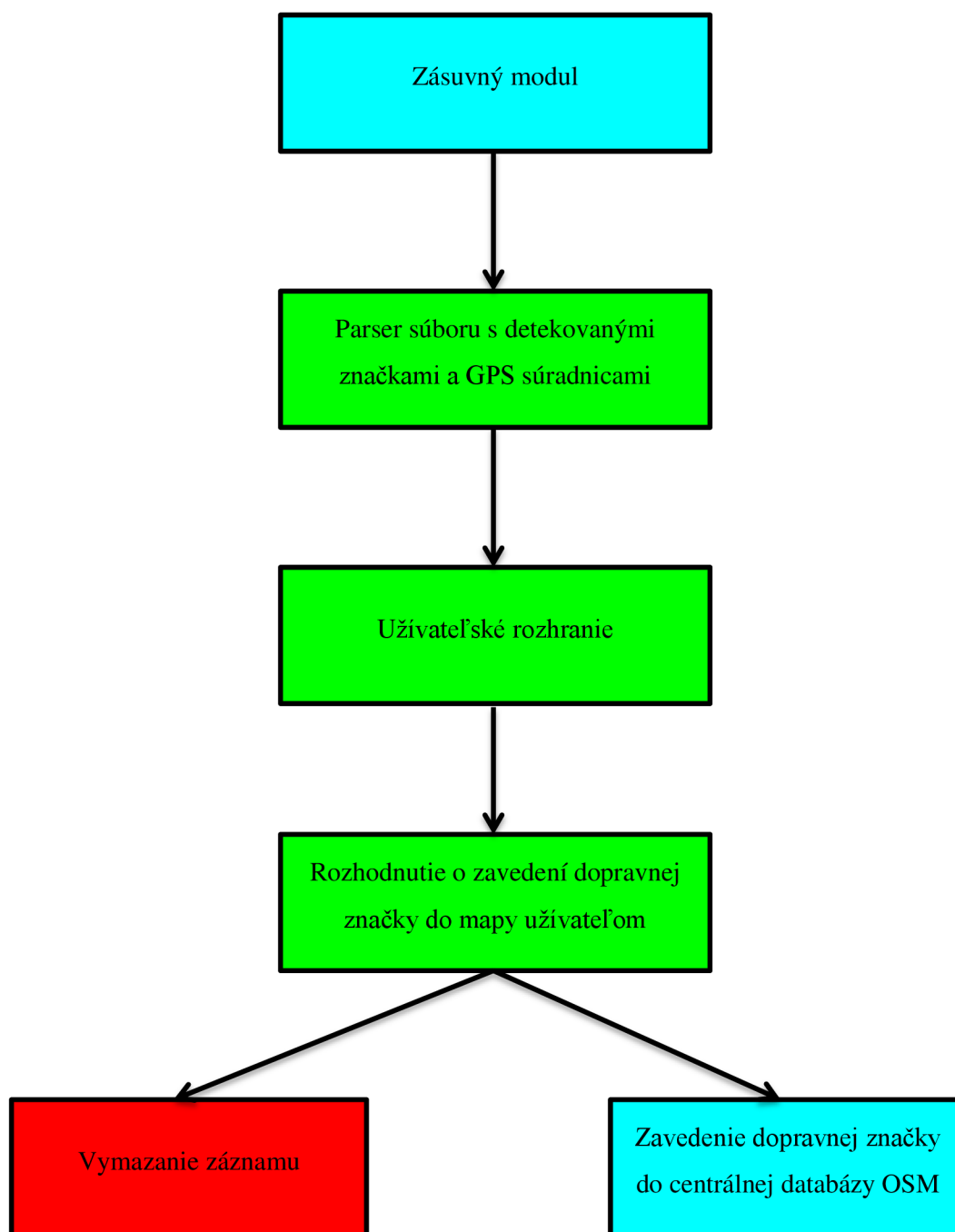
3 Návrh riešenia

3.1 Schéma aplikácie slúžiaca pre spracovanie obrazu



Obr. 12: Schéma, ktorá popisuje spracovanie a klasifikáciu vstupného obrazu.

3.2 Schéma zásuvného modulu



Obr. 13: Schéma popisujúca jednotlivé časti zásuvného modulu pre JOSM.

3.3 Získanie vstupných dát

3.3.1 Vstupný videozáznam

Vstupom aplikácie bude videozáznam obsahujúci dopravné značky a GPS záznam, ktorý bol zhotovený videokamerou počas jazdy dopravným prostriedkom. Video záznam bol zhotovený videokamerou v mobilnom telefóne iPhone 5. Technická špecifikácia videokamery telefónu je nasledujúca: 8-megapixelov iSight kamera, HD (1080p) až 30 snímok za sekundu, video stabilizátor [26]. Pre získavanie GPS záznamu bude použitá aplikácia Record GPS [27], ktorá umožňuje zisťovať GPS polohu v určitých časových intervaloch a ten následne exportovať vo formáte XML. V tomto prípade je nutné zosynchronizovať časovú stopu videozáznamu s GPS záznamom. Ďalšou možnosťou je využitie aplikácie Car Camera DVR , ktorá umožňuje priamo vo vstupnom videozázname zobrazovať aktuálnu GPS polohu. Práca s takýmto typom záznamu je užívateľsky jednoduchšia, užívateľ nemusí získavať GPS záznam inými prostriedkami, má ho priamo vo videu. Nevýhodou tohto typu záznamu je väčšia časová náročnosť na spracovanie.

3.3.2 Získanie snímky z videozáznamu

Z každého videozáznamu je nutné získať snímku, ktorá bude následne spracovaná aplikáciou určenej pre spracovanie obrazu. Získavanie snímok z videa bude realizované vo viacerých FPS, z dôvodu následného testovania časovej náročnosti a úspešnej detekcie dopravných značiek.



Obr. 14: Snímka s GPS záznamom aplikáciou Car Camera DVR[28].

3.4 Návrh aplikácie pre spracovanie obrazu

3.4.1 Základny popis aplikácie

Cieľom návrhu aplikácie je navrhnuť základný koncept aplikácie, ktorá bude schopná z videa rozpoznať jednotlivé dopravné značky a ukladať ich do výstupného súboru. Tento súbor bude slúžiť pre ďalšie spracovanie. V návrhu sú opísané základné stavebné bloky, ktoré budú detailne rozpísané až v časti implementácie. Implementačným programovacím jazykom tejto aplikácie je jazyk C++ za použitia OpenCV knižnice.

3.4.2 Predspracovanie obrázka (*preprocessing*)

Po získaní snímku z videozáznamu nasleduje jeho úprava. Táto úprava obrázka bude mať za následok zvýšenie kvality rozpoznávania jednotlivých dopravných značiek v obraze. Rozhodol som sa preto použiť Gausove rozostrenie. Gausove rozostrenie mi umožní redukovat' určitý šum v obraze a tým pádom zlepšiť kvalitu obrazu pri rôznych svetelných podmienkach.

3.4.3 Segmentácia, detekcia objektov

Ako segmentačnú metódu som si zvolil prahovanie na základe farby. Daná metóda sa bude využívať na rozpoznávanie 3 typoch dopravných značiek podľa farby. Rozpoznávať sa budú červené dopravné značky, niektoré modré dopravné značky, žlté dopravné značky (značka hlavnej cesty a koniec hlavnej cesty). Pre použitie segmentácie na základe farieb je vhodné obrázok previesť do iného farebného modelu. Konkrétny farebný model bude špecifikovaný až v časti implementácie, kde sa experimentálne zistí, ktorý farebný model bude poskytovať najlepšie výsledky pri mnou vybranej segmentačnej metóde.

Následne sa z vybraného farebného obrázku experimentálne určia vhodné hodnoty prahu pre červenú, modrú a žltú farbu. Takto sa získajú 3 čiernobiely obrázky na ktorých bude pozadie reprezentované čiernou farbou a popredie bielou farbou. Následne sa vhodnou morfológickou operáciou zvýrazia body v obraze. Na takto upravený čiernobiely obraz sa použije metóda na zistenie kontúr v obraze. Táto metóda je vhodná pri zdetegovaní objektov na základe rôznych geometrických tvarov. Keďže dopravné značky sú najčastejšie reprezentované nasledujúcimi geometrickými tvarmi:

- trojuholníkový tvar
- obdĺžnikovitý tvar
- šesťuholníkový tvar
- kosoštvorcový tvar
- kruhový tvar

Rozhodol som sa preto tieto geometrické tvary z obrazu detekovať. Výsledkom tejto časti spracovania obrazu bude zdetegovaný objekt, ktorý sa zaradí do jednotlivých kategórií na základe jeho geometrického tvaru. V nasledujúcej fáze, pri klasifikácii sa zistí o akú konkrétnu dopravnú značku sa jedna a popri prípade sa odstráni falošne detekcie.

3.4.4 Klasifikácia využitím HOG príznačkov a SVM klasifikátora

Predtým ako sa začne samotný proces klasifikácie je nutne natrénovať klasifikátor vhodnými tréningovými dátami. Po natrénovaní klasifikátora sa prejde k samotnému rozpoznávaniu objektov v obraze. V prvej fáze sa extrahujú príznaky z každého nájdeného objektu a tie sa pošlú do klasifikátora, ktorý rozhodne či objekt je vhodným kandidátom na dopravnú značku alebo nie. V prípade, že objekt takýmto vhodným kandidátom je, prejde sa k druhej fáze klasifikácie a to k zisteniu o akú dopravnú značku sa vlastne jedná. Keďže už pri detekcii objektov z obrazu sa kategorizovali objekty do jednotlivých kategórií na základe ich geometrických tvarov, tak práve táto ich kategorizácia je vhodná pri vytvorení piatich hlavných klasifikátorov. Každý takýto hlavný klasifikátor, ktorý funguje na úrovni mnou zvolených geometrických tvarov obsahuje menšie klasifikátory pre jednotlivé dopravné značky. Napríklad taký trojuholníkový objekt sa bude klasifikovať v rámci klasifikátora pre trojuholníkové značky a už nie v rámci klasifikátora pre kruhové značky..

3.4.5 Vytvorenie výstupného súboru s GPS súradnicami

Po úspešnom rozpoznaní dopravnej značky je nutné značku uložiť a prepojiť s editorom JOSM, ktorý slúži pre zavedenie dopravných značiek do OSM máp. To sa realizuje pomocou pluginu, ktorého návrh je predstavený v kapitole 3.4.6. Samotné prepojenie rozpoznanej značky s pluginom bude realizované formou xml súboru, ktorý bude výstupným bodom aplikácie pre rozpoznávanie dopravných značiek. Tento súbor bude obsahovať nasledujúce údaje:

- *Description:* Typ dopravnej značky
- *Img:* Absolútna cesta k rozpoznávanému obrázku
- *Time:* Čas zdetekovanej značky vo videu

Ako si môžeme všimnúť medzi údajmi výstupného súboru chýba údaj o GPS pozícii nájdenej značky. Tento údaj sa na základe časovej stopy pre dopravnú značku dopočíta až v Zásuvnom module a to hlavne z nasledujúcich dôvodov:

- Zbytočne, pridaná ďalšia časová režia na spracovanie obrazu.
- V jazyku C++ nie sú vstavané knižnice určené pre prácu s xml a gpx súbormi, je nutné si takéto knižnice vyhľadať, stiahnuť a importovať

3.4.6 Návrh zásuvného modulu pre JOSM

Základný popis zásuvného modulu

Cieľom návrhu zásuvného modulu (*pluginu*) pre editor JOSM je navrhnuť základný koncept zásuvného modulu, ktorý bude slúžiť na pridávanie nájdených dopravných značiek z videa do editoru JOSM. V návrhu sú predstavené jednotlivé funkcie zásuvného modulu kde užívateľ následne rozhodne či tieto údaje bude chcieť pridať do OSM mapy alebo nie. Užívateľovi bude umožnené v plugine rozhodnúť o tom či danú dopravnú značku pridať alebo nepridať do mapy, takisto mu bude umožnené zmeniť typ dopravnej značky. Implementačným programovacím jazykom tohto pluginu bude jazyk Java, bez nutnosti použitia ďalších štandardných knižníc.

Vstup zásuvného modulu

Vstupnými parametrami do zásuvného modulu budú súbory, ktoré budú v nasledujúcom formáte:

- XML súbor obsahujúci rozpoznané dopravné značky
- GPX súbor obsahujúci GPS súradnice zachytené v určitých časových intervaloch

Po načítaní týchto súborov sa následne vyparsujú z každého súboru dôležité údaje, ktoré sú nutné pre zavedenie dopravnej značky do mapy. Prakticky, ide o priradenie GPS súradníc jednotlivým dopravným značkám na základe ich časovej stopy. Keďže GPS súradnice sú málokedy získane v každej sekunde (väčšinou sa zisťujú v nejakom časovom intervale napr. 10sekúnd) je nutné ich dopočítať napr. pre každú sekundu. To sa bude realizovať na základe korelácie medzi predchádzajúcim záznamom a nasledujúcim záznamom. Po zistení všetkých dôležitých údajov sa prejde k realizácii užívateľského rozhrania, kde sa tieto údaje zobrazia.

Užívateľské rozhranie

Užívateľské rozhranie zásuvného modulu pre JOSM bude vytvorené s ohľadom intuitívnosť a minimalizmus pre užívateľa. Užívateľské rozhranie bude zobrazovať údaje získane v predchádzajúcom kroku nasledovným spôsobom. V strede okna, ktoré reprezentuje zásuvný modul bude zobrazený obrázok nájdenej dopravnej značky, v nezmenenom tvare. Ďalej sa v ľavom hornom rohu zobrazia GPS súradnice značky v podobe *Latitude*, *Longitude* a priradená odpovedajúca GPS hodnota. Typ dopravnej značky s možnosťou zmeny typu tejto dopravnej značky bude tiež ponúknutý užívateľovi. Užívateľ formou potvrdzujúcich tlačidiel rozhodne o vložení dopravnej značky do editoru JOSM, poprípade rozhodne a zahodí takejto značky.

Zavedenie dopravnej značky do mapy

Zavedenie značiek do mapy sa bude realizovať nasledujúcimi spôsobmi. V prvom kroku plugin vytvorí novú vrstvu (*layer*) pre JOSM. V nasledujúcom kroku po potvrdení značky užívateľom vytvorí nový bod s GPS súradnicami, ktorý pridá do vrstvy. Pre daný bod je typ značky špecifikovaný *tagom* s atribútmi **klúč a hodnota** (*key and value*). Tomuto bodu sa priradia GPS súradnice a typ značky. Typ značky je špecifikovaný *tagom* a atribútmi **klúč a hodnota**.



Obr. 15 Obrázok so zavedenými značkami do mapy

4 Implementácia

Pri vývoji aplikácie pre rozpoznávanie dopravných značiek a zásuvného modulu boli použité techniky predstavené v kapitole Teoretická časť. Aplikácia pre rozpoznávanie dopravných značiek bola vyvíjaná v jazyku C++ v prostredí Eclipse 3.8. Zásuvný modul bol vyvíjaný v jazyku Java taktiež v prostredí Eclipse 3.8. Tak ako aplikácia aj zásuvný modul boli vyvíjané a testované pod operačným systémom Linux Ubuntu 14.04 32-bit.

Použitá knižnica:

- **OpenCV** – Táto knižnica bola použitá pri vývoji aplikácie určenej na rozpoznávanie dopravných značiek. Táto knižnica ponúka veľké množstvo algoritmov a optimalizácií, ktoré sa vo veľkej miere používajú v počítačovom videní. V prvotnej fáze bola aplikácia vyvíjaná pod verziou knižnice 3.0 Beta, avšak nakoľko sa jednalo o Beta verziu, tak niektoré algoritmy nemali plnú podporu oproti minulým verziám. Preto som bol nútený prejsť k verzii 2.4, v ktorej je výsledná aplikácia zimplementovaná.

4.1 Aplikácia pre rozpoznávanie dopravných značiek z videa

Aplikácia pre rozpoznanie dopravných značiek je klasická konzolová aplikácia, s grafickým vstupom vo forme videa. Aplikácia je spúšťaná s jedným parametrom, ktorý reprezentuje relatívnu alebo absolútnu cestu k vstupnému videozáznamu. Tento parameter špecifikuje užívateľ. Výstupom aplikácie je XML súbor s rozpoznanými dopravnými značkami a absolútnou cestou k miestu uloženia rozpoznannej značky.

4.1.1 Načítanie videa a získanie obrázku

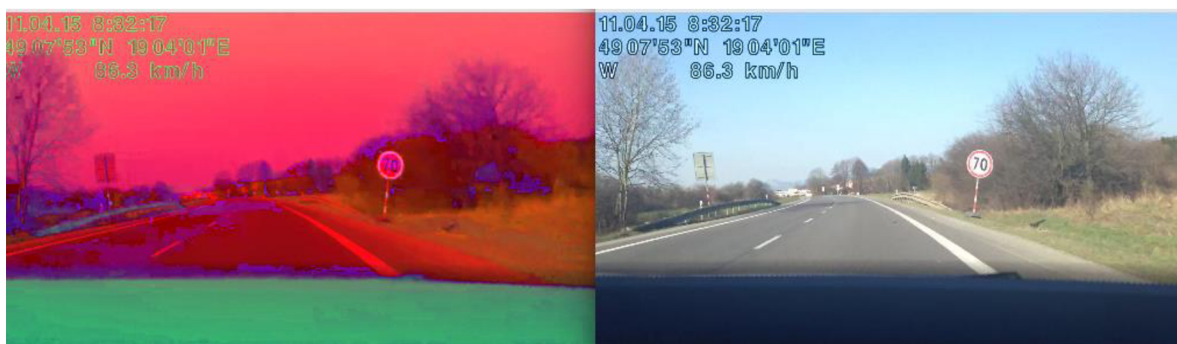
Prvým krokom aplikácie pre rozpoznávanie dopravných značiek je načítať video po jednotlivých snímkach. Každá takáto snímka sa následne bude analyzovať. Veľkosť načítanej snímky z videa som si stanovil na 512x288 px. Táto veľkosť je dostačujúca pri následnej práci s obrazom. V tejto fáze aplikácie som sa stretol s veľmi závažným problémom, ktorý sa týkal príliš vysokej časovej náročnosti na načítanie snímky z videa. To je zapríčinené hlavne tým, že mnou získané videozáznamy sú vo full HD rozlíšení, ktoré je 1920x1080 px. Načítanie takejto jednej snímky trvalo 2,6 krát dlhšie oproti normálu. Tento problém je možné čiastočne eliminovať znížením kvality vstupného videa avšak za cenu zníženia kvality snímky, a tým pádom znížením úspešnosti rozpoznávania dopravných značiek.

4.1.2 Predspracovanie obrázka

Predspracovanie obrázka je vhodné, pre zlepšenie kvality snímky a odstránenia neželaných artefaktov v obraze. V aplikácii som si zvolil Gausovo rozostrenie, vid'. obr. 6. Gausovo rozostrenie je v OpenCV realizované pomocou funkcie `GaussianBlur`. Avšak výraznú zmenu na kvalitu detekcie a rozpoznávania som nezaznamenal.

4.1.3 Zmena farebného modelu

Získaná snímka zo vstupného videozáznamu je vo formáte RGB. Avšak takýto farebný model nie je úplne vhodný pre ďalšie spracovanie, hlavne z dôvodu, že RGB model je dosť citlivý na zmeny jasu v obraze. Preto som sa rozhodol experimentovať s rôznymi farebnými modelmi. Zvolil som si HSV farebný model, ktorý v segmentácii poskytuje uspokojivé výsledky. Prevedenie RGB farebného obrázka do modelu HSV sa v OpenCV realizuje pomocou funkcie `cvtColor`, ktorej parametrom je farebný model do ktorého chceme obrázok previesť. Experimentoval som takisto aj s CIELAB farebným modelom, bohužiaľ sa mi pre neho nepodarilo nastaviť vhodné prahovacie hodnoty, ktoré by mi poskytovali uspokojujúce výsledky.



Obr. 16 : Vľavo obrázok po prevedení do HSV, vpravo originálny obrázok v RGB.

4.1.4 Segmentácia

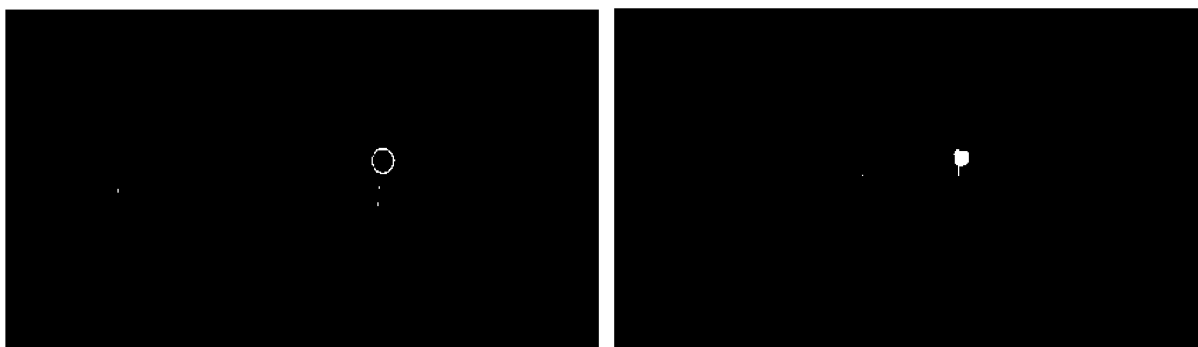
Ako segmentačnú metódu som si zvolil prahovanie na základe červenej, tmavomodrej, žltej farbe. Pri prahovaní dokážem oddeliť popredie od pozadia obrázka, kde popredie bude prezentované bielou farbou a pozadie čiernou farbou. Prahovacie hodnoty pre jednotlivé farby som experimentálne stanovil na základe mnou získaných videozáznamov. Prahovacie hodnoty pre jednotlivé farby sú zobrazené v nasledujúcej tabuľke. Príklad takejto segmentácie je v tabuľke. č.1.

	HUE	SATURATION	VALUE
RED	(140 - 190)	(68 - 255)	(71 - 255)
BLUE	(86 - 140)	(100 - 255)	(132 - 250)
YELLOW	(35 - 80)	(70 - 242)	(60 - 255)

Tabuľka č.1: Hodnoty prahov, pre jednotlivé farby.

4.1.5 Morfológické operácie

Po fáze segmentácie obrazu sa prechádza k nasledujúcemu kroku, t.j. k použitiu vhodnej morfológickej operácie. Po experimentovaní s rôznymi morfológickými operáciami som sa rozhodol použiť morfológickú operáciu uzavretie, čiže postupnosť erózia a dilatácia. Táto operácia dokáže pospájať pixely, ktoré sú blízko seba, do súvislejších celkov a tým zlepšiť kontúry objektov v obraze. Príklad takejto operácie je na obr. č. číslo 18. V OpenCV sa uzavretie realizuje pomocou funkcie `morphologyEx`.



Obr. 17: Prahovanie podľa červenej farby.

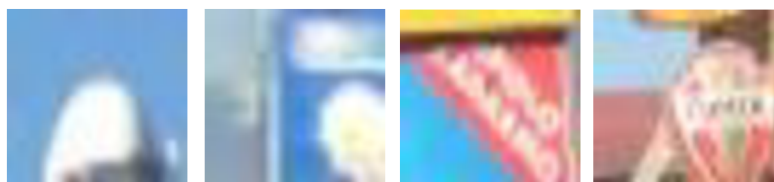
Obr. 18: Morfológická operácia uzavretie.

4.1.6 Extrakcia kontúr z obrázka

Po použití morfológickej operácie uzavretie sa prejde k hľadaniu objektov v obraze. To sa realizuje spôsobom, ktorý sa nazýva metóda extrakcie príznakov. Metóda extrakcie príznakov dokáže popísať objekty v obraze na základe ich geometrických vlastností. Táto metóda tiež dokáže eliminovať príliš malé objekty v obraze, ktoré by predstavovali veľké množstvo falošných kandidátov na dopravnú značku. V OpenCV je táto metóda predstavená funkciou `findContours`. Výsledkom tejto časti detekcie obrazu je tzv. Region of Interest (ROI), to je obrázok s objektom, ktorý sa našiel v obraze. viď obrázky č. 19 a č. 20. Tento obrázok sa následne upraví na normalizovanú veľkosť 32x32 px, a následne pripustí k samotnej klasifikácii. Keďže táto fáza produkuje veľké množstvo objektov (tie na ktorých sa dopravná značka nachádza a tie na ktorých nie je dopravná značka) tak až v samotnom procese klasifikácie sa tieto falošne zdetekované objekty odstránia.



Obr. 19: ROI, správne detekcie.



Obr. 20: ROI, falošné detekcie.

4.1.7 Klasifikácia

Za klasifikačnú metódu som si zvolil Support Vectore Machine, SVM, ktorá je detailne popísaná v kapitole o klasifikácii. Prvým a najdôležitejším krokom pre úspešnú klasifikáciu je natrénovanie klasifikátora. Trénovanie klasifikátora prebieha vo viacerých krokoch. V prvom kroku je nutné si určiť pozitívnu a negatívnu trénovaciu sadu. Za pozitívnu trénovaciu sadu som si zvolil jednotlivé typy dopravných značiek rozpoznávaných v predchádzajúcom kroku. Tieto obrázky sú pod rôznym natočením a na rôznych pozadiach. Za negatívnu trénovaciu sadu som si zvolil obrázky s objektmi na ktorých sa dopravné značky nenachádzajú.

V nasledujúcom kroku tréovania, sa jednotlivé obrázky trénovacej sady prevedú do šedotónového obrázku (*GrayScale*). Tento krok sa vykonáva hlavne pre to, že šedotónový obrázok je menej závislý od zmenách jasu v obraze. Následne sa z týchto obrázkov vypočítajú príznaky HOG. V OpenCV je to tento výpočet HOG príznakov realizovaný metódou `compute`, ktorá je zavolaná na objekt typu `HOGDescriptor`. HOG príznakmi sa následne natrénuje klasifikátor. V aplikácii mám natrenovaných toľko klasifikátorov, koľko dopravných značiek rozpoznávam a jeden hlavný klasifikátor, ktorý slúži na odstránenie falošných detekcií. Pri tréovaní klasifikátora je dôležité si zvoliť vhodný typ kernelu. Typ kernelu som si experimentárne stanovil kernel typu RBF. Trénovanie SVM klasifikátora sa v OpenCV realizuje metódou `train`, ktorá sa použije na objekt typu `CvSVM`. Po natréovaní klasifikátora sa prejde k samotnej klasifikácii, čo sa realizuje nasledujúcim spôsobom:

- Každý obrázok obsahujúci zdetekovaný objekt v obraze, ROI (Region of Interest) sa prevedie do šedotónového obrazu.
- Z každého šedotónového obrazu sa získajú HOG príznaky, ktoré sa následne použijú ako vstup do klasifikátora
- Rozhodnutie o tom o akú dopravnú značku sa jedná sa realizuje v OpenCV pomocou metódy `predict`

4.1.8 Sledovanie objektu v obraze

Algoritmy sledovania optického toku v obraze som sa rozhodol použiť hlavne, z tohto dôvodu aby sa mi tá istá dopravná značka nezdetegovala na každej snímke za sebou ale iba práve raz. Využil

som na to Lucas – Kanadeho algoritmus (LK), ktorý je v OpenCV reprezentovaný funkciou `calcOpticalFlowPyrLK`.

Tento algoritmus funguje na princípe sledovania bodov z aktuálnej snímky a snímky slúžiacej ako sledovací bod. Tieto body nemôžu byť úplne náhodne. Je nutné aby tieto body, ktoré sa budú sledovať boli vo vnútri dopravnej značky. Tie som extrahoval pomocou algoritmu, ktorý je založený na hľadani rohov (corners) a použil ich v LK algoritme.

V OpenCV je metóda hľadania rohov reprezentovaná funkciou `GoodFeaturesToTrack`. V prípade, že algoritmus rozpozná v aktuálnej snímke, inú dopravnú značku ako tú, ktorá bola rozpoznaná na snímke slúžiacej ako sledovací bod, aktualizuje ju aktuálnou snímkou a zapíše do súboru. V opačnom prípade sa nič nedeje, pretože sa stále jedná snímku s tou istou dopravnou značkou. Tento algoritmus ma ale jeden zádrhel, ktorý tkvie v tom, že ak sa vo videu na rovnaj ceste rozpoznajú za sebou dve rovnaké dopravné značky, algoritmus ich vyhodnotil, že sa jedná o tú istú dopravnú značku. V skutočnosti sú to však dve rôzne dopravné značky toho istého typu. Tento problém sa mi podarilo celkom úspešne vyriešiť nasledujúcim spôsobom. Ak bezprostredne medzi dvoma rozpoznávanými dopravnými značkami vo videu je časový rozdiel minimálne 3 sekundy tak vieme s určitosťou povedať, že sa už nejedná o tú istú snímku. A aktualizujeme snímku, ktorá slúži ako sledovací bod.



Obr. 21: Rozpoznaná dopravná značka vo videu ohraničená modrým štvorcem.

4.2 Zásuvný modul (Plugin) pre editor JOSM

Prepojenie Zásuvného modulu s editorom JOSM

V prvom kroku pri vývoji pluginu bolo nutné si naštudovať JOSM API, ktoré poskytuje rozhranie pre komunikáciu Pluginu s JOSM. Samotné prepojenie je realizované v triede `SVK_Main`, ktorá je invokovaná pri spustení pluginu samotným editorom JOSM. V tejto triede sa taktiež volá konštruktor, ktorý vytvorí nový objekt triedy `Map`. V triede `Map` sú realizované všetky potrebné časti zásuvného modulu, ktoré sú formou užívateľského rozhrania ponúknuté užívateľovi.

4.2.1 Parsovanie xml súboru s rozpoznanými dopravnými značkami

Parsovanie údajov o type, mieste uloženia a čase nájdenia dopravnej značky je realizované v triede `Input_Parse`. Táto trieda obsahuje metódy pre kontrolu správnosti xml súboru a pre samotné parsovanie.

4.2.2 Vypočítanie GPS súradníc

Problém dopočítania GPS súradníc a jeho riešenie bolo detailne opísané v kapitole Návrh. Je realizované v triede `GPS`, ktorá poskytuje dve public metódy `Map_Get_Lat()` a `Map_Get_Lon()`, ktoré na základe časovej stopy rozpoznannej dopravnej značky priradia dopravnej značke jej súradnice *latitude* a *longitude*.

Rozhranie Pluginu

Rozhranie Pluginu je realizované v triede `Map`, ktoré zobrazuje užívateľovi informácie o nájdenej značke, jej súradnice a príslušný obrázok. Užívateľovi sú formou tlačidiel *Yes* a *No* ponúkné možnosť zaviesť značku do mapy editoru JOSM. V prípade ak chce užívateľ zmeniť typ dopravnej značky z dôvodu chybnjej klasifikácie alebo z dôvodu, že sa daná dopravná značka vôbec neklasifikovala je užívateľovi táto zmena ponúknutá vo forme jednoduchého `ComboBox`, kde sa užívateľovi zobrazia textové názvy všetkých možných dopravných značiek na výber, kde sa on rozhodne.



Obr. 22: Zásuvný modul pre editor JOSM.

4.3 Testovanie

Testovanie aplikácie prebehlo na operačnom systéme linux Ubuntu 14.04, pod ktorým táto aplikácia bola aj vyvíjaná. Testovanie prebehlo na základe mnou získaných 10 videozáznamov, ktoré boli vybrané s ohľadom na ich rôznorodosť. Volil som tak na základe viacerých kritérií. Tieto kritéria sú nasledujúce:

- Rôzne svetelné podmienky: Pre testovanie som si zvolil videá, ktoré boli natočené pri jasnom slnečnom počasí, a tie ktoré boli natočené počas sychravého počasia. Aby som vedel zistiť aký vplyv má úroveň svetla na kvalitu detekcie a klasifikácie. Vhodné by bolo za testovacie videa požiť ešte videa natočené v noci, a videá natočené za daždivého počasia.
- Rôzne uhly pozorovania značky, rôzne úrovne stabilizácie: Toto kritérium, bolo určené pre výber testovacích videí bolo myslené nasledujúcim spôsobom. Niektoré videá boli získané, za použitia stabilizátora kamery, ktorý podstatnou mierou ovplyvňuje plynulosť videa a tým pádom aj kvalitu detekcie a klasifikácie. Niektoré videa zase boli získané bez použitia stabilizátora kamery, kde hneď už na prvý pohľad je jasné, že tieto videá nie sú úplne plynulé. Ďalším kritériom boli rôzne uhly pozorovania dopravné značky. Kde na niektorých videách bol záber kamery zameraný výhradne na svoj vlastný jazdný pruh a na niektorých videách bol záber kamery zameraný na celú cestu, čiže aj na opačný jazdný pruh a na cesty, ktoré sa pripájali .
- Rôzne farby a geometrické útvary dopravných značiek: Toto kritérium berie do úvahy farbu a geometrické útvary dopravných značiek. Do úvahy sa vzali tieto 3 rôzne typy farieb dopravných značiek, žltá, červená a modrá. A tieto rôzne geometrické tvary značiek kruh, obdĺžnik, kosoštvorec, trojuholník a šesťuholník .

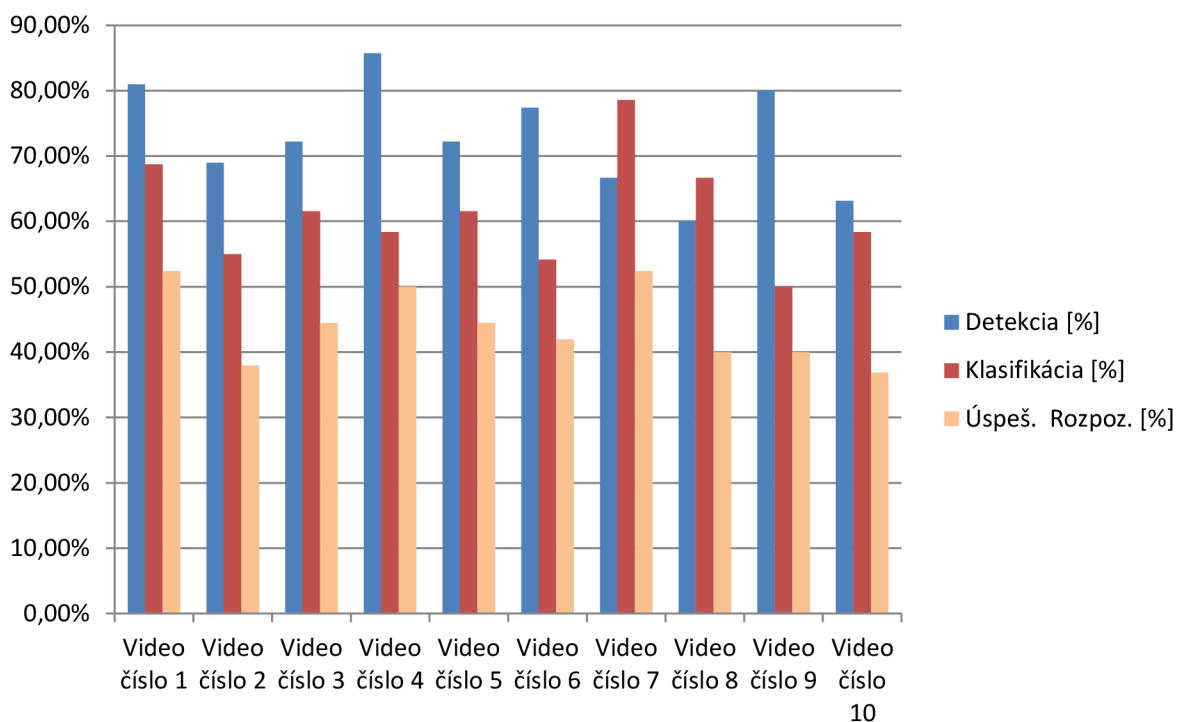
4.4 Výsledky testovania

Všetky videa spolu		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	41	10	80,39%	24	17	58,54%	47,06%
	Trojuholníkové	37	10	78,72%	20	17	54,05%	42,55%
	Šesťuholníkové	0	2	0,00%	0	0	N/A	0,00%
Modré	Okrúhle	11	5	68,75%	2	8	20,00%	12,50%
	Štvoruholníkové	40	20	66,67%	28	12	70,00%	46,67%
Žlté	Kosoštvorcové	16	9	64,00%	14	2	87,50%	56,00%
Všetky značky spolu		145	56	72,14%	88	56	61,11%	43,78%

Tabuľka 2: Zhodnotenie výsledkov klasifikácie dopravných značiek podľa typu značky

	Detekcia			Klasifikácia			Úspeš. Rozpoz. [%]
	Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Video číslo 1	17	4	80,95%	11	5	68,75%	52,38%
Video číslo 2	20	9	68,97%	11	9	55,00%	37,93%
Video číslo 3	13	5	72,22%	8	5	61,54%	44,44%
Video číslo 4	12	2	85,71%	7	5	58,33%	50,00%
Video číslo 5	13	5	72,22%	8	5	61,54%	44,44%
Video číslo 6	24	7	77,42%	13	11	54,17%	41,94%
Video číslo 7	14	7	66,67%	11	3	78,57%	52,38%
Video číslo 8	12	8	60,00%	8	4	66,67%	40,00%
Video číslo 9	8	2	80,00%	4	4	50,00%	40,00%
Video číslo 10	12	7	63,16%	7	5	58,33%	36,84%
Spolu	145	56	72,14%	88	56	61,11%	43,78%

Tabuľka 3: Zhodnotenie výsledkov klasifikácie dopravných značiek



Graf 1. Graf reprezentujúci sumárne výsledky detekcia a klasifikácie dopravných značiek

4.5 Zhodnotenie výsledkov testovania

4.5.1 Vplyv svetelných podmienok

Najúspešnejšie dopadli detekcie dopravných značiek z videí č.4, č.9 a č.1. Najhoršie to však bolo vo videu č.10. Svetelné podmienky majú na to určitý vplyv, ale menší ako som prvotne očakával. Je to z nasledujúceho dôvodu a to, že najúspešnejšie obstála detekcia vo videách s optimálnymi svetelnými podmienkami videa č.4 a č.1 tak aj s nie úplne ideálnymi svetelnými podmienkami video č.9.

4.5.2 Vplyv stabilizácie videa

Keďže najhoršie dopadla detekcia vo videu č.10, ktorá je výrazne nižšia ako u ostatných videí bude to určite zapríčinené aj tým, že pri natáčaní tohto videa nebola použitá žiadna stabilizácia kamery, čo je hneď vidno z prvého pohľadu na toto video.

4.5.3 Vplyv farieb a geometrických tvarov

Vplyv farieb a geometrických tvarov na úspešnosť detekcie a klasifikácie je veľmi vysoký. Najúspešnejšie dopadla detekcia červených okrúhlych značiek. Najhoršie dopadla detekcia červených šesťuholníkových značiek. Preto moja aplikácia nie je vhodná pre rozpoznávanie šesťuholníkových značiek (STOP značka). Úspešnosť klasifikácie ak neberieme do úvahy šesťuholníkové značky je u modrých kruhových značiek len 20%. Je to zapríčinené nedostatočne hlbokou analýzou týchto značiek a nedostatočným počtom tréningových dát. Preto je vhodné použiť lepšie tréningové dáta a vyšší počet tréningových dát, ktoré by mohli zlepšiť úspešnosť klasifikácie týchto typov dopravných značiek..

5 Záver

V bakalárskej práci som sa zaoberal rozpoznávaním dopravných značiek z videa a ich zavedením pomocou zásuvného modulu do mapy. Počas celého procesu tvorby bakalárskej práce som prišiel do styku s viacerými metódami detekcie a klasifikácie obrazu. Zoznámil som sa s projektom OpenStreetMap a jeho editorom JOSM, ktorý slúži pre zavedenie rozpoznaných značiek formou môjho pluginu. Počas testovania som experimentoval s viacerými farebnými modelmi a spôsobmi klasifikácie. Od svojho úmyslu klasifikovať za použitia metódy *TemplateMatching* som musel upustiť hneď spočiatku, pretože táto metóda mala 0% úspešnosť. Preto som sa rozhodol plne zamerať na klasifikačnú metódu SVM, ktorá obsahuje 36 typov trénovacích sád, ktoré pokrývajú dopravné značky v prílohe A. Na úrovni klasifikácie, či je objekt vhodným kandidátom na dopravnú značku som pre natrénovanie klasifikátora využil 206 snímok, ktoré reprezentujú pozitívnu trénovaciu sadu a 323 snímok, ktoré reprezentujú negatívnu trénovaciu sadu.

Na testovacej sade pozostávajúcej celkovo z desiatich videozáznamov, bola priemerná úspešnosť detekcie je 72,14% a klasifikácie 61,11%. Pre zlepšenie úspešnosti detekcie by bolo vhodné experimentovať s jednotlivými hodnotami prahov pri segmentácii prahovaním alebo experimentovať s inými lepšími metódami segmentácie. Pre zlepšenie úspešnosti klasifikácie by bolo vhodné experimentovať s viacerými typmi príznakov, ktoré popisujú objekt, takisto aj experimentovať s inými klasifikačnými metódami a s rozšírením trénovacej sady, ktorú si môže užívateľ sám rozšíriť. Na server OpenStreetMap bolo pomocou pluginu vložených 145 značiek.

Medzi možné rozšírenia aplikácie sa ponúka niekoľko možností. Najvhodnejším rozšírením by určite bolo rozpoznávanie dopravných značiek a zavedenie do mapy v reálnom čase, kde by užívateľ mal len kontrolnú úlohu, korigoval by len chybné rozpoznania. Ďalším možným rozšírením je plná podpora dopravných značiek, ktoré sa nachádzajú v cestnej premávke s možnosťou rozpoznania textu na informatívnych značkách a značkách, ktoré signalizujú začiatok alebo koniec obce. Medzi možné rozšírenie Pluginu sa ponúka hlavne možnosť tzv. PhotoMappingu. Jedná sa o v podstate o to, že ak užívateľ klikne v mape na danú dopravnú značku, ukáže sa aj presný snímok na ktorom sa daná dopravná značka našla

Literatúra

- [1] OpenStreetMap stats report run at 2014 *OpenStreetMap* [online]. 2004 [cit. 2014-12-28].
Dostupné z: http://www.openstreetmap.org/stats/data_stats.html
- [2] JOSM. *JOSM* [online]. 2014 [cit. 2014-12-28].
Dostupné z: <http://josm.openstreetmap.de>
- [3] Freemap Slovakia. *FreeMapSK* [online]. 2007 [cit. 2014-12-28].
Dostupné z: <http://wiki.freemap.sk/StarterGuideKonfigJOSM>
- [4] JOSM Download. *JOSM* [online]. 2014 [cit. 2014-12-28].
Dostupné z: <http://josm.openstreetmap.de/wiki/Download>
- [5] Colour Models. *Brown: Computer Science* [online]. 2015 [cit. 2015-05-18]. Dostupné z:
<http://cs.brown.edu/courses/cs092/VA10/HTML/ColorModels.html>
- [6] Počítačová grafika: Barevný model, DPI. *Palackého Univerzita, Olomouc: Katedra technické a informační výchovy* [online]. 2008 [cit. 2015-05-16]. Dostupné z:
<http://www.kteiv.upol.cz/frvs/ict-kubricky/?page=pocitacova-grafika/barevny-model-dpi>
- [7] Farebný model CIE. *FIT STUBA* [online]. 2015 [cit. 2015-05-16].
Dostupné z: http://student.fiit.stuba.sk/~pifkova04/farebne_modely/cie_yuv.html
- [8] Pracujeme s barvou na počítači. *DIGI arena.cz* [online]. 2015 [cit. 2015-05-16].
Dostupné z: http://digiarena.e15.cz/pracujeme-s-barvou-na-pocitaci_4/ch-47505
- [9] *Spring Conference on Computer Graphics* [online]. 2014 [cit. 2015-01-25]. Dostupné z:
<http://www.sccg.sk/~ftacnik/IP-5.pdf>
- [10] GUO, H. A Simple Algorithm for Fitting a Gaussian Function [DSP Tips and Tricks]. *Signal Processing Magazine, IEEE* [online]. 2011, (5) [cit. 2015-05-16].
DOI: 10.1109/MSP.2011.941846. ISSN 1053-5888.
Dostupné z: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5999593>
- [11] Shapiro, L. G. & Stockman, G. C: "Computer Vision", page 137, 150. Prentice Hall, 2001
- [12] AUTOR NEUVEDEN. *skeletonmaker* [online]. [cit. 19.1.2015].
Dostupný na WWW: <https://code.google.com/p/skeletonmaker/>
- [13] *Metódy Rozpoznání objektu v obraze* [online]. 2014 [cit. 2015-01-25].
Dostupné z:
<http://www.fbmi.cvut.cz/files/predmety/3528/public/Metody%20rozpozn%C3%A1n%C3%AD%20objekt%C5%AF%20v%20obrazu.pdf>

- [14] Knihovna pro extrakci příznaků z obrazových dat. 2011. Ing. Martin ZUKAL, a Ing. Radim BURGET, Ph.D. *Signal processing laboratory* [online]. [cit. 2015-05-16].
Dostupné z: <http://splab.cz/download/software/knihovna-pro-extrakci-priznaku>
- [15] *Hough Transform* [online]. 2014 [cit. 2015-01-25]. Dostupné z:
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>
- [16] MAI, HUNG, ZHONG. *Evolution of the Hough Transform* [online]. [cit. 19.1.2015].
Dostupný na WWW:
http://blog.daum.net/_blog/BlogTypeView.do?blogid=0QhP0&articaleno=13
- [17] Pattern recognition systems - Lab 5: Histograms of Oriented Gradients. *Univesitatea Technica Cluj Napoca* [online]. 2011 [cit. 2015-05-16].
Dostupné z: http://users.utcluj.ro/~raluca/prs/prs_lab_05e.pdf
- [18] JURŠICA, Ladislav, Peter HUBINSKÝ a Juraj KNOT. Detekcia a Sledovanie Objektov (1). In: *Atp journal* [online]. 2010 [cit. 2014-12-29].
Dostupné z: <http://www.atpjournal.sk/buxus/docs/atp-2005-06-69.pdf>
- [19] Generalized image matching by the method of differences. LUCAS, a Bruce DAVID. *Carnegie Mellon University* [online]. 1985 [cit. 2015-05-16].
Dostupné z:
http://www.ri.cmu.edu/pub_files/pub4/lucas_bruce_d_1984_1/lucas_bruce_d_1984_1.pdf
- [20] Image denoising: pointwise adaptive approach. POSEHL, Jorg a Vladimír SPOKOJNÝ. *Project euclid* [online]. 2015 [cit. 2015-05-16].
Dostupné z: <https://projecteuclid.org/euclid.aos/1046294457>
- [21] CAETANO, Mário. *ESA ADVANCED TRAINING COURSE ON LAND REMOTE SENSING: Image Classification* [online]. 2009 [cit. 2015-01-25]. Dostupné z:
http://earth.esa.int/landtraining09/D2L2_Caetano_Classification_Techniques.pdf
- [22] Neuronové sítě. JIŘINA, Marcel. *České vysoké učení technické v Praze: Fakulta Kybernetiky* [online]. 2015 [cit. 2015-01-25].
Dostupné z: http://gerstner.felk.cvut.cz/biolab/33KP/prednasky_ann/prezentace_ns.ppt
- [23] *Template Matching* [online]. 2007 [cit. 2015-01-25].
Dostupné z:
http://docs.adaptivevision.com/current/studio/machine_vision_guide/TemplateMatching.html

- [24] Bernhard Schölkopf, Alex Smola: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning), MIT Press, Cambridge, MA, 2002, [ISBN 0-262-19475-9](#)
- [25] Introduction to Support Vector Machines. SPOKOJNÝ, Vladimír. *OpenCV* [online]. 2015 [cit. 2015-05-16]. Dostupné z: http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.htm
- [26] iPhone5 -Technical Specification. *Apple* [online]. 2015 [cit. 2015-01-25]. Dostupné z: <http://support.apple.com/kb/SP655>
- [27] iTunes Preview: Record my GPS position. *Apple* [online]. 2015 [cit. 2015-01-25]. Dostupné z: <https://itunes.apple.com/us/app/record-my-gps-position/id405865492?mt=8>
- [28] iTunes Preview: Car Camera DVR. Dashboard GPS Black Box. *Apple* [online]. 2015 [cit. 2015-01-25]. Dostupné z <https://itunes.apple.com/us/app/car-camera-dvr.-dashboard/id557827515?mt=8>

Príloha A Zoznam rozpoznávaných značiek :

Výstražné značky:

1. Zákruta vpravo
2. Zákruta vľavo
3. Nerovnosť vozovky
4. Nebezpečenstvo šmyku
5. Sneh alebo poľadovica
6. Priechod pre chodcov
7. Práca na ceste
8. Iné nebezpečenstvo

Značky upravujúce prednosť a dodatkové tabuľky s tvarom križovatky:

1. Daj prednosť v jazde
2. Stop, daj prednosť v jazde
3. Križovatka s vedľajšou cestou
4. Križovatka s vedľajšou cestou vpravo
5. Križovatka s vedľajšou cestou vľavo
6. Hlavná cesta

Zákazové značky:

1. Zákaz vjazdu všetkých vozidiel v oboch smeroch
2. Zákaz vjazdu všetkých vozidiel
3. Zákaz vjazdu nákladných automobilov
4. Zákaz vjazdu vozidiel , ktorých okamžitá hmotnosť presahuje určitú hranicu
5. Zákaz odbočovania vpravo
6. Zákaz odbočovania vľavo
7. Zákaz predchádzania
8. Zákaz zastavenia
9. Najvyššia dovolená rýchlosť 40km/h
10. Najvyššia dovolená rýchlosť 50km/h
11. Najvyššia dovolená rýchlosť 60km/h
12. Najvyššia dovolená rýchlosť 70km/h
13. Najvyššia dovolená rýchlosť 80km/h

Príkazové značky:

1. Príkázaný smer obchádzania vpravo
2. Príkázaný smer obchádzania vľavo
3. Príkázaný smer obchádzania vpravo alebo vľavo
4. Príkázaný smer jazdy
5. Kruhový objazd

Informatívne prevádzkové, smerové a iné značky:

1. Jednosmerná premávka
2. Priechod pre chodcov
3. Odporúčaná rýchlosť 70km/h

Informatívne iné značky:

1. Zastávka autobusu
2. Čerpacia stanica
3. Opraváreň
4. Reštaurácia
5. Hotel alebo motel
6. Táborisko pre stany a obytné prívesy
7. Sedačkový vlek

Príloha B

Výsledky testov

Video číslo 1		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	1	0	100,00%	1	0	100,00%	100,00%
	Trojuholníkové	6	1	85,71%	4	2	66,67%	57,14%
	Šesťuholníkové	0	0	N/A	0	0	N/A	N/A
Modré	Okrúhle	1	0	100,00%	0	0	N/A	0,00%
	Štvoruholníkové	8	2	80,00%	5	3	62,50%	50,00%
Žlté	Kosoštvorcové	1	1	50,00%	1	0	100,00%	50,00%
Všetky značky spolu		17	4	80,95%	11	5	68,75%	52,38%

Tabuľka 4: Výsledky testovania – video č.1

Video číslo 2		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	8	4	66,67%	5	3	62,50%	41,67%
	Trojuholníkové	5	1	83,33%	3	2	60,00%	50,00%
	Šesťuholníkové	0	0	N/A	0	0	N/A	N/A
Modré	Okrúhle	2	1	66,67%	0	2	0,00%	0,00%
	Štvoruholníkové	5	2	71,43%	3	2	60,00%	42,86%
Žlté	Kosoštvorcové	0	1	0,00%	0	0	N/A	0,00%
Všetky značky spolu		20	9	68,97%	11	9	55,00%	37,93%

Tabuľka 5: Výsledky testovania video č.2

Video číslo 3		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	7	2	77,78%	4	3	57,14%	44,44%
	Trojuholníkové	5	1	83,33%	3	2	60,00%	50,00%
	Šesťuholníkové	0	1	0,00%	0	0	N/A	N/A
Modré	Okrúhle	0	0	N/A	0	0	N/A	N/A
	Štvoruholníkové	1	1	50,00%	1	0	100,00%	50,00%
Žlté	Kosoštvorcové	0	0	N/A	0	0	N/A	N/A
Všetky značky spolu		13	5	72,22%	8	5	61,54%	44,44%

Tabuľka 6: Výsledky testovania video č.3

Video číslo 4		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	5	0	100,00%	2	3	40,00%	40,00%
	Trojuholníkové	5	1	83,33%	4	1	80,00%	66,67%
	Šesťuholníkové	0	0	N/A	0	0	N/A	N/A
Modré	Okrúhle	0	0	N/A	0	0	N/A	N/A
	Štvoruholníkové	1	1	50,00%	0	1	0,00%	0,00%
Žlté	Kosoštvorcové	1	0	100,00%	1	0	100,00%	100,00%
Všetky značky spolu		12	2	85,71%	7	5	58,33%	50,00%

Tabuľka 7: Výsledky testovania video č.4

Video číslo 5		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	4	1	80,00%	3	1	75,00%	60,00%
	Trojuholníkové	4	1	80,00%	2	2	50,00%	40,00%
	Šesťuholníkové	0	0	N/A	0	0	N/A	N/A
Modré	Okrúhle	0	1	0,00%	0	0	N/A	0,00%
	Štvoruholníkové	4	1	80,00%	2	2	50,00%	40,00%
Žlté	Kosoštvorcové	1	1	50,00%	1	0	100,00%	50,00%
Všetky značky spolu		13	5	72,22%	8	5	61,54%	44,44%

Tabuľka 8: Výsledky testovania video č.5

Video číslo 6		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	1	1	50,00%	0	1	0,00%	0,00%
	Trojuholníkové	6	0	100,00%	3	3	50,00%	50,00%
	Šesťuholníkové	0	0	N/A	0	0	N/A	N/A
Modré	Okrúhle	4	2	66,67%	0	4	0,00%	0,00%
	Štvoruholníkové	9	2	81,82%	6	3	66,67%	54,55%
Žlté	Kosoštvorcové	4	2	66,67%	4	0	100,00%	66,67%
Všetky značky spolu		24	7	77,42%	13	11	54,17%	41,94%

Tabuľka 9: Výsledky testovania video č.6

Video číslo 7		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	4	0	100,00%	3	1	75,00%	75,00%
	Trojuholníkové	1	2	33,33%	0	1	0,00%	0,00%
	Šesťuholníkové	0	0	N/A	0	0	N/A	N/A
Modré	Okrúhle	0	1	0,00%	0	0	N/A	0,00%
	Štvoruholníkové	5	2	71,43%	4	1	80,00%	57,14%
Žlté	Kosoštvorcové	4	2	66,67%	4	0	100,00%	66,67%
Všetky značky spolu		14	7	66,67%	11	3	78,57%	52,38%

Tabuľka 10: Výsledky testovania video č.7

Video číslo 8		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	5	0	100,00%	3	2	60,00%	60,00%
	Trojuholníkové	1	1	50,00%	0	1	0,00%	0,00%
	Šesťuholníkové	0	0	N/A	0	0	N/A	N/A
Modré	Okrúhle	1	0	100,00%	1	0	100,00%	100,00%
	Štvoruholníkové	4	7	36,36%	4	0	100,00%	36,36%
Žlté	Kosoštvorcové	1	0	100,00%	0	1	0,00%	0,00%
Všetky značky spolu		12	8	60,00%	8	4	66,67%	40,00%

Tabuľka 11: Výsledky testovania video č.8

Video číslo 9		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	2	0	100,00%	1	1	50,00%	50,00%
	Trojuholníkové	2	0	100,00%	0	2	0,00%	0,00%
	Šesťuholníkové	0	0	0,00%	0	0	N/A	N/A
Modré	Okrúhle	1	0	100,00%	0	1	0,00%	0,00%
	Štvoruholníkové	1	1	50,00%	1	0	100,00%	50,00%
Žlté	Kosoštvorcové	2	1	66,67%	2	0	100,00%	66,67%
Všetky značky spolu		8	2	80,00%	4	4	50,00%	40,00%

Tabuľka 12: Výsledky testovania video č.9

Video číslo 10		Detekcia			Klasifikácia			Úspešnosť rozpoznávania [%]
		Poz.	Neg.	Úspeš. [%]	Poz.	Neg.	Úspeš. [%]	
Červené	Okrúhle	4	2	66,67%	2	2	50,00%	33,33%
	Trojuholníkové	2	2	50,00%	1	1	50,00%	25,00%
	Šesťuholníkové	0	1	0,00%	0	0	N/A	0,00%
Modré	Okrúhle	2	0	100,00%	1	1	50,00%	50,00%
	Štvoruholníkové	2	1	66,67%	2	0	100,00%	66,67%
Žlté	Kosoštvorcové	2	1	66,67%	1	1	50,00%	33,33%
Všetky značky spolu		12	7	63,16%	7	5	58,33%	36,84%

Tabuľka 13: Výsledky testovania video č.10

Príloha C Obsah DVD

- **Aplikacia** - Adresár s aplikáciou
 - src – zdrojové súbory
 - Blue_Rect_Signs - trénovacie data pre modré obdĺžnikové značky
 - Blue_Ellipse_Signs - trénovacie data pre modré kruhové značky
 - Red_Ellipse_Signs – trénovacie data pre červené kruhové značky
 - Red_Triangle_Signs – trénovacie data pre červené trojuholníkové značky
 - Yellow_Square_Signs -trénovacie data pre žlté štvorcové značky
 - videa- ukážky videí
 - Najdene_Znacky – uložiisko všetkých nájdených značiek z videa
 - Video_File_Output – výstupné súbory so značkami
 - SVM_Models – SVM klasifikátor
 - BP_Obraz- binárny spustiteľný súbor pod linuxom
 - make - makefile
 - lib- externé knižnice
- **technicka-sprava .pdf** -Technicka správa vo formate PDF
- **Plugin-** adresár s pluginom
 - doc-programová dokumentácia
 - jar – Samotný plugin do JOSM
 - src- zdrojové kódy pluginu
- **README**

