



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# **ÚSEKOVÉ MĚŘENÍ RYCHLOSTI PRO ANALÝZU DOPRAVY**

SECTION SPEED MEASUREMENT FOR TRAFFIC ANALYSIS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**PAVLA KUBÍČKOVÁ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. JAKUB SOCHOR**

BRNO 2016

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

**Zadání bakalářské práce**

Řešitel: **Kubíčková Pavla**

Obor: Informační technologie

Téma: **Úsekové měření rychlosti pro analýzu dopravy**  
**Section Speed Measurement for Traffic Analysis**

Kategorie: Zpracování obrazu

**Pokyny:**

1. Nastudujte algoritmy pro detekci a sledování vozidel ve videu.
2. Nastudujte algoritmy pro detekci a rozpoznání registračních značek automobilů v obraze.
3. Navrhněte systém, který bude využívat dvou stacionárních kamer pro úsekové měření rychlosti.
4. Navržený systém implementujte.
5. Pořídte vhodnou datovou sadu pro vyhodnocení přesnosti systému a proveďte toto vyhodnocení.
6. Vytvořte plakát a video pro prezentaci vaší práce.

**Literatura:**

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 a 4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Sochor Jakub, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Tato bakalářská práce se zabývá úsekovým měřením rychlosti automobilů. Práce popisuje existující metody detekce registračních značek a rozpoznání znaků v nich obsažených. Následně jsou popsány metody kaskádového klasifikátoru a klasifikátoru SVM, které jsou v této práci použity. V závěrečné části je zpracované vyhodnocení jednotlivých částí systému.

## Abstract

This bachelor thesis focuses on section speed measurement for traffic analysis. This thesis describes existing methods of detection of license plates and classification of their characters. Methods of cascade classifier and classifier SVM are described in this work. Evaluation of individual parts of the system is processed in the final section.

## Klíčová slova

kaskádový klasifikátor, klasifikátor SVM, detekce registračních značek, optické rozpoznání znaků, měření rychlosti

## Keywords

cascade classifier, classifier SVM, license plate detection, optical character recognition, speed measurement

## Citace

KUBÍČKOVÁ, Pavla. *Úsekové měření rychlosti pro analýzu dopravy*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Sochor Jakub.

# Úsekové měření rychlosti pro analýzu dopravy

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Jakuba Sochora. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....  
Pavla Kubíčková  
18. května 2016

## Poděkování

Ráda bych poděkovala vedoucímu Ing. Jakubovi Sochorovi za vedení a odbornou pomoc, kterou mi poskytoval v průběhu řešení práce.

© Pavla Kubíčková, 2016.

*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>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Registrační značky</b>	<b>4</b>
2.1	Poznávací značky u nás . . . . .	4
2.2	Změna evidence a nové registrační značky z roku 2001 . . . . .	6
<b>3</b>	<b>Existující řešení</b>	<b>7</b>
3.1	Existující metody detekce registračních značek . . . . .	7
3.1.1	Metoda kombinující Houghovu transformaci a detekci kontur . . . . .	7
3.1.2	Metoda Houghovy transformace pro lokalizaci registračních značek . . . . .	9
3.1.3	Lokalizace registrační značky založena na histogramu a morfologických operacích . . . . .	9
3.1.4	Metoda detekce registrační značky založená na barevném obrazu . . . . .	12
3.2	Existující metody klasifikace znaků . . . . .	13
3.2.1	Porovnání s předem definovanou šablonou . . . . .	13
3.2.2	Metoda založená na analýze oblasti znaku . . . . .	14
3.2.3	Algoritmus založený na neuronových sítích . . . . .	15
<b>4</b>	<b>Zpracování obrazu</b>	<b>16</b>
4.1	Vyhlazování obrazu . . . . .	16
4.2	Prahování . . . . .	17
4.3	Detekce hran . . . . .	19
4.4	Morfologické operace . . . . .	21
4.4.1	Dilatace . . . . .	21
4.4.2	Eroze . . . . .	21
<b>5</b>	<b>Použité metody pro zpracování obrazu</b>	<b>23</b>
5.1	Odečítání pozadí . . . . .	23
5.1.1	Předzpracování . . . . .	23
5.1.2	Modelování pozadí . . . . .	23
5.1.3	Mixture of Gaussians (MoG) . . . . .	24
5.2	Kaskádový klasifikátor . . . . .	25
5.2.1	Integrální obraz . . . . .	25
5.2.2	Haarovy vlnky . . . . .	26
5.2.3	AdaBoost . . . . .	26
5.2.4	Detekce objektů v obraze . . . . .	27
5.3	Support vector machines . . . . .	28
5.3.1	Lineární SVM . . . . .	28

5.3.2	Multitřídni SVM . . . . .	28
<b>6</b>	<b>Návrh aplikace</b>	<b>30</b>
6.1	Detekce registrační značky . . . . .	30
6.1.1	Efektivnější přístup detekce registrační značky . . . . .	32
6.2	Redukce falešných kandidátů registračních značek . . . . .	32
6.3	Klasifikace znaků . . . . .	32
<b>7</b>	<b>Implementace</b>	<b>33</b>
7.1	Použité nástroje . . . . .	33
7.2	Trénování kaskádového klasifikátoru . . . . .	33
7.3	Trénování SVM klasifikátoru . . . . .	34
<b>8</b>	<b>Vyhodnocení</b>	<b>36</b>
8.1	Vyhodnocení detekce registrační značky . . . . .	36
8.2	Vyhodnocení klasifikace znaků . . . . .	38
8.3	Vyhodnocení rozpoznání registrační značky . . . . .	40
8.4	Vyhodnocení reidentifikace vozidel . . . . .	41
8.5	Vyhodnocení rychlosti automobilů . . . . .	41
<b>9</b>	<b>Závěr</b>	<b>43</b>
	<b>Literatura</b>	<b>44</b>
<b>A</b>	<b>Obsah DVD</b>	<b>46</b>

# Kapitola 1

## Úvod

Úsekového měření rychlosti lze využít ve velkých městech, kde je možné pomocí rozložení stacionárních kamer provádět měření rychlosti automobilů na libovolných úsecích vozovky. Pro zlepšení funkce měření můžeme rozmístit hned několik vstupních a výstupních kamer. Tyto kamery však nemusí měřit pouze rychlosti vozidel, ale mohou provádět statistiky o hustotě provozu v různých částech města, nebo je lze využít při pátrání po určitém vozidle. Nyní se již využívá úsekového měření také na dálnicích, kde jsou kamery instalovány na mýtných branách. Lze tak jednoduše zjistit, jakou průměrnou rychlostí vozidla jedou. Oproti měření pomocí radaru má úsekové měření tu výhodu, že může pokrýt mnohem větší oblast.

Cílem bakalářské práce bylo rozpoznat registrační značky automobilů pořízených stacionárními kamerami. Na základě takto získaných dat vypočítat úsekovou rychlost jednotlivých automobilů. Dále nastudovat algoritmy pro detekci a rozpoznání registračních značek automobilů v obraze.

V první kapitole je popsána historie registračních značek a jejich postupné změny s přihlédnutím k vývoji Evropských zemí ve 20. a 21. století. V následující kapitole se nachází popis existujících metod pro detekci registračních značek a následně možné metody k rozpoznání registračních značek. Třetí kapitola nás seznamuje se základními operacemi zpracování obrazu, které se využívají k potlačení nežádoucích deformací nebo naopak k vylepšení obrazu pro další zpracování. Dále jsou zde popsány metody, které byly využity k implementaci systému. V kapitolách Návrh systému a Implementace je popsáno fungování celého systému. V poslední kapitole je vyhodnocena úspěšnost jednotlivých kroků.

## Kapitola 2

# Registrační značky

V této kapitole bylo čerpáno z knihy Filipa Zeleného a Dalibora Feereisla [18]. Doplnující informace byly získány od Michala Hájka[9].

První automobily se objevily v roce 1886. S postupným přibýváním automobilů koncem 19. století bylo potřeba identifikovat vozidla. Ve velkých městech byla vozidla evidována a označena identifikačními tabulkami. První označení bylo umístěno na levé straně vozidla. Tyto registrační značky na sobě nesly označení majitele, jeho jméno, adresu a registrační číslo. Tento způsob značení byl poprvé použit v Paříži roku 1893. Později se rozšířilo do dalších měst. Poznávací značky podobné dnešním registračním značkám se objevily v roce 1895 v Lucembursku. Byly umístěné v přední i zadní části vozidla a poprvé na sobě nesly jen čísla a písmena. Tato označení byla přidělována státem na celém území Lucemburska.

11. října 1908 dorazil tento nový systém značení i do Francie a postupně se začal používat i v dalších zemích. Značky byly v této době černé s bílými symboly.

### 2.1 Poznávací značky u nás

V době, kdy byla v Evropě povinná registrace vozidel, byly České země součástí Rakouska-Uherska. Poznávací značka používaná na našem území obsahovala číslice od 1 do 999 a písmeno, které značilo oblast na území Rakouska-Uherska. Tabulky byly bílé s černými znaky. Zadní registrační značka obsahovala dva řádky. V horním řádku bylo písmeno případně římská číslice a ve spodním evidenční číslo vozidla. Přední značky byly většinou jednořádkové. Změna systému značení vozidel proběhla roku 1932, kdy začaly být automobily označovány černými tabulkami s bílým písmem a bílou linkou na okraji znázorněné na obrázku 4.5. Na značce se nacházelo opět písmeno, které označovalo zemi, pomlčka a nejvýše pětimístné číslo. Číslice mezi tisícovým a stovkovým číslem jsou odděleny tečkou. Registrační značka mohla být připevněna k vozidlu jako tabulka, nebo napsaná přímo na vozidle.



Obrázek 2.1: Ukázka poznávací značky po roce 1932. Převzato z [6].



V době druhé světové válce v letech 1938 až 1945 došlo k několika změnám. Německé úřady začaly na obsazeném území přidělovat říšské poznávací značky. Na zbývajícím území protektorátu Čech a Moravy byly zavedeny nové poznávací značky. Jako rozeznávací značka v protektorátu byla používána dvě písmena z nichž první bylo vždy P a druhé označovalo oblast.

PA	Čechy
PB	Morava
PC	protektorátní pošta a dráhy
PD	Praha

Stejně jako v Německu poznávací značky musely být tabulky orazítkované úřadem, který značku přidělil. Vozidla s povolením používání obsahovala dvě razítka z nichž jedno stvrzovalo registraci a druhé potvrzující červený znak „V“.



Obrázek 2.2: Ukázka poznávací značky ve 40. letech 20.stol. Převzato z [9].

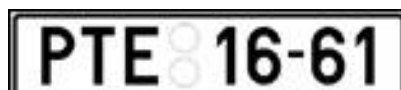
Systém poznávacích značek se v průběhu let měnil. V roce **1960** došlo v Československu k novému správnímu uspořádání. Snížil se počet krajů i okresů. Majitel vozidla, který neměl na SPZ správně uvedený nový okres, v němž měl trvalé bydliště, musel být předvolán k výměně. Dne 26. září 1960 vydalo ministerstvo vnitra nové pokyny pro přidělování SPZ. Nový rozdělovník přiděloval jednotlivé série každém okresu.



Obrázek 2.3: Poznávací značka z let 1960-1986. Převzato z [9].



Obrázek 2.4: Poznávací značka z let 1986-1994. Převzato z [9].



Obrázek 2.5: Poznávací značka z let 1994-2001 [9]

## 2.2 Změna evidence a nové registrační značky z roku 2001

Nový systém poznávacích značek vznikl 1. července 2001 a značky byly označovány jako registrační (RZ). Čitelnost značek se výrazně zhoršila oproti jiným systémům v jiných státech, kde byl kladen důraz na vyšší čitelnost. Registrační značky obsahují 5 až 7 znaků, z nichž alespoň jedním znakem je písmeno a jedním je číslice. První písmeno označuje kraj, ve kterém byla registrační značka vydána. Například Olomoucký kraj je značen písmenem M a Pardubický E.

V České republice existuje několik zvláštních registračních značek:

- značka pro jednorázové použití s omezenou platností, které nelze použít pro mezinárodní provoz
- značka trvale manipulační, která začíná s kódem kraje
- značka pro zkušební účely začínající písmenem „F“
- značka pro vozidla s průkazem historického vozidla bez RZ začínající písmenem „V“

Tabulky RZ se liší také jejich rozměry:

- 520 x 110 mm pro osobní a nákladní automobil, autobus, speciální a přípojně vozidlo
- 340 x 200 mm pro nákladní automobil, autobus, speciální a přípojně vozidlo
- 280 x 200 mm pro osobní automobil
- 280 x 200 mm pro traktor, zvláštní motorové vozidlo a zvláštní přípojně vozidlo
- 200 x 160 mm pro motocykl
- 80 x 110 mm pro moped



Obrázek 2.6: Nové registrační značky od roku 2001

Značky pro pracovníky diplomatických misí měly modrou barvu se žlutým písmem. Pro osoby s diplomatickou imunitou jsou rezervovány první dva znaky DD.

# Kapitola 3

## Existující řešení

V této kapitole jsou popsány existující metody pro detekci registračních značek a následně klasifikaci znaků.

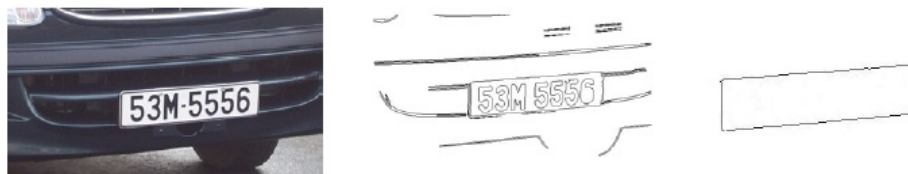
### 3.1 Existující metody detekce registračních značek

#### 3.1.1 Metoda kombinující Houghovu transformaci a detekci kontur

Informace o této metodě byly převzaty z článku Tran Duc Duana a kol. [5]. První část této metody se zabývá předzpracováním obrazu pro detekci registrační značky.

Při tomto postupu je nejdříve obraz upraven funkcemi, ve kterých jsou barvy v obrazu převedeny do odstínů šedi. Poté přichází na řadu normalizace a ekvalizace histogramu. Dále je zde použit Sobelův operátor, který zdůrazní hrany a následně je obraz převeden do binární podoby. Takový obraz se již dá použít jako vstup pro modul, jenž detekuje registrační značku.

Ve druhé části tohoto postupu použijeme pro zjištění oblasti, kde se nachází registrační značka, algoritmus, který zjišťuje obrysy uzavřených objektů. Tento algoritmus však může mít problémy při zpracování méně kvalitních snímků. Někdy vytváří neúplné obrysy uzavřených oblastí, které neidentifikují obraz registrační značky. (viz. obrázek 3.2)



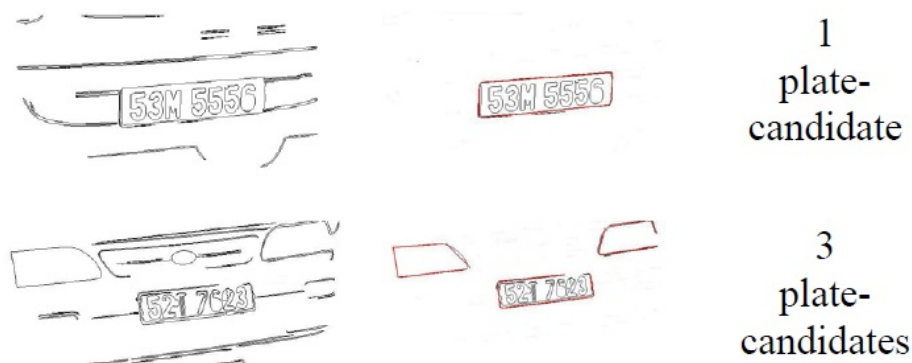
Obrázek 3.1: Úspěšné nalezení registrační značky metodou detekce kontur. Převzato z [5].



Obrázek 3.2: Neúspěšné nalezení registrační značky metodou detekce kontur. Převzato z [5].

V další části je aplikována Houghova transformace na obraz s nalezenými konturami, která vyhledá hrany. Tato transformace se zaměřuje na dvě paralelní přímky, které jsou považovány za potenciální registrační značky. Každá možná registrační značka je ověřována pomocí několika hodnotících algoritmů.

Tento postup je časově méně náročnější než použití Houghovy transformace bez použití vyhledání kontur. Neboť Houghova transformace je aplikována na menší počet obrazových bodů. Objekty jako jsou světla či maska, mohou být falešně detekované jako značka, neboť jsou stejného tvaru a mají dvě paralelní přímky.



Obrázek 3.3: Úspěšná detekce registrační značky. Převzato z [5]

Pro odstranění falešných kandidátů registrační značky je použit modul vyhodnocení, který určí, zda je kandidát registrační značkou. Vyhodnocení může být na základě poměru šířky a výšky detekované oblasti nebo pomocí horizontálních příčných řezů.

Plate-candidates	Number of objects	Result
	2	Non-plate
	2	Non-plate
<del>53M 4893</del>	10	Plate
<del>52T 7013</del>	4	Plate

Obrázek 3.4: Vyhodnocení na základě horizontálního řezu. Převzato z [5]

### 3.1.2 Metoda Houghovy transformace pro lokalizaci registračních značek

Metoda Houghovy transformace k detekci registračních značek byla převzata od autorů Kamata a Ganesana [10].

V této metodě byl na obrázek ve stupních šedi použit Sobelův operátor, jehož úkolem je detekovat hrany. Obvykle hranový detektor vyhledává sousední pixely v horizontálním a vertikálním směru. Zde byl použit modifikovaný Sobelův operátor, který k vyhledání hran používá rohových pixelů. Následně bylo použito adaptivní prahování, které převedlo obraz s detekovanými hranami na binární obraz.

Po dokončení úprav obrazu je aplikována Houghova transformace pomocí níž jsou v obraze detekovány horizontální a vertikální přímky. O registrační značku se jedná v případě nalezení čtyř přímek, které vytváří uzavřenou oblast a splňují následující kritéria:

- protilehlé úsečky mají stejnou velikost
- protilehlé úsečky jsou rovnoběžné
- přilehlé úsečky jsou v pravém úhlu
- poměr délky kolmých úseček je konstantní.

### 3.1.3 Lokalizace registrační značky založena na histogramu a morfologických operacích

Tato metoda využívající histogramy obrazu a matematickou morfologii byla popsána v článku, který publikoval Feng Yang a Zheng Ma [17]. Při psaní bylo čerpáno také z diplomové práce Michala Vacka [15].

Navržený algoritmus znázorněný na obrázku 3.5 má dva hlavní moduly: 1. hrubá detekce registrační značky, 2. přesné určení polohy značky.

V první části je vyhledáno několik kandidátů poznávacích značek. Tyto regiony jsou nalezeny pomocí horizontální projekce vertikálního gradientu.

Druhá fáze detekuje přesnou polohu registrační značky pomocí matematické morfologie a následně extrahuje oblasti registračních značek z daných kandidátů.

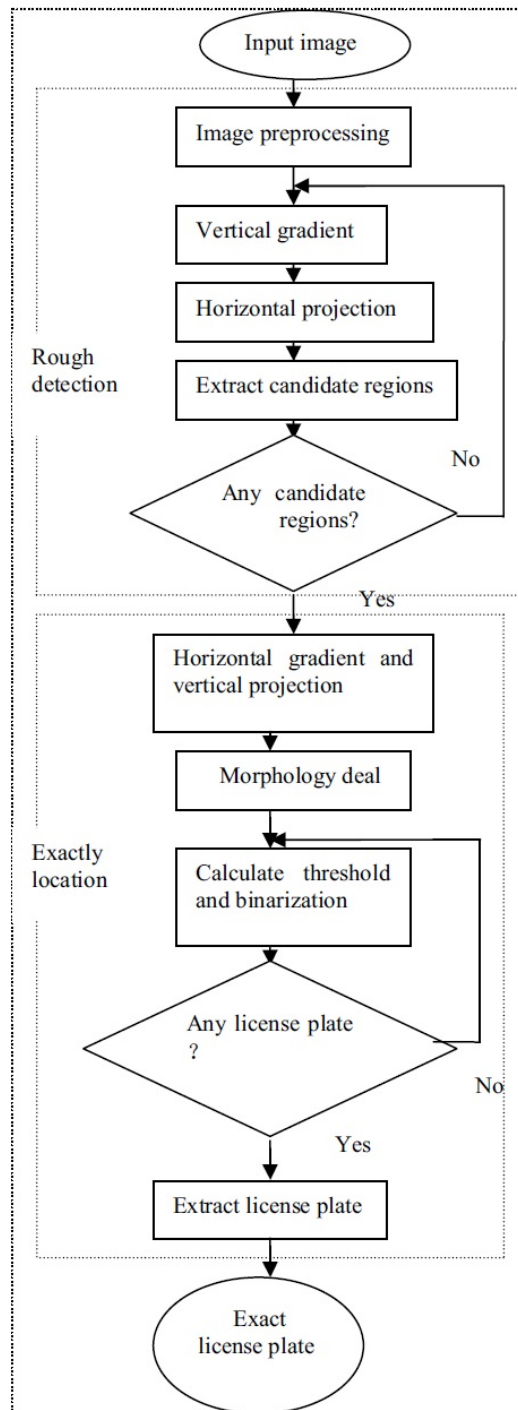
#### Hrubá detekce

Jednou z vlastností registračních značek je, že většina z nich má černé písmo na bílém pozadí. Této vlastnosti lze využít k detekci dané oblasti. Vypočítáme průměrnou hodnotu změny gradientu a porovnáme mezi sebou. V místě, kde je nejvíce přechodů, můžeme předpokládat oblast registrační značky. Výpočet vertikálního gradientu lze realizovat pomocí rovnice:

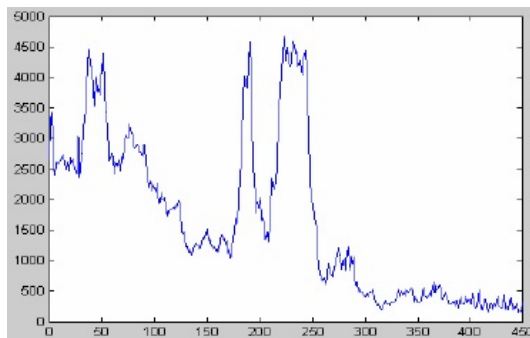
$$g_v(i, j) = |f(i, j - 1) - f(i, j)| \quad (3.1)$$

V oblasti registrační značky je zřejmé, že horizontální projekce vertikálního gradientu bude mít velkou hodnotu. Provedením výpočtu horizontální projekce pomocí vzorce 3.2 získáme horizontální projekci na obrázku 3.6.

$$T_H(i) = \sum_{j=1}^n g_v(i, j) \quad (3.2)$$

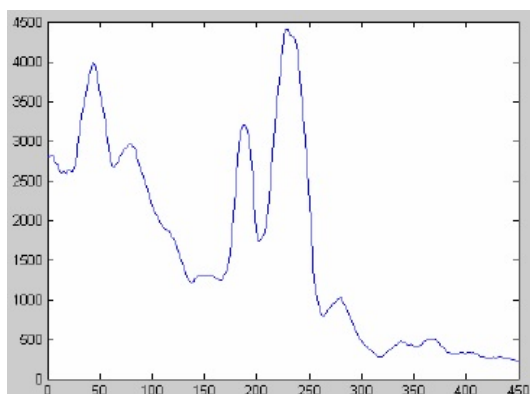


Obrázek 3.5: Vývojový diagram algoritmu. Převzato z [17].



Obrázek 3.6: Horizontální projekce vertikálního gradientu. Převzato z [17].

Horizontální pozice registrační značky musí být vrcholem projekce. Na křivce je zaznamenáno mnoho defektů, proto je použit Gaussov filtr k vyhlazení projekce. Výsledek Gaussova vyhlazování je znázorněn na obrázku 3.7. Poté je možné zjistit vrchol křivky, který určuje oblast značky. Následně k získání kandidátů využíváme vlastností značek. Značky obvykle leží ve spodní části obrazu a maxima horizontální projekce značí možnou horizontální polohu registrační značky.



Obrázek 3.7: Horizontální projekce po Gaussově vyhlazování. Převzato z [17].

## Přesné určení polohy registrační značky

Každý kandidát z předchozí fáze je vyhodnocen pomocí matematické morfologie. Nejprve se provede vertikální projekce horizontálního gradientu, podobně jako byla zjišťována v předchozím kroku horizontální projekce vertikálního gradientu. Dále byla použita matematická morfologie, která slouží jako filtr k odstranění nežádoucího šumu a extrakci příznaků. Základními morfologickými operacemi je eroze a dilatace, které jsou podrobně popsány v sekci 4.4.

Předpokládáme-li, že vertikální projekce horizontálního gradientu je označena jako A. Jednodimenzionální prvek B použijeme k dilataci A. Výsledkem je následně prvek C. Poté byl použit prvek D k erozi prvku C a výsledkem je E.

Pro nalezení pravého a levého okraje registrační značky z kandidátních oblastí, je potřeba po matematické morfologii a vertikální projekci převést obraz do binární podoby pomocí prahování. Aritmetický výpočet:

$$f_T(1, i) = \begin{cases} 1 & f_T(1, i) \geq T \\ 0 & f_T(1, i) < T, \end{cases} \quad (3.3)$$

kde  $f_T(1, i)$  je vertikální projekce po matematické morfologii,  $T$  je práh. Prah je vypočítán pomocí rovnice 3.4, kde  $aver$  je průměrná hodnota  $f_T(l, i)$  a  $t$  je parametr váhy. V tomto algoritmu je určena hodnota  $t = 1.23$ .

$$T = t * aver \quad (3.4)$$

Nejprve jsou odstraněny oblasti s malou šířkou, poté oblasti s příliš malým nebo velkým poměrem šířky a výšky. V oblasti, která nám zůstane, se nachází registrační značka. Pokud by zbylo příliš mnoho oblastí registrační značky, měla by být aplikována přísnější kontrola na přesné určení registrační značky.

### 3.1.4 Metoda detekce registrační značky založená na barevném obrazu

Tato metoda je založena na optimalizovaném algoritmu template matching a analýze barev, která byla publikována Ashtarim a kol. [1]. Algoritmus vyhledává objekty v obraze pomocí šablony objektu nebo popisu. Použitím statické šablony se snižuje výkon a flexibilita, neboť není schopný vyhledat obraz libovolných velikostí.

Vhodnou šablonou pro všechny registrační značky vozidel je modrý obdélník, který lze pozorovat na většině Íránských vozidlech i automobilech v Evropě. Registrační značka bude snadno vyhledaná díky nalezení modrého obdélníku ve standardním poměru s použitím geometrického popisu jako šablony.

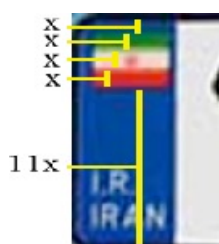
V této metodě není nutné převádět barevný obraz do odstínu šedi nebo binárního. Barvy jsou využity k detekci pozice registrační značky. V tomto postupu je však nutné převést obraz do osmi základních barev nebo do jiného barevného spektra. Například do barevného modelu HSV, který poskytuje lepší výsledky ve srovnání s jinými spektry. Šablona je vyhledávána periodicky shora dolů přes snímek. Na obrázku 3.8 je znázorněn způsob svislého vyhledávání u Íránské značky.

Systém detekuje kandidáta, když index při prohledávání narazí na modré pixely, následně zelené, bílé a červené stejné velikosti, v tomto případě se jedná o íránskou vlajku a následuje modrý jedenáctkrát delší pruh než předchozí. Toto místo je následně považováno za oblast, která je součástí registrační značky. Délka registrační značky je známá a šířka je dopočítána díky známému poměru hran značky.





Obrázek 3.8: Vertikální vyhledávání značky. Převzato z [1].



Obrázek 3.9: Poměr pixelů v modrém obdelníku Íránské registrační značky. [1].

## 3.2 Existující metody klasifikace znaků

### 3.2.1 Porovnání s předem definovanou šablonou

Tato a následující metoda je publikována v článku Petra Ciky a spol.[3]. Předpokladem této metody jsou oddělené znaky registrační značky. Předdefinovaná šablona se skládá z obrazů všech možných znaků a čísel, které se mohou vyskytovat na registrační značce. Předdefinovaný obraz musí mít stejnou velikost s porovnávaným obrazem. Obraz šablony je normalizován na předem danou velikost. Extrahovaný znak je převeden na stejnou velikost jako šablona. Poté následuje porovnávání. Podobnost symbolů je určena pomocí normalizované křížové korelace ( $NCC$ ). Křížová korelace je dána vztahem 3.5, kde  $I$  a  $J$  definují velikost obrázku znaku,  $O$  a  $E$  definují originální a extrahovaný symbol.

$$NCC = \frac{\sum_{i=0}^{I-1} \sum_{j=0}^{J-1} O_{i,j} E_{i,j}}{\sum_{i=0}^{I-1} \sum_{j=0}^{J-1} O_{i,j}^2} \quad (3.5)$$

Díky  $NCC$  je nejpodobnější symbol vybrán z předdefinované databáze.

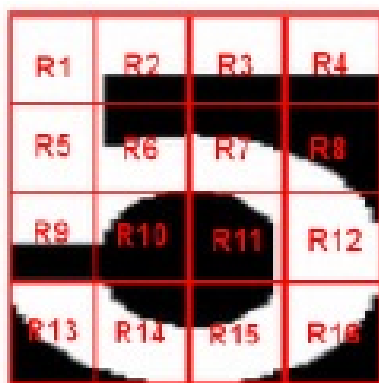


Obrázek 3.10: Předdefinovaný normalizovaný model. Převzato z [3].

### 3.2.2 Metoda založená na analýze oblasti znaku

Analýza je založena na rozdělení běžného znaku do 16 oblastí viz. 3.11. Každá oblast by měla být ve stejné poloze s oblastí šablony. Tento postup není efektivní, proto je zvolena technika, která pro své fungování používá charakteristické vektory. Pro každou šablonu byl vypočten charakteristický vektor bílých pixelů, který byl následně uložen do databáze. Tuto databázi bílých pixelů následně používáme pro porovnání analyzovaných symbolů a symbolů z databáze. Před analýzou musí být převedeny všechny symboly na stejnou velikost. Po výpočtu vektoru bílých pixelů následuje výpočet vektoru vzdálenosti pro každou šablonu z databáze, který se vypočítá podle rovnice 3.6, kde  $D$  je vektor vzdálenosti,  $N$  je počet oblastí,  $V_{P_i}$  je počet šablon v oblasti a  $V_{A_i}$  je počet bílých pixelů v oblasti analyzovaného symbolu. Podobnost symbolů může být určena jako minimální vzdálenost vektoru.

$$D = \sqrt{\sum_{i=0}^N (V_{P_i} - V_{A_i})} \quad (3.6)$$



Obrázek 3.11: Oblasti symbolů. Převzato z [3].

### 3.2.3 Algoritmus založený na neuronových sítích

Metoda, kterou publikovali Nijhuis a kol. [11], je založená na neuronových sítích, V této metodě je použit jako vstup binární obraz. Pro rozpoznání znaků byly použity dva různé typy neuronových sítí. Klasifikace se provádí na základě příznaků. Pomocí neuronové sítě DTCNN jsou vygenerovány čtyři druhy příznaků: horizontální projekce, vertikální projekce, počet vertikálně propojených komponent a počet horizontálně propojených komponent.

Pro tyto příznaky byly vytvořeny speciální neuronové sítě DTCNN. Vstupem pro jednotlivé neuronové sítě byly binární obrazy s výškou každého znaku 15 pixelů a tomu úměrnou šířkou. Každý z těchto příznaků byl následně transformován do pěti vstupů MLP neuronové sítě, což vede k celkem 24 vstupům. Znak byl rozpoznán pouze v případě, že výstup z odpovídajícího výstupního neuronu přesahoval hodnotu 0,85 a ostatní výstupní neurony měly výstupní hodnotu nižší než 0,25.

## Kapitola 4

# Zpracování obrazu

Při zpracování této kapitoly jsem vycházela z informací uvedených v knihách [7], [14] a [4]. Obraz je definován funkcí  $f(x, y)$ , kde  $x$  a  $y$  jsou prostorové souřadnice bodu. Amplituda  $f$ , jenž se nachází v bodě  $(x, y)$ , určuje intenzitu jasu. Pokud jsou souřadnice  $(x, y)$  s hodnotami intenzity jasu  $f$  konečné, tzv. diskrétní veličiny, můžeme obraz označit za digitální. Digitální obraz se skládá z konečného počtu bodů, z nichž každá má určité místo a hodnotu. Tyto obrazové elementy nazýváme pixely.

Předzpracování obrazu jsou operace na nejnižší úrovni abstrakce. Takové operace jsou velmi užitečné pro další zpracování, neboť pomáhají potlačit nežádoucí deformace obrazu jako je hluk a naopak vylepšuje obrazové funkce. Mezi takové operace můžeme zařadit jasovou geometrickou transformaci, filtraci nebo využití hranových detektorů k detekci hran.

Segmentace je jedna z nejdůležitějších a nejsložitějších fází zpracování obrazu. V této části analyzujeme data, jejichž cílem je rozdělení obrazu na segmenty, které nás zajímají a ostatní oblasti, jimiž se nebudeme zabývat. V segmentaci můžeme zkoumat objekty obsažené v obraze, jenž jsou potřebné k rozpoznávání nebo jiné analýze.

### 4.1 Vyhlazování obrazu

Digitální obraz získáváme skenováním, fotografováním nebo natáčením videa. Takto získaný materiál může být ovlivněn nežádoucími jevy jako je zrnění nebo artefakty. Tyto jevy je možné potlačit pomocí vyhlazování obrazu, které jsou přiblíženy v knihách [14] a [4]. Cílem vyhlazování obrazu je potlačení obrazového šumu a jeho výkyvů. Vyhlazovací metody využívají úzkého okolí pixelu, z něhož se vypočítá nová hodnota jasu výstupního obrazu. Nevýhodou tohoto způsobu opravy je rozmazání tenkých hran v obraze, které mohou nést důležité informace pro další zpracování dat. Existují však i postupy vyhlazování, které umí tyto hrany zachovat. Tyto metody vypočítávají průměr pouze podobných míst v sousedství se zpracovávaným bodem.

#### Vyhlazování průměrováním

Vyhlazování průměrováním je jedním z nejsnadnějších způsobů vyhlazování pomocí konvoluce s použitím konvoluční masky. Výsledkem konvoluce je průměr hodnot jasu sousedních bodů v obraze.

## Gaussovo vyhlazování

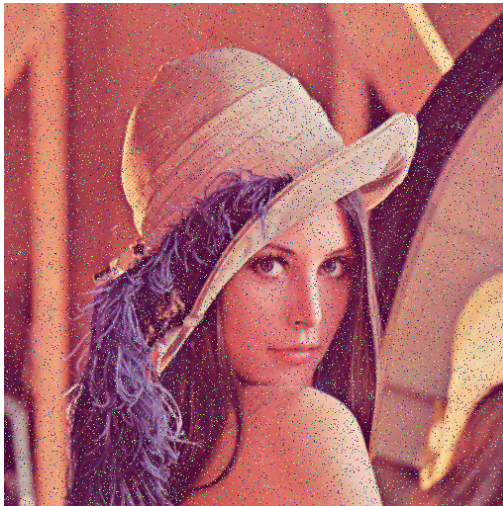
Gaussovo vyhlazování je metoda, při které koeficienty, které se blíží ke středu masky, mají vyšší váhu a odpovídají hodnotám na Gaussově křivce. Dvourozměrné Gaussovo rozdělení hustoty pravděpodobnosti  $h(x, y)$  se střední hodnotou  $m = (0, 0)$  udává vztah

$$h(x, y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (4.1)$$

kde  $\sigma^2$  je rozptyl. Parametr  $\sigma$  určuje strmost Gaussovy funkce.

## Mediánová filtrace

Mediánová filtrace je vhodná k potlačování šumu typu „pepř a sůl“. Je to nelineární filtrace obrazu, jejíž fungování je založeno na výběru mediánu z hodnot nacházejících se na obraze, pomocí posouvání hypotetické masky. Medián je prvek, který se nalézá ve středu uspořádané posloupnosti hodnot. Pokud je počet prvků  $n$  lichý, jedná se o prvek  $f_{med} = f_{(n+1)/2}$ . Když je počet prvků sudý, je medián aritmetický průměr obou hodnot  $f_{med} = (f_{n/2} + f_{(n+1)/2})/2$ . Seřazení prvků  $N$  má časovou složitost  $O(N^2)$  při  $O(N \log(N))$ , dle použité třídící metody.



Obrázek 4.1: Obraz se šumem typu „pepř a sůl“



Obrázek 4.2: Výsledný obraz mediánové filtrace

## 4.2 Prahování

Prahování je nejjednodušší segmentační proces nenáročný na výpočet [14]. Jedná se o nejstarší segmentační metodu, která se stále používá. Prahování slouží k převedení šedotónového obrazu na černobílý obraz. Práh je určitá hodnota podle níž se stanovuje výsledná hodnota bodu. Pokud je původní hodnota jasu vyšší než zvolená hodnota prahu  $T$  tj.  $f(x, y) > T$ , nazýváme jej bodem objektu. Ostatní body, které nesplňují tuto podmínku jsou považovány za pozadí. Segmentovaný obraz  $g(x, y)$  je dán vztahem:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (4.2)$$

Rozlišujeme globální a lokální prahování. **Globální prahování** má určitou hodnotu prahu  $T$ , která je určena pro celý obraz. Pro hodnotu jasu v obraze, pro niž platí  $f(x, y) > T$ , je přiřazena 1, jinak 0. V případě **lokálního prahování** se hodnota prahu  $T$  mění, proto je v různých oblastech obrazu použita rozdílná hodnota prahu.

### Metoda automatického nalezení prahu

Automatické nalezení prahu je možné v případech, kdy jsou na histogramu obrazu dva viditelné vrcholy, které rozlišují pozadí a popředí obrazu. Tato metoda je založena na postupném hledání prahu  $T$ , který je určen průměrem středních hodnot obou vrcholů.

Algoritmus pro automatické nalezení prahu:

1. Výběr počátečního prahu  $T$ .
2. Rozdělení obrazu na základě hodnoty prahu  $T$  na dvě skupiny obrazových bodů:  
 $G_1$  - pixely, jejichž hodnoty intenzity jsou větší než hodnota  $T$ , odpovídající popředí  
a  
 $G_2$  - body pozadí s hodnotami intenzity jasu  $\leq T$ .
3. Výpočet průměru hodnot intenzit  $m_1$  a  $m_2$  pro pixely z množin  $G_1$  a  $G_2$ .
4. Výpočet nové hodnoty prahu.

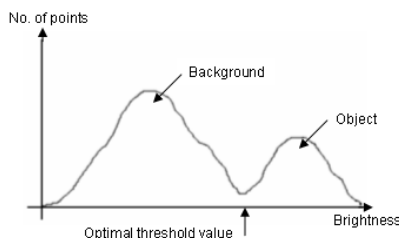
$$T_{new} = \frac{1}{2}(m_1 + m_2) \quad (4.3)$$

5. Opakování kroků 2 až 4 do doby, než rozdíl mezi hodnotami  $|T - T_{new}|$  je nižší než předdefinovaný parametr  $\Delta T$ .

Počáteční hodnota prahu  $T$  musí být vyšší než minimální a nižší než maximální hodnota intenzity jasu v daném obraze. Parametr  $\Delta T$  slouží k řízení počtu iterací. Platí, že čím nižší hodnota  $\Delta T$  je, tím více iterací algoritmus bude muset provést.

## Nalezení optimálního prahu - Otsuova metoda

Otsuova metoda dokáže nalézt optimální práh zcela automaticky. Histogram, který je považován za funkci hustoty pravděpodobnosti, jehož křivka je rozdělena na dvě skupiny vrcholů. Jedna z křivek reprezentuje popředí a druhá pozadí. Optimální práh by měl být takový, aby rozdělení skupin byla co nejdál od sebe viz obrázek 4.3<sup>1</sup>.



Obrázek 4.3: Nalezení optimálního prahu Otsuovou metodou.

## 4.3 Detekce hran

Detektory hran jsou důležité metody předzpracování obrazu, které slouží k vyhledávání náhlých změn jasových funkcí. Popis detekce hran a hranových detektorů byl publikován v knihách [14] a [8]. Hraný v obraze jsou významné, neboť nesou mnoho informací, které můžeme potřebovat pro další zpracování. Vznikají kvůli nespojitostem, odrazivosti povrchu nebo osvětlení. Místa s výraznou změnou jasových hodnot, vlastnostmi obrazových elementů a jeho okolí nazýváme hranami. Jedná se o proměnný vektor se dvěma složkami, velikost a směr. Změny obrazové funkce lze popsat gradientem  $\nabla$ , který udává směr největšího růstu funkce například od černé  $f(x, y) = 0$  k bílé  $f(x, y) = 255$  a velikost, což je reprezentováno mírou strmosti růstu funkce. Hranovým bodem myslíme bod, který má větší velikost gradientu. Ostatní body nepovažujeme za hranové.

Při předpokladu obrazové funkce  $f(x, y)$  je velikost gradientu dána vztahem:

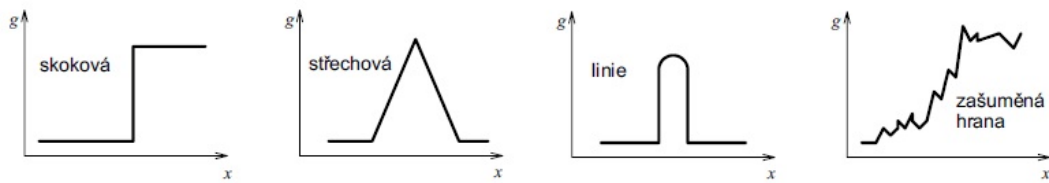
$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (4.4)$$

a směr gradientu:

$$\psi = \text{arg}\left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right) \quad (4.5)$$

Hranice objektů v obraze je možno nalézt lokálními operátory. Pixely s vysokou hodnotou gradientu jsou body hranice v případě, že objekt je oblastí se stejnou hodnotou jasu. Hranice vznikají spojením hranových pixelů a směr hrany je často definován normálou ke směru gradientu. Hraný lze dělit dle jasového profilu ve směru gradientu v daném pixelu. Na obrázku 4.4 jsou čtyři příklady hran, z nichž první tři jsou idealizované a poslední odpovídá zašumělým hranám, které se objevují v reálných obrazech.

<sup>1</sup>Obrázek převzatý z webové stránky: <http://www.softscients.web.id/2014/09/aplikasi-black-white-dengan-metode-otsu.html>



Obrázek 4.4: Jasové profily nejběžnějších hran. [19]

Rozlišujeme tři typy gradientních operátorů:

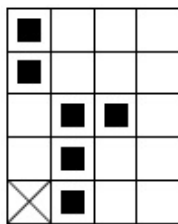
1. ***Operátory využívající aproximace první derivace obrazové funkce pomocí diferencí realizovaných diskrétní konvolucí.***  
 Některé z nich jsou rotačně neměnné např. Laplaceův operátor. Výpočet je proveden pouze z jedné konvoluční masky. Ostatní operátory této kategorie je možno vyjádřit maskami pro konvoluci nebo konvolučními jádry s různými orientacemi. U směrových operátorů je tolik masek, kolik směrů operátor rozlišuje. Hlavní nevýhodou je jejich závislost na šumu a velikosti objektů v obraze. Z masek s odlišnými orientacemi je vybrána ta, která nejlépe aproximuje obrazovou funkci a díky ní je určen i směr gradientu. Do této kategorie řadíme operátory Robertsova, Laplacea, Prewittové, Sobelova, Robinsonova a Kirschova.
2. ***Operátory hledající hrany v místech, kde druhá derivace obrazové funkce prochází nulou.***  
 Velikost masky jednotlivých operátorů musí odpovídat velikosti detailů v obraze. V praxi je jednodušší a spolehlivější hledání průchodů nulou druhé derivace než plochá maxima derivace první. Operátor Marrův a Hildrethové a Cannyho hranový detektor je příkladem takto fungujících operátorů.
3. ***Operátory snažící se lokálně aproximovat obrazovou funkci parametrickým modelem například polynomem dvou proměnných.***



## 4.4 Morfologické operace

Morfologické operace jsou publikované v knize [8]. Matematická morfologie vychází z vlastností bodových množin. Pomocí množin libovolné dimenze můžeme modelovat reálné obrazy. Matematický popis objektů v rovině je popsán v Euklidovském dvojrozměrném prostoru  $\varepsilon^2$  společně se systémem všech podmnožin.

Binární matematická morfologie je založena na množině dvojic celých čísel. Pixely reprezentují dvě celá čísla, která se vztahují ke dvojici souřadných os. Binární obraz je dvojrozměrná bodová množina, ve které pixely objektů v daném obraze zastupují množinu  $X$ . Tyto obrazové body jsou označeny číslem jedna. Pixely doplňku  $X^c$ , jež reprezentují pozadí, jsou zastoupeny číslicí 0. Počátek, označený křížkem, vychází ze souřadnic  $(0, 0)$ .



Obrázek 4.5: Bodová množina [8]

**Morfologická transformace**  $\Psi$  je dána relací mezi obrazem (bodová množina  $X$ ) s jinou, typicky menší bodovou množinou  $B$ , která se nazývá strukturní element. Tento strukturní element se vztahuje k lokálnímu počátku, s názvem reprezentativní bod. Aplikace morfologické transformace pracuje na principu systematického posouvání elementu  $B$  po celém obraze. Výsledek transformace se následně zapíše do výsledného obrazu v reprezentativním pixelu.

### 4.4.1 Dilatace

Dilatace  $\oplus$  sčítá body dvou množin pomocí vektorového součtu. Dilatace  $X \oplus B$  je množinou bodů všech vektorových součtů pro páry pixelů, vždy jeden z množiny  $X$  a jeden z množiny  $B$ .

$$X \oplus B = \{p \in \varepsilon : p = x + b, x \in X, b \in B\} \quad (4.6)$$

Dilatace zvětšuje objekty. Používá se k opravě malých mezer v obraze. Chceme-li zachovat původní rozměr objektů, můžeme zkombinovat dilataci s erozí.

### 4.4.2 Eroze

Eroze  $\ominus$  je duální operací k dilataci. Eroze skládá dvě množiny podle předpisu:

$$X \ominus B = \{p \in \varepsilon : p + b \in X \text{ pro každé } b \in B\} \quad (4.7)$$

Každý bod obrazu  $p$  je ověřen, zda se nachází výsledek  $p + b$  v množině  $X$ . Pokud výsledek leží v této množině, je zapsán v reprezentativním bodě do výstupního obrazu pod číslem

jedna. V opačném případě nula. Eroze je založena na systematickém posouvání strukturního elementu  $B$  po obraze  $X$ . V případě, že bod  $B$  je posunutý o vektor  $p$  obsažen v obraze  $X$ , potom bod odpovídající reprezentativnímu bodu  $B$ , patří do eroze  $X \ominus B$ .

Eroze se používá ke zjednodušení struktury objektů. Objekty s tloušťkou jedna se ztratí a tím se složitější objekty rozdělí na několik jednodušších objektů.

Morfologickými operacemi lze snadno nalézt obrys objektů pomocí odečtení erodovaného obrazu od původního.

***example example example***

(a) Originální obraz

(b) Dilatace

(c) Eroze

Obrázek 4.6: Ukázka eroze a dilatace

## Kapitola 5

# Použité metody pro zpracování obrazu

### 5.1 Odečítání pozadí

Metoda odečítání pozadí je publikována v článku autorů Cheung a Kamath [2].

Identifikace pohybujících předmětů z videa je základním úkolem pro sledování a analýzu provozu, detekce lidí a jejich sledování nebo rozpoznávání gest apod. K identifikaci pohybujících objektů je používáno odečítání pozadí. Každý snímek videa se porovná s modelem pozadí. Pixely aktuálního snímku, které se významně odchyľují od pozadí, jsou považovány za pohybující se objekty.

Algoritmus odčítání pozadí má čtyři hlavní kroky: předzpracování, modelování pozadí, detekce popředí a validace dat.

#### 5.1.1 Předzpracování

Předzpracování se skládá z jednoduchých úloh zpracování obrazu, které mění vstupní video do formátu, se kterým systém pracuje. Nejprve je potřeba vyhlazování, díky kterému odstraníme šum ve videosnímčích. Zjednodušením může být i převod barevného obrazu na šedo tónový.

#### 5.1.2 Modelování pozadí

Modelování pozadí používá nový snímek videa k výpočtu a aktualizaci modelu pozadí. Takový model pozadí je statický popis celého pozadí scény.

#### Nerekurzivní techniky

Tyto techniky využívají metodu klouzavého okna pro odhad pozadí. Uloží do vyrovnávací paměti předchozích  $N$  snímků a odhaduje pozadí na základě časové změny každého pixelu ve vyrovnávací paměti. Nerekurzivní techniky jsou přizpůsobivé, neboť nepotřebují dlouhou historii snímků. Naopak nevýhodou může být velikost bufferu a určení jeho velikosti. Pomalu pohybující se objekty potřebují větší buffer než objekty s vyšší rychlostí. Vzhledem k pevné velikosti vyrovnávací paměti lze tento problém částečně vyřešit ukládáním snímků v nižší snímkové frekvenci.

## Rekurzivní techniky

Rekurzivní techniky nevyužívají buffer pro výpočet pozadí, ale rekurzivně aktualizují jediný model pozadí, který je založen na každém vstupním snímku. Starší vstupní snímky mohou mít vliv na stávající model pozadí. Ve srovnání s nerekurzivní technikou vyžadují méně ukládání, ale jakákoliv chyba v modelu pozadí může přetrvávat po delší dobu. Pro snížení vlivu starších modelů většina metod počítá s exponenciálním růstem.

### 5.1.3 Mixture of Gaussians (MoG)

Rozšířená metoda s modelem pozadí, které je tvořeno pomocí Gaussova rozložení. Tato metoda využívá více Gaussových křivek současně. MOG využívá funkci hustoty pro každý pixel. MOG je parametrická a je možné ji přizpůsobovat bez velké vyrovnávací paměti. Rozdělení pixelu  $f(I_t = u)$  je modelováno směsí  $K$  Gaussových křivek podle rovnice 5.1, kde  $\eta(u; \mu_{i,t}, \sigma_{i,t})$  je  $i$ -tá Gaussova křivka se střední hodnotou intenzity  $\mu_{i,t}$  a směrodatnou odchylkou  $\sigma_{i,t}$  a  $\omega_{i,t}$  je váha z  $i$ -té křivky.

$$f(I_t = u) = \sum_{i=1}^K \omega_{i,t} \cdot \eta(u; \mu_{i,t}, \sigma_{i,t}) \quad (5.1)$$

Typicky se hodnota  $K$  pohybuje v rozmezí od tří do pěti v závislosti na dostupné paměti. Pro každý vstupní pixel  $I_t$  je nejprve potřeba identifikovat křivku  $i$ , jejíž průměr je nejbližší k  $I_t$ . Gaussova křivka  $i$  označuje pasující křivku, pokud platí  $|I_t - \mu_{i,t-1}| \leq D \cdot \sigma_{i,t-1}$ , kde  $D$  označuje malou odchylku prahové hodnoty. Parametry pasující křivky jsou aktualizovány následovně:

$$\begin{aligned} \omega_{i,t} &= (1 - \alpha)\omega_{i,t-1} + \alpha \\ \mu_{i,t} &= (1 - \rho)\mu_{i,t-1} + \rho I_t \\ \sigma_{i,t}^2 &= (1 - \rho)\sigma_{i,t-1}^2 + \rho(I_t - \mu_{i,t})^2, \end{aligned} \quad (5.2)$$

kde  $\alpha$  je uživatelem definovaná míra učení s hodnotou  $0 \leq \alpha \leq 1$  a  $\rho$  je míra učení pro parametry, která může být vyjádřena následovně:

$$\rho \approx \frac{\alpha}{\omega_{i,t}} \quad (5.3)$$

Pokud nelze najít žádnou křivku s nejnižší vahou, je nahrazena novou s průměrem  $I_t$ , velkým počátečním rozptylem  $\sigma_0$  a malou vahou  $\omega_0$ . Ostatní křivky budou mít stejný průměr a rozptyl, ale váha se sníží podle vzorce 5.4.

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} \quad (5.4)$$

Nakonec jsou všechny váhy normalizovány, aby součet těchto vah byl jedna. Ke zjištění, zda je pixel v popředí, je potřeba všechny křivky ohodnotit podle poměru váhy a rozptylu. Vysoce hodnocené křivky mají nižší rozptyl a delší dobu výskytu, které typicky charakterizuje pozadí. Pokud  $i_1, i_2, \dots, i_K$  je pořadí křivek po seřazení. Prvních  $M$  křivek, které splňují následující kritérium, jsou křivkami pozadí:

$$\sum_{k=i_1}^{i_M} \omega_{k,t} \geq \Gamma, \quad (5.5)$$

kde  $\Gamma$  je hodnota prahu.  $I_t$  je pixel popředí, pokud se  $I_t$  liší během  $D$  snímků od některé z křivek pozadí.

## Detekce popředí

V této části se porovnává vstupní snímek videa s modelem pozadí. Pixely odlišné od modelu pozadí jsou označovány jako možné popředí. Nejčastější metoda používaná pro detekci popředí je pomocí prahové hodnoty  $T$ :

$$|I_t(x, y) - B_t(x, y)| > T \quad (5.6)$$

Další známá detekce popředí je založená na normalizovaném prahu:

$$\frac{|I_t(x, y) - B_t(x, y) - \mu_d|}{\sigma_d} > T_s, \quad (5.7)$$

kde  $\mu_d$  je střední hodnota a  $\sigma_s$  směrodatná odchylka. Většina systémů určí práh popředí  $T$  nebo  $T_s$  experimentálně.

## Validace dat

Validace dat je proces zlepšování získaných výsledků z předchozích kroků. Všechny metody využívající pozadí mají drobné omezení. Nezhledňují sousední pixely, ale pracují pouze s jedním. Rychlost adaptace nemusí odpovídat rychlosti pohybujících objektů v popředí. Mohou být detekovány objekty jako pohybující se listy nebo stíny objektů, které jsou detekovány jako pohybující objekty.

## 5.2 Kaskádový klasifikátor

Kaskádový klasifikátor byl popsán v článku [12]. Tento detektor je určený k vyhledávání objektů, který pracuje s obrazy v odstínech šedi. Detektor se skládá ze tří částí: integrální obraz, Haarovy vlnky a AdaBoost. Jednou z předností tohoto detektoru je především rychlost a spolehlivost. U tohoto detektoru také příliš nezáleží na osvětlení a velikosti objektu, které sledujeme.

### 5.2.1 Integrální obraz

Integrální obraz je určen pro rychlé a účinné vypočítání hodnot příznaků ze vstupního obrazu. Nejdříve je obraz převeden do reprezentace integrálního obrazu, ve kterém každý bod odpovídá součtu hodnot předchozích bodů podle rovnice 5.8.

$$\begin{aligned} s(x, y) &= s(x, y - 1) + I(x, y), \\ I_{int}(x, y) &= I_{int}(x - 1, y) + s(x, y), \end{aligned} \quad (5.8)$$

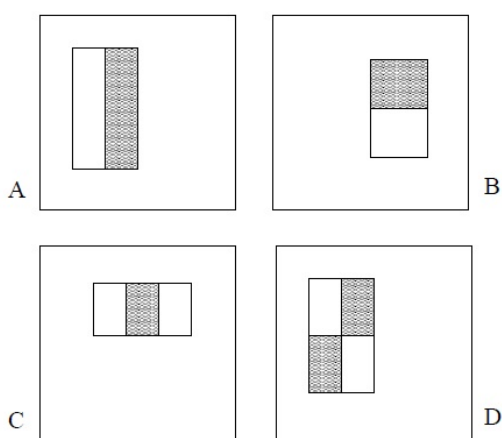
kde  $s(x, y)$  je součet všech hodnot v řádku obrazu. Dále  $I(x, y)$  reprezentuje hodnoty intenzity jednotlivých pixelů vstupního obrazu a  $I_{int}(x, y)$  jsou jednotlivé hodnoty integrálního obrazu. Pak platí, že:

$$\begin{aligned} s(x, 0) &= 0, \\ I_{int}(0, y) &= 0. \end{aligned} \quad (5.9)$$

Tímto postupem se zdatelně zjednoduší výpočet hodnot jednotlivých příznaků.

## 5.2.2 Haarovy vlnky

Detektor Viola-Jones je založen na získávání jednoduchých příznaků s malými výpočetními nároky. Tyto příznaky jsou založeny na podobné definici jako Haarovy vlnky na obrázku 5.1. Hodnota tohoto příznaku je vypočítána jako skupina pixelů obrazu, která odpovídá světlé části, od níž je odečtena skupina pixelů odpovídající tmavé části. Haarovy vlnky mohou být tvořeny dvěma až čtyřmi obdélníkovými oblastmi. Každý příznak je použit na celý vstupní obraz, ve stejný moment dochází ke změně rozměrů příznaků z velikosti  $1 \times 1$  na velikost, která odpovídá vstupnímu obrazu. Pro vstupní obraz o velikosti  $19 \times 19$  je tak získáno přibližně 64 000 hodnot příznaků, jenž slouží jako vstup učícího klasifikačního algoritmu. Tento algoritmus vybere pouze určité množství příznaků spolu se stejným počtem natrénovaných slabých klasifikátorů, díky kterým lze dobře klasifikovat vstupní obraz. Při detekci je poté použito už jen toto malé množství příznaků.



Obrázek 5.1: Obdélníkové příznaky [16]

## 5.2.3 AdaBoost

AdaBoost je název pro klasifikační algoritmus, který je založen na principu strojového učení boosting. Tato metoda zlepšuje klasifikační přesnost jakéhokoliv algoritmu strojového učení. Nejprve jsou vytvořeny klasifikátory, jenž nesou označení „slabí žáci“ neboli „weak learners“. Slabí žáci jsou tvořeni výběrem vzorků ze základní trénovací množiny. První klasifikátor nemá příliš vysokou přesnost, proto jsou postupně přidávány další klasifikátory. Poté je vygenerován soubor klasifikátorů, který je označen jako „silný žák“ neboli „strong learner“. Přesnost klasifikace tohoto souboru je libovolně vysoká podle vzorků v trénovací množině.

Slabé klasifikátory  $h_t(x)$ , jsou vybrány z množiny klasifikátorů  $H$ . Lineární kombinací těchto klasifikátorů vznikne nelineární silný klasifikátor  $H(x)$ . Množina  $S$ , jenž je složena z dvojic  $(x_i, y_i)$ , ve které  $x_i$  označuje hodnotu příznaku a  $y_i$  značí třídu odpovídající příznaku leží pro  $i = 1, \dots, M$ .  $M$  je velikost trénovací množiny. AdaBoost k vážení používá trénovací váhy  $D_t$ , jenž jsou nejdříve nastaveny rovnoměrně. V každé smyčce se provede výběr slabého klasifikátoru, který obsahuje nejmenší chybu klasifikace při daných váhách  $D_t$ . Dále je ověřeno, zda chyba není vyšší než 0,5. Poté následuje výpočet koeficientu slabého klasifikátoru v lineární kombinaci  $H(x)$ . Na závěr jsou aktualizovány jednotlivé váhy  $D_t$  trénovací množiny.

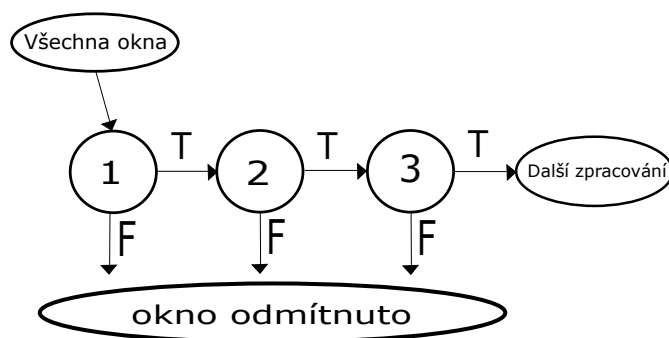
Slabý klasifikátor je vybrán tím způsobem, aby jeho přesnost byla lepší, než při náhodné klasifikaci. Za předpokladu, že chyba  $\epsilon_t$  překročí hodnotu 0,5, poté není splněn základní požadavek a není zaručeno, že algoritmus bude pracovat správně. Tím, že se váhy aktualizují, váha špatně klasifikovaného měření se zvětší a váha dobře klasifikovaných bude zmenšena. V další fázi je vyhledán slabý klasifikátor, jenž lépe klasifikuje špatně provedená měření. Proto AdaBoost snižuje chybu v trénování exponenciálně podle rostoucího počtu klasifikátorů. Při přílišném navýšení počtu klasifikátorů může dojít k přetrénování, což zapříčiní špatné fungování algoritmu. K tomuto jevu však nedochází často.

#### 5.2.4 Detekce objektů v obraze

Pro detekci objektů v obraze je nejdříve potřeba vytvořit trénovací množinu obrazů, které obsahují pozitivní vzory daného objektu a negativní vzory pozadí. Všechny tyto vzory musí mít stejnou velikost  $w$ . Velikost  $w$  by měla mít co nejmenší hodnotu, pokud však má dostatečné optické rozlišení. Vzory slouží k natrénování klasifikačního algoritmu AdaBoost.

Při detekci objektu není zpracováván celý obraz najednou, ale jen po částech. Části jsou vybrány posuvným podoknem, které se posouvá po osách  $x$  a  $y$ . Toto podokno také mění svoji velikost  $s \times w$  podle předpokládané velikosti objektu, který má být detekován. Zde jsou  $x$  a  $y$  počáteční souřadnice podokna,  $w$  značí základní velikost okna, která odpovídá velikosti trénovacích vzorů a  $s$  označuje faktor zvětšení daného podokna. Jako vstup slouží vybrané příznaky daného podokna a klasifikátor určí, zda toto podokno obsahuje nebo neobsahuje sledovaný objekt.

Detekce objektu je výpočetně velmi náročná, protože je obraz prohledáván velkým množstvím podoken. Výpočetní doba se však dá zkrátit snížením průměrné doby, kdy se detektor věnuje prohledávání každého podokna. Tohoto efektu je docíleno zapojením kaskádového klasifikátoru. K vytvoření klasifikátoru, který dokáže vybrat téměř všechny pozitivní případy a zároveň i velké množství negativních případů, je potřeba pouze malé množství příznaků. Jelikož je většina podoken v obraze negativní, snaží se ve stupních kaskády zamítnout co nejvíce těchto negativních podoken. Díky tomuto mohou do dalších stupňů kaskády přestupovat jen podokna, která jsou označena jako pozitivní. Takto vyřadí většinu podoken, jenž obsahují pozadí již při prvních stupních kaskády, a proto není nutné extrahovat a klasifikovat hodnoty příznaků pro další stupně kaskády.



Obrázek 5.2: Schématické znázornění kaskádového klasifikátoru

## 5.3 Support vector machines

V této části bylo čerpáno z diplomové práce Jakuba Sochora [13] a publikace Milana Sonky a kol [14].

Support vector machines (SVM) je klasifikační metoda založená na strojovém učení, která byla vyvinuta v roce 1998 Vladimírem N. Vapnikem. SVM nemusí být využita pouze ke klasifikaci, ale také k regresi. Klasifikace probíhá rozdělením tříd v příznakovém prostoru nadrovinou. Její velikost je definována jako vzdálenost mezi nadrovinou v  $n$ -dimenzionálním prostoru příznaků a nejbližších tréninkových vzorů všech tříd, které se nazývají support vectors. Support vectors stanovují funkci rozlišení.

Platí, že  $S = (x_1, y_1), \dots, (x_l, y_l)$  je konečná množina dvojic, kde  $x_i$  je z oboru  $\mathbb{R}^N$  a  $y_i$  z oboru  $\mathbb{Y}$  pro všechna  $1 < i < l$ , kde  $l$  je počet prvků množiny. Vektory  $x_i$  jsou klasifikované objekty a  $y_i$  je třída objektů. Klasifikátorem je funkce  $f: \mathbb{R}^n \rightarrow \mathbb{Y}$ , která mapuje objekt ke třídě. Platí, že množina  $\mathbb{Y}$  může být libovolná konečná množina. Je vhodné předpokládat, že  $\mathbb{Y}$  je vlastní konečná podmnožina celých čísel.

### 5.3.1 Lineární SVM

Lineární binární support vector machines klasifikuje pouze objekty dvou tříd za předpokladu, že objekty jsou lineárně oddělitelné. Hlavní myšlenkou je klasifikovat objekty pomocí nadroviny. Nadrovina může být použita pro klasifikaci ve formě vektoru  $\omega$  a skalárního prahu  $b$ . Vektor  $x$  je klasifikován jako výsledek  $\omega \cdot x - b$ , kde operace značená tečkou je skalární součin definována následující rovnicí.

$$\omega \cdot x = \omega^T x \quad (5.10)$$

Nadrovina je nazvána optimální nadrovinou tehdy, pokud vektory  $x_i$  jsou odděleny bez chyb podle rovnic a vzdálenost mezi nejbližším vektorem a nadrovinou je maximální. Optimální nadrovina je vyobrazena na obrázku 5.3, kde nejbližší vektory k nadrovině se nazývají support vectors a jsou označeny červeně.

$$\omega \cdot x_i - b \leq 1 \quad \text{if } y_i = 1 \quad (5.11)$$

$$\omega \cdot x_i - b \geq -1 \quad \text{if } y_i = -1 \quad (5.12)$$

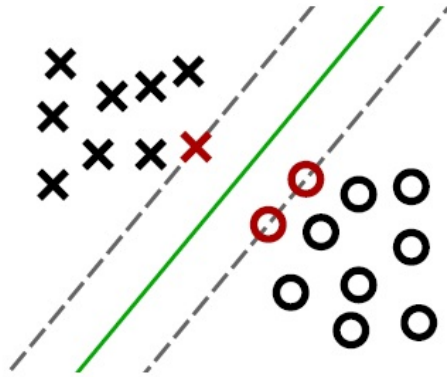
$$y_i \cdot (\omega \cdot x_i - b) \leq 1 \quad (5.13)$$

### 5.3.2 Multitřídní SVM

Existují dva způsoby klasifikace objektů do jedné z více tříd se SVM. Prvním způsobem je vytvoření binárního klasifikátoru SVM pro každou třídu, která bude klasifikovat objekt do třídy  $m$  nebo netřídy  $m$ . Proto je nezbytné spustit binární klasifikátor až  $k$  krát v případě, že se klasifikace provádí do  $k$ -tříd. Druhý přístup využívá pouze jeden SVM klasifikátor, který je schopen zařadit objekt do více tříd.

Předpokládáme-li, že  $l(|S| = l)$  je trénovací vektor a  $k$  klasifikačních tříd ( $|\mathbb{Y}| = k$ ). Iterační proměnné  $i$  a  $j$  pro trénování párů ( $i, j \in 1, \dots, l$ ) a  $m$  iterační proměnná tříd, kde  $m \in 1, \dots, k$ . Součet Lagrangianových násobitelů pro trénovací dvojice  $i$  pro všechny třídy označeny jako  $A_i$  rovnicí 5.14.





Obrázek 5.3: Optimální nadrovina (zelená) a support vectors (červené). Převzato z [13].

$$A_i = \sum_{m=1}^k a_i^m \quad (5.14)$$

Proměnná  $c_i^m$  je rovna 1, pokud trénovací vektor  $x_i$  patří do třídy  $m$  5.15.

$$c_i^m = \begin{cases} 1 & y_i = m \\ 0 & y_i \neq m \end{cases} \quad (5.15)$$

## Kapitola 6

# Návrh aplikace

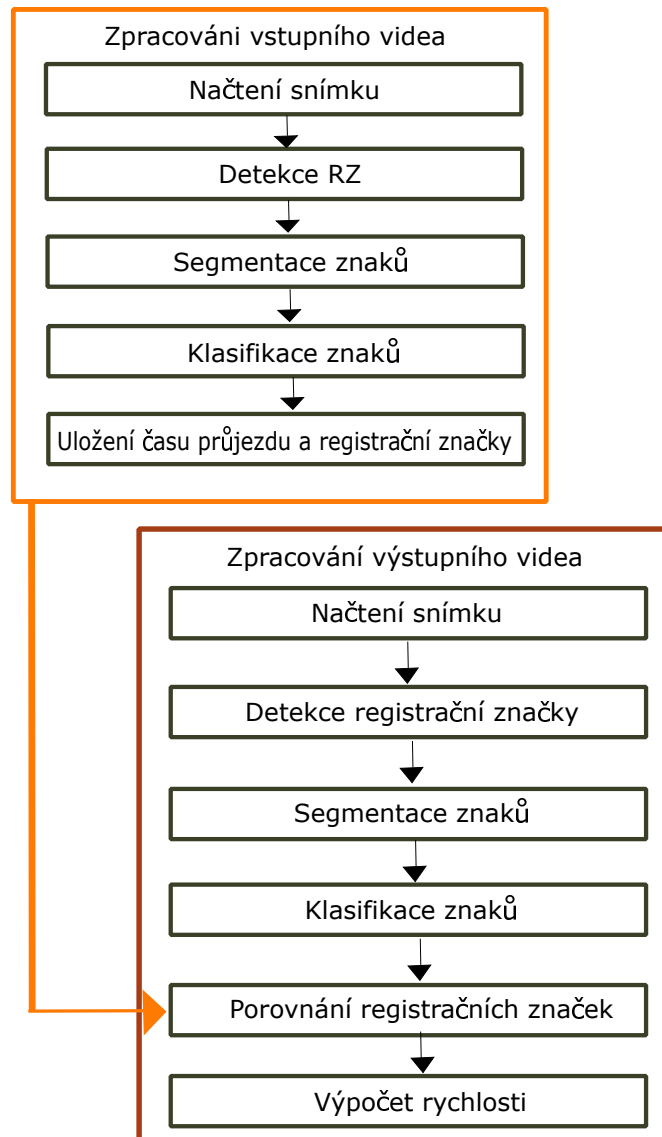
Každý program, jehož úkolem je úsekové měření rychlosti, využívá ke svému fungování dva videozáznamy. V prvním záznamu je zachycen vstup vozidla do měřeného úseku a v druhém výstup vozidla z tohoto úseku. Poté co jsou zpracovány oba záznamy, může program přistoupit ke spárování vozidel, které se v nich vyskytují. Na závěr je již pouze vypočítána průměrná rychlost daných automobilů.

Schéma systému je znázorněno na obrázku 6.1. Aplikace je rozdělena do několika na sebe navazujících částí. V první části dochází k lokalizaci poznávací značky. V následující fázi jsou vyhledány oblasti znaků. Poté je podle počtu nalezených oblastí určeno, zda se jedná o registrační značku. Pokud počet znaků odpovídá registrační značce, může program přistoupit k jejich klasifikaci. Výsledkem rozpoznávání jednotlivých znaků je celé znění registrační značky. Následně jsou tato data uložena, společně s časem zachycení značky, do souboru. Po dokončení zpracovávání prvního videa, může program začít zpracovávat i druhé video, které obsahuje záznam s automobily, jenž opouští měřený úsek. Tento záznam je zpracováván stejným způsobem jako první. Poté jsou porovnány jednotlivé registrační značky, zda se nacházejí v obou videozáznamech. Po spárování vozidel již může aplikace vypočítat průměrnou rychlost automobilu v měřeném úseku.

Program je navržen tak, aby rozeznával pouze České registrační značky, které se skládají ze sedmi znaků. Ve zpracovávaných videích se však vyskytují i slovenské a rakouské registrační značky. Tyto značky, z jiných zemí Evropské unie, obsahují státní znaky, které nejsou programem rozeznány. Systém dále nerozezná nové značky na přání, neboť obsahují osm znaků místo sedmi jako je to u klasických registračních značek.

### 6.1 Detekce registrační značky

Registrační značka je rozeznávána pomocí kaskádového klasifikátoru, jehož popis se nachází v sekci 5.2. Tento klasifikátor vyhledává značky přes celý pořízený snímek. Tento způsob detekce registrační značky se zakládá na pozitivních a negativních trénovacích vzorcích. Pozitivní vzorky měly shodnou velikost  $42 \times 11$  pixelů, na rozdíl od negativních s různými velikostmi. Pro trénování klasifikátoru byl použit algoritmus Gentle AdaBoost, který byl aplikován na příznaky HAAR. Trénování klasifikátoru proběhlo v 16 iteracích. Kaskádový klasifikátor pracuje s velkým množstvím podoken. V každém stupni kaskády jsou postupně zamítána negativní podokna. Do dalšího stupně kaskády pak vstupuje méně dat, které mohou obsahovat hledanou oblast. Oblast, která projde všemi stupni kaskády je prohlášena za registrační značku.



Obrázek 6.1: Dekompozice systému



(a) Příznaky HAAR

(b) Příznaky LBP

Obrázek 6.2: Detekce registrační značky použitím kaskádového klasifikátoru

### 6.1.1 Efektivnější přístup detekce registrační značky

Pro rychlejší a efektivnější vyhledávání registrační značky je nejdříve použita metoda odčítání pozadí, která určí pohybující se objekty. V tomto případě se jedná o automobily, jenž detekovány a následně ohrazeny boxy. Tyto boxy vymezí prostor, v němž může být detekována registrační značka. Poté je tato oblast prohledána kaskádovým algoritmem, jenž určí polohu poznávací značky. Odčítání pozadí může výrazně zrychlit prohledávání, protože kaskádový algoritmus nemusí prohledávat celý obraz.



(a) Příznaky HAAR

(b) Příznaky LBP

Obrázek 6.3: Detekce registrační značky kombinací odečítání pozadí a kaskádového klasifikátoru

## 6.2 Redukce falešných kandidátů registračních značek

Poté co je nalezen kandidátní výřez registrační značky, je potřeba rozlišit, zda se jedná o opravdovou nebo falešnou oblast registrační značky. Při tomto postupu vycházíme z předpokladu, že znak je tvořen spojitým regionem, který je oddělen od ostatních znaků díky bílému pozadí.

Nejprve je obraz adaptivním prahováním převeden na černobílý obraz, na kterém jsou zřetelněji vidět symboly. Tento obraz se nazývá binární. Získané kontury v potencionální registrační značce obalíme obdelníky, ve kterých se nachází všechny spojitě objekty v obraze. V obrazu se nachází mnoho těchto oblastí. Následně jsou odstraněny oblasti, jejichž velikost nemůže odpovídat velikosti znaku. V případě, kdy zbylo sedm nalezených oblastí, jedná se o oblast registrační značky. Tyto oblasti jsou předány k samostatné klasifikaci v pořadí zleva doprava.

## 6.3 Klasifikace znaků

Postupně jsou všechny segmenty znaků předány ke klasifikaci pomocí natrénovaného multitřídního klasifikátoru SVM, který je detailněji popsán v sekci 5.3. Trénování proběhlo na vzorcích o stejné velikosti  $20 \times 40$  pixelů. Obraz znaku je reprezentovaný vektorem příznaků. Do vektoru jsou vloženy hodnoty horizontální i vertikální projekce a hodnoty jasů.

Výsledek klasifikace je třída, která nám určí, o jaký znak se jedná. V této části také probíhá validace znaku. Klasifikátor má totiž potíže rozeznat písmeno B od číslice 8, proto když je klasifikováno číslo 8 na druhé pozici, automaticky se přepíše písmeno B.

# Kapitola 7

## Implementace

### 7.1 Použité nástroje

Pro implementaci byla použita knihovna počítačového vidění **OpenCV** (Open Source Computer Vision) implementována v jazyce C++. Knihovna je multiplatformní se zaměřením na zpracování obrazu v reálném čase. OpenCV poskytuje mnoho algoritmů a funkcí pro zpracování obrazů nebo videí.

Pro výpočet měřené vzdálenosti pomocí GPS souřadnic byla použita knihovna **Geopy**. Tato knihovna umožňuje snadný přístup k informacím a metodám pro určování polohy GPS souřadnicemi.

### 7.2 Trénování kaskádového klasifikátoru

Pro trénování klasifikátoru je potřeba připravit datovou sadu pozitivních a negativních obrazů. Datová sada byla vytvořena z fotografií i natočených videí, ze kterých byla vyřezána jak pozitivní, tak i negativní data. Pozitivní obrazy jsou takové, které obsahují pouze registrační značku bez přebytečného pozadí. Naopak negativní vzorky mohou být jakékoliv fotografie, které neobsahují objekty registrační značky. Pro trénování bylo použito 1503 pozitivních a 1767 negativních dat.

Každá sada je uložena do odlišné složky. Pozitivní obrazy jsou umístěny do adresáře „positives“ a negativní do adresáře „negatives“.



Obrázek 7.1: Ukázka pozitivních vzorků

Trénovací program musí znát obrazy, které jsou připraveny pro jeho použití. Následujícími příkazy získáme soubory, ve kterých jsou uloženy cesty ke vzorkům jednotlivých složek.

```
find ./negatives -iname "*.jpg" >> negatives.dat
find ./positives -iname "*.jpg" >> positives.dat
```

Nyní máme připravené dva soubory s cestami k jednotlivým vzorkům, které použijeme jako vstup skriptu, vytvořeného „Naotoshi Seo“, `createtrainsamples.pl`. Dalšími vstupy



Obrázek 7.2: Ukázka negativních vzorků

skriptu se zadávají rozměry vzorů či počet obrazů v datové sadě. Poté se kombinují pozitivní vzorky s náhodnými negativními vzory a následně je spuštěna funkce knihovny OpenCV `opencv_createsamples`. Výstupem jsou soubory s příponou `.vec`, binárního formátu, který obsahuje obrazy.

```
perl createtrainsamples.pl positives.dat negatives.dat samples 3270
"opencv_createsamples -bgcolor 0 -bgthresh 0 -maxxangle 1.1 -maxyangle 1.1
maxzangle 0,5 -maxidev 40 -w 52 -h 11"
```

Následně je vytvořen soubor kolekci pro soubory `.vec`.

```
find ./samples -name '*.vec' > samples.dat
```

Použitím programu `mergevec` získáme sloučení `.vec` souborů.

```
./mergevec samples.dat samples.vec
```

V metodě trénování kaskády lze použít různé příznaky, jako jsou HAAR, LBP nebo HOG. Pro trénování byl použit algoritmus Gentle AdaBoost. Po skončení trénování je připraven soubor `.xml` (výstup programu), jedná se o klasifikátor, který bude detekovat registrační značky v programu.

```
opencv_traincascade -data LBP13 -vec samples.vec -bg negatives.dat -numStages
13 -minHitRate 0.999 -maxFalseAlarmRate 0.5 -numPos 2000 -numNeg 1767 -w 52 -h
11 -mode ALL -precalcValBufSize 1024 -precalcIdxBufSize 1024 -featureType LBP
```

### 7.3 Trénování SVM klasifikátoru

K rozpoznání znaků je zapotřebí vytvořit klasifikátor, který umožní rozpoznat jednotlivé znaky. K trénování jsou potřeba trénovací vzorky jednotlivých znaků vyříznutých z registračních značek. Znaky jsou rozříděny do tříd podle toho, o jaký symbol se jedná. Dostatečný počet vzorků byl pořízen především na všechny číslice a písmena A a B. Ostatní znaky měly ve svých třídách menší zastoupení.



Obrázek 7.3: Ukázka trénovacích vzorků nejčastějších symbolů

Z jednotlivých tříd byly postupně zpracovávány jednotlivé obrazy, na kterých se vyskytoval konkrétní znak. Nejprve došlo k extrakci vektorových příznaků jednotlivých znaků.

Vektorové příznaky, které byly použity, se skládaly z hodnot horizontální a vertikální projekce a intenzity jasu celého obrazu.

Následně byly tyto trénovací příznaky jednotlivých vzorů ukládány společně s třídou, do které je vzor zařazen. Jakmile byla získána všechna trénovací data, dochází k samotnému učení. Jediným zadaným parametrem je kritérium pro ukončení iterace na hodnotu 0,000001. Výsledkem je natrénovaný klasifikátor SVM použitelný k rozpoznání registrační značky.

## Kapitola 8

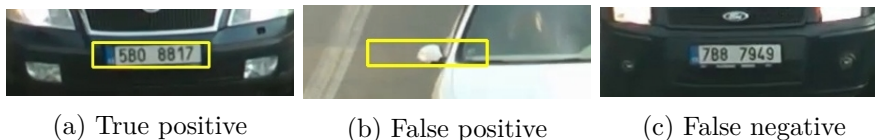
# Vyhodnocení

### 8.1 Vyhodnocení detekce registrační značky

Vyhodnocení registračních značek probíhalo na každém 15. snímku videa. Tyto snímky byly následně ručně anotovány. Anotace probíhala nejdříve označením souřadnic levého horního rohu a následně dolního pravého rohu registrační značky. V těchto anotovaných snímcích byly vyhledány registrační značky a porovnány s anotacemi, zda se jedná o pravou registrační značku nebo ne.

Výsledkem mohly být tři různé možnosti nalezení, které jsou zobrazeny na obrázku 8.1.

- **true positive (TP)** - nalezená oblast je skutečnou registrační značkou
- **false positive (FP)** - falešně nalezená oblast registrační značky
- **false negative (FN)** - oblast registrační značky nebyla nalezena



Obrázek 8.1: Případy detekce registračních značek

Registrační značky jsou detekovány pomocí funkce `detectMultiScale()`, která vrací seznam obdélníků, kde by se měly nacházet detekované registrační značky. Parametrem této funkce je `scaleFactor`, který určuje, jakým násobkem bude detekční okno zvětšováno. V tabulce 8.1 je vyobrazena závislost na tomto parametru.

scaleFactor	1.01	1.05	1.1	1.2	1.3	1.4
TP	1026	1018	1018	999	953	845
FP	2574	586	242	135	148	81
FN	32	40	40	59	105	213

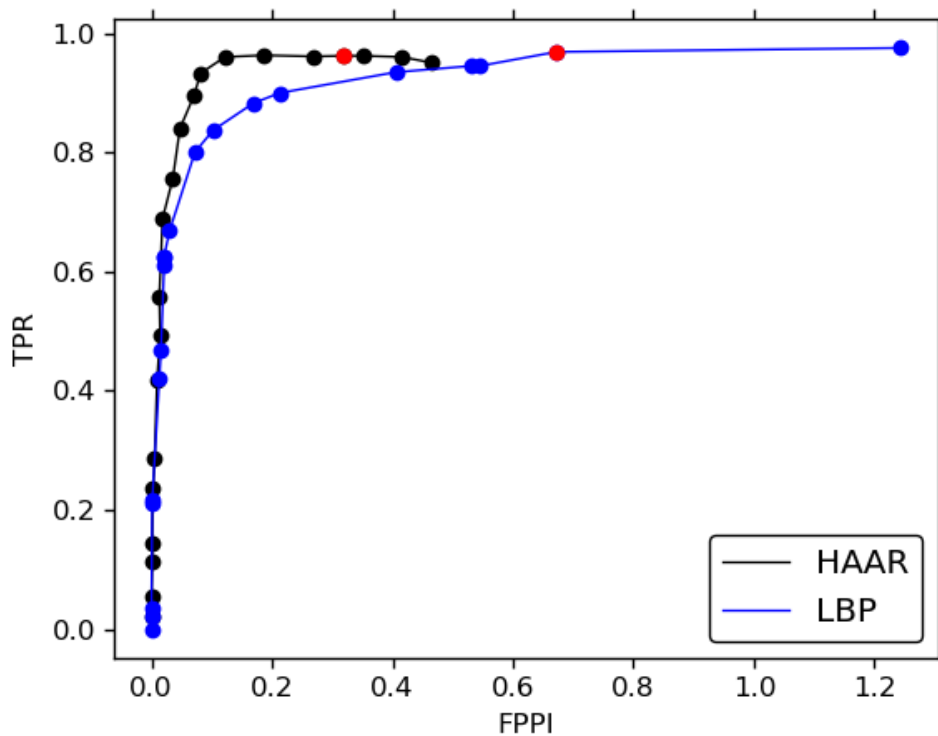
Tabulka 8.1: Závislost detekce registračních značek na parametru `scaleFactor`



Na základě experimentů parametru byla vybraná hodnota 1.1, která je ideálním poměrem mezi počtem správně nalezených a falešně detekovaných registračních značek. Čím je hodnota parametru vyšší, tím je detekováno méně registračních značek. Naopak při zvolení parametru 1.01 by bylo nalezeno více skutečných registračních značek, ale falešných detekcí by přibýlo až o desetinásobek.

ROC křivky na obrázku 8.2 znázorňují TPR(true positive rate) v závislosti na FPPI (false positives per image). TPR je podíl úspěšně nalezených značek ku celkovému počtu značek 8.1. FPPI je průměrný počet falešně nalezených registračních značek na snímku videa.

$$TPR = \frac{\sum TP}{\sum TP + \sum FN} \quad (8.1)$$



Obrázek 8.2: ROC křivka - červeně označené body znázorňují něco

## 8.2 Vyhodnocení klasifikace znaků

Pro vyhodnocení znaku byly vyřezány jednotlivé oblasti znaků z registračních značek. Poté bylo nutné odstranit nečitelné znaky, které nebylo možné dále zpracovat. Po odstranění znaků byly anotovány jednotlivé symboly.

Vyhodnocení probíhalo porovnáním anotovaných dat společně s daty, která byla získána klasifikátorem SVM. Pořízené datové sady byly rozděleny na části podle oblastí, kde byly odebrány. Pro klasifikaci symbolů byly vytvořeny 3 sady pro každý dataset. Při rozdělení obrazu na dvanáctiny je první sada pořízena z oblasti devět a deset. Druhá datová sada označena „11“ je oblast jedenáctá a poslední sada „12“ je nejspodnější oblast obrazu, znázorněno na obrázku 8.1. V tabulce 8.2 je vyobrazena matice záměn pořízena z datasetuA\_11.

Na matici lze pozorovat, že nejhorší výsledky jsou zaznamenány u písmene B, které je ve většině případech klasifikováno jako číslice 8. Písmeno D v této sadě nebylo rozpoznáno vůbec, klasifikátor jej považoval za číslici 0. Ostatní znaky, které nejsou znázorněny v matici záměn, nebyly v sadě vůbec nebo se vyskytovaly pouze jednou.

V tabulce 8.3 je znázorněna procentuální úspěšnost klasifikace znaků pro jednotlivé oblasti. Čím níže je registrační značka vyjmuta, tím jsou znaky čitelnější.

Obsahem tabulky 8.4 je znázornění úspěšnosti klasifikace jednotlivých znaků v odlišných oblastech obrazu.



Obrázek 8.3: Rozdělení obrazu

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	M	U	Z
0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	47	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2
3	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	58	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	40	2	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	73	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	58	0	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	32	0	0	28	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
D	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

Tabulka 8.2: DatasetA\_11 - Matice záměn

	datasetA_8	datasetA_11	datasetA_12
TP	741	543	502
FP	84	44	34
Úspěšnost v %	89,8	92,5	93,7

Tabulka 8.3: Úspěšnost klasifikace znaků

Znak	DatasetA_8			DatasetA_11			DatasetA_12		
	TP	FP	Úspěšnost v %	TP	FP	Úspěšnost v %	TP	FP	Úspěšnost v %
0	51	0	100	44	0	100	36	0	100
1	65	0	100	44	0	100	54	0	100
2	64	3	95,5	47	3	94	37	1	97,4
3	59	9	86,8	39	0	100	33	0	100
4	79	0	100	58	0	100	50	0	100
5	47	2	95,9	40	2	95,2	41	0	100
6	51	0	100	41	0	100	39	0	100
7	46	1	97,9	39	0	100	37	0	100
8	101	0	100	73	1	98,7	54	0	100
9	86	0	100	58	0	100	54	0	100
A	19	0	100	10	0	100	14	0	100
B	36	53	40,5	28	32	46,8	30	24	55,6
C	3	1	75	3	0	100	4	0	100
D	1	1	50	0	3	0	0	4	0

Tabulka 8.4: Úspěšnost klasifikace jednotlivých znaků

### 8.3 Vyhodnocení rozpoznání registrační značky

Registrační značky byly vyhodnocovány v různých částech snímku. Registrační značky jsou uloženy ve třech sadách. První sada je pořízena z třetí čtvrtiny snímku, další je sedmou osminou snímku a poslední sada je spodní částí snímku. V následující tabulce 8.5 je znázorněno, jaké rozdíly jsou mezi rozpoznáním celé registrační značky v závislosti na tom, kde je značka umístěna v jaké části snímku.

V tabulce 8.6 je zapsáno vyhodnocení, které analyzuje počet chybně rozpoznávaných registračních značek.

	DatasetA3_4	DatatsetA7_8	DatasetA8_8
Počet RZ	200	189	451
Detekované RZ	162	172	419
TP	20	55	334
FP	142	117	85
Úspěšnost v %	12,35	31,98	79,71

Tabulka 8.5: Úspěšnost rozpoznání registračních značek v závislosti na umístění značky ve snímku.

	DatasetA3_4		DatatsetA7_8		DatasetA8_8	
počet RZ	162		172		419	
bez chyby	20	12,35%	55	31,98%	334	79,71%
1 chyba	18	11,11%	27	15,7%	55	13,13%
2 chyby	6	3,7%	3	1,74%	16	3,82%
3 chyby	8	4,94%	6	3,49%	4	0,95%
4 chyby	10	6,17%	9	5,23%	3	0,72%
5 chyb	22	13,58%	17	9,88%	5	1,19%
6 chyb	47	29,01%	28	16,28%	1	0,24%
7 chyb	31	19,14%	27	15,7%	1	0,24%

Tabulka 8.6: Počet chyb v rozpoznání registračních značek

## 8.4 Vyhodnocení reidentifikace vozidel

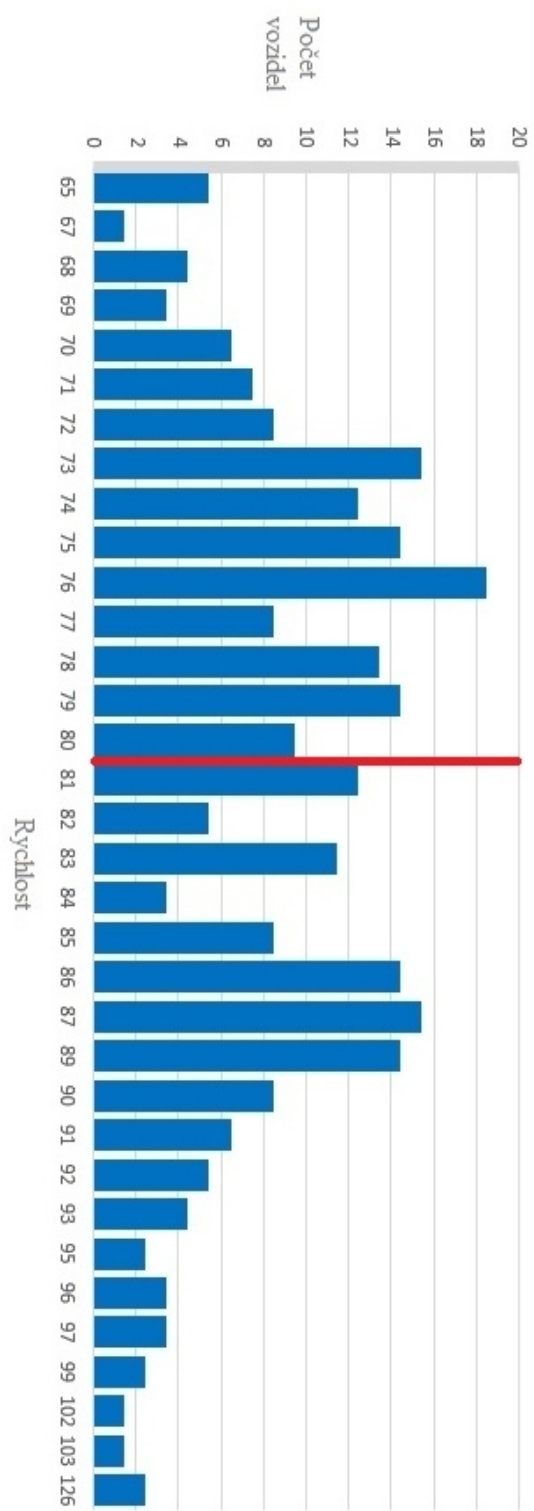
Reidentifikace vozidel je méně úspěšnější než předchozí fáze, neboť je to výsledek celého systému. K vyhodnocení byla použita anotovaná data z obou videí, která byla mezi sebou porovnána. Tímto postupem bylo zjištěno zda se registrační značky vyskytují ve vstupním i výstupním videozáznamu. Tento výsledek je porovnán s výstupem z programu a ten je následně vyobrazený v tabulce 8.7.

	<b>datasetAB</b>	<b>datasetCD</b>
<b>TP</b>	61	38
<b>FN</b>	51	111
<b>Recall</b>	54%	25,5%

Tabulka 8.7: Vyhodnocení reidentifikace

## 8.5 Vyhodnocení rychlosti automobilů

Cílem této práce je vyhodnocení rychlosti vozidel v měřeném úseku. V grafu 8.4 jsou vyobrazeny údaje pořízené programem na více datových sadách. Tyto výsledky byly sjednoceny a vloženy do grafu. V grafu lze vidět, že automobily jely rychlostí od 65 km/hod do 126 km/hod. Červená čára označuje v tomto konkrétním případě maximální povolenou rychlost. Nejvíce automobilů jelo rychlostí 76 km/hod.



Obrázek 8.4: Graf znázorňující rychlosti automobilů

## Kapitola 9

# Závěr

Pro účely detekce registračních značek bylo vytvořeno  $x$  pozitivních a  $x$  negativních dat, pomocí nichž byl natrénován klasifikátor. Pro klasifikaci dat bylo použito pro každou číslici zhruba 400 vzorků. U některých znaků byl počet vzorků nižší, neboť nebylo možné získat plnou datovou sadu pro všechny znaky. V Brně a jeho přilehlém okolí se vyskytují převážně znaky A a B. Některé symboly na registračních značka se však vyskytují zcela ojediněle.

Nejdříve je systémem zpracováno první video se vstupem vozidla do měřeného úseku. Poté je zpracováno i druhé video s vozidly, jež opouštějí danou oblast. Výstupem druhého videozáznamu je naměřená rychlost. V jednotlivých snímcích videa jsou nejprve vyhledány oblasti registračních značek, které jsou kontrolovány, zda se opravdu jedná o registrační značku. Následně dochází ke klasifikaci jednotlivých znaků registrační značky. Výsledek klasifikace je uložen společně s časem průjezdu vozidla v prvním videu. Druhé video je zpracováno stejným způsobem, ale klasifikovaná značka je porovnávána, zda se vyskytla již v předchozím videu. Pokud se jedná o stejné vozidlo, následuje výpočet průměrné rychlosti.

U rozpoznání registračních značek bylo dosaženo nejlepších výsledků ve spodní části snímku. Rozpoznávání v této části snímku dosahovalo úspěšnosti až 80%. Práce by mohla být dále rozšířena o rozpoznání nových registračních značek na přání nebo značek ostatních evropských zemí.

# Literatura

- [1] Ashtari, A. H.; Nordin, M. J.; Kahaki, S. M. M.: A new reliable approach for Persian license plate detection on colour images. In *Electrical Engineering and Informatics*, July 2011, ISSN 2155-6822, s. 1–5.
- [2] Cheung, S.-C. S.; Kamath, C.: Robust Background Subtraction with Foreground Validation for Urban Traffic Video. *EURASIP J. Appl. Signal Process.*, ročník 2005, Leden 2005: s. 2330–2340, ISSN 1110-8657.
- [3] Cika, P.; Zukal, M.; Sebel, M.: Detection and REcognition of License PLates of Czech Vehicles. *Elektrorevue*, ročník 2, 2011, ISSN 1213-1539.
- [4] Dobeš, M.: *Zpracování obrazu a algoritmy v C#*. BEN, 2008, ISBN 978-80-7300-2, 143 s.
- [5] Duan, T. D.; Duc, D. A.; Du, T. L. H.: Combining Hough transform and contour algorithm for detecting vehicles' license-plates. In *Intelligent Multimedia, Video and Speech Processing*, Oct 2004, s. 747–750.
- [6] Feuereisl, D.: Poznávací značky automobilů v Českých zemích. [online], Naposledy navštíveno 30. 4. 2016.  
URL <http://www.feudal.cz/spz/html/1932-1939.htm>
- [7] Gonzalez, R. C.; Woods, R. E.: *Digital Image Processing*. Upper Saddle River: Prentice Hall, 2002, ISBN 0-13-094650-8, 793 s.
- [8] Hlaváč, V.; Sedláček, M.: *Zpracování signálů a obrazů*. Praha: České vysoké učení technické, 2009, ISBN 978-80-01-04442-1, 252 s.
- [9] Hájek, M.: České registrační značky. [online], Naposledy navštíveno 10. 5. 2016.  
URL [http://www.3260.cz/historie\\_poznavacich\\_znacek.htm](http://www.3260.cz/historie_poznavacich_znacek.htm)
- [10] Kamat, V.; Ganesan, S.: An efficient implementation of the Hough transform for detecting vehicle license plates using DSP'S. In *Real-Time Technology and Applications Symposium*, May 1995, ISSN 1080-1812, s. 58–59.
- [11] Nijhuis, J. A. G.; Brugge, M. H. T.; Helmholt, K. A.; aj.: Car license plate recognition with neural networks and fuzzy logic. In *Neural Networks, 1995. Proceedings.*, ročník 5, Nov 1995, s. 2232–2236.
- [12] Přinosil, J.; Krolkowski, M.: Využití detektoru Viola-Jones pro lokalizaci obličeje a očí v barevných obrazech. *Elektrorevue*, ročník 31, 2008, ISSN 1213-1539.
- [13] Sochor, J.: *Traffic Analysis from Video*. Diplomová práce, FIT VUT v Brně, 2014.



- [14] Sonka, M.; Hlaváč, V.; Boyle, R.: *Image processing, analysis, and machine vision*. Thomson, 2008, ISBN 0-495-24438-4, 866 s.
- [15] Vacek, M.: *Detekce poznávací značky v obraze*. Diplomová práce, FIT VUT v Brně, 2009.
- [16] Viola, P.; Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. 2001, s. 511–518.
- [17] Yang, F.; Ma, Z.: Vehicle license plate location based on histogramming and mathematical morphology. In *Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, Oct 2005, s. 89–94.
- [18] Zelený, F.; Feereisl, D.: *Poznávací značky v Čechách, na Moravě a ve Slezsku*. SAXI, 2011, ISBN 978-80-904767-2-1.
- [19] Šonka, M.; Hlaváč, V.: *Počítačové vidění*. Praha: Grada a.s., 1992, ISBN 80-85424-67-3, 272 s.

# Příloha A

## Obsah DVD

K práci jsou přiloženy dvě DVD. Tato DVD obsahují následující složky.

- **TechnickaZprava**  
Složka obsahuje technickou zprávu ve formátu PDF a její zdrojové kódy.
- **PrezentacniMaterial**  
Tato složka obsahuje plakát a video reprezentující tuto práci.
- **ZdrojoveKody**  
Složka obsahuje zdrojové kódy pro měření rychlosti a další pomocné programy. Postup překladu lze nalézt v souboru README.
- **Videa**  
Složka obsahuje videa ke zpracování a výsledná videa, která jsou výstupem programu.
- **TrenovaciSady**  
V této složce jsou trénovací sady pro kaskádový klasifikátor a pro klasifikátor SVM.