

# 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

## DETEKCE A ROZPOZNÁNÍ REGISTRAČNÍ ZNAČKY VOZIDLA

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

MICHAL ŘEPKA

BRNO 2015



**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

# DETEKCE A ROZPOZNÁNÍ REGISTRAČNÍ ZNAČKY VOZIDLA

LICENSE PLATE DETECTION AND RECOGNITION

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MICHAL ŘEPKA**

**VEDOUcí PRÁCE**

SUPERVISOR

**doc. Ing. ADAM HEROUT, Ph.D.**

BRNO 2015

## Abstrakt

Tato práce se zabývá detekcí a rozpoznáním objektů ze snímků s využitím metod počítačového vidění. Předmětem detekce jsou české registrační značky vozidel. Cílem práce bylo vytvořit algoritmus pro jejich automatickou anotaci. K tomu navrhované řešení využívá detekci hran a kaskádové klasifikátory vytvořené pomocí strojového učení. Implementace tohoto návrhu byla následně testována nad pořízenou datovou sadou za pomoci vytvořené testovací aplikace.

## Abstract

This paper addresses the problem of object detection and recognition from still images using methods of computer vision. The objects of detection are czech license plates and the goal of this paper was to create an automatic license plate annotation tool. Suggested solution uses edge detection and machine learned cascading classifiers. Created application was then tested on dataset taken by the author.

## Klíčová slova

Počítačové vidění, Detekce objektů, Registrační značka, RZ, Státní poznávací značka, SPZ, Rozpoznání znaků, Automatická anotace, Detekce hran, Strojové učení, Kaskádové klasifikátory

## Keywords

Computer vision, Object detection, Registration plate, RP, License Plate, LP, Character recognition, Automatic annotation, Edge detection, Machine learning, Cascading classifiers

## Citace

Michal Řepka: Detekce a rozpoznání registrační značky vozidla, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Detekce a rozpoznání registrační značky vozidla

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Adama Herouta, Ph.D.

.....

Michal Řepka

20. května 2015

## Poděkování

Chtěl bych poděkovat především vedoucímu práce panu doc. Ing. Adamu Heroutovi, Ph.D., který poskytoval svůj drahocenný čas a rady na pravidelných konzultacích. Dokonce i o státních svátcích.

Děkuji.

© Michal Řepka, 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

<b>1 Úvod</b>	<b>2</b>
<b>2 Specifikace použití</b>	<b>3</b>
2.1 Úhly a vzdálenosti . . . . .	3
2.2 Přesnost výsledku . . . . .	4
2.3 Rychlost zpracování . . . . .	4
2.4 Formát registračních značek . . . . .	4
<b>3 Použité metody pro rozpoznání objektů</b>	<b>5</b>
3.1 Cannyho detektor hran . . . . .	5
3.2 Kaskáda klasifikátorů . . . . .	6
<b>4 Popis navrženého algoritmu</b>	<b>9</b>
4.1 Detekce kandidátních oblastí . . . . .	10
4.2 Rozpoznání znaků . . . . .	11
4.3 Logika selekce výsledku . . . . .	11
4.4 Segmentace znaků . . . . .	13
<b>5 Implementace</b>	<b>14</b>
5.1 Trénování klasifikátorů . . . . .	14
5.2 Aplikace pro testování . . . . .	15
<b>6 Testování a analýza výsledků</b>	<b>17</b>
6.1 Datové sady . . . . .	17
6.2 Výsledky detektoru . . . . .	18
6.3 Výsledky rozpoznání znaků . . . . .	19
6.4 Celkové výsledky rozpoznání RZ . . . . .	20
<b>7 Závěr</b>	<b>21</b>
<b>A Obsah CD</b>	<b>24</b>

# Kapitola 1

## Úvod

Tato práce se zabývá vytvořením algoritmu pro rozpoznání registrační značky vozidla (dále jen RZ) ze snímku, který by bylo možné využít v mobilní aplikaci. V praxi by taková aplikace měla sloužit při dokumentaci vozidel a zároveň automatické anotaci jejich RZ. Potencionální uživatelé by mohli být například policisté, kteří by tak urychlili a zefektivnili svoji práci.

Většina existujících řešení tohoto problému je využita v kamerových systémech, na parkovištích nebo kolem pozemních komunikací. I když byly inspirací, tato práce bere především v úvahu požadované vlastnosti výsledné aplikace, které se kvůli pohyblivosti fotoaparátu, stálé přítomnosti uživatele a potřebné rychlosti zpracování od existujících, statických řešení mírně odkloňují.

Tyto požadavky na systém budou popsány v kapitole 2. Teoretické základy, na kterých práce staví, tedy praktiky využívané ve zpracování obrazu popřípadě počítačového vidění budou diskutovány v kapitole 3. Z těchto teoretických základů bylo navrženo řešení popsané v kapitole 4, tedy algoritmus rozpoznání RZ, skládající se ze tří na sebe navazujících částí. Kapitola 5 popisuje prostředky a pomůcky, ať už využité nebo vytvořené autorem, které umožnily realizaci návrhu. Část algoritmu zodpovědná za rozpoznání znaků využívá strojové učení, k čemuž potřebuje snímky RZ tak, jak budou foceny ve výsledné aplikaci. Bylo tedy nutné vytvořit datovou sadu obsahující velké množství různých znaků. Část této sady byla využita i jako testovací sada pro experimenty. Jejich výsledky a analýzu, stejně jako popis datových sad, je možné vidět v kapitole 6.

## Kapitola 2

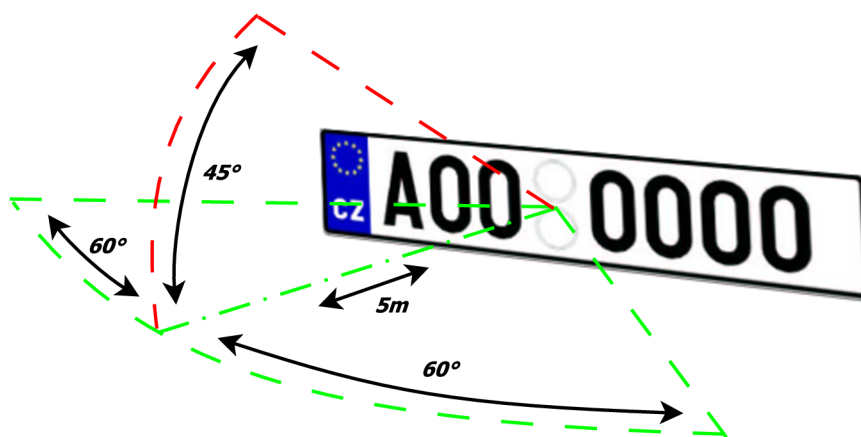
# Specifikace použití

Jak již bylo zmíněno v úvodu, vyvíjený systém má předem známé použití na mobilní platformě, které na něj klade výrazně odlišné nároky než kdyby bylo vyvíjeno na kamerový, statický systém. V této kapitole budou relevantní nároky formulovány, odůvodněny a budou definovány důsledky, které potencionálně ovlivní návrh algoritmu.

### 2.1 Úhly a vzdálenosti

Mýtné brány nebo zařízení pro měření úsekové rychlosti se mohou spolehnout, že focená vozidla budou stále ve stejné vzdálenosti a úhlu od zařízení. Díky tomu mohou počítat s určitými vlastnostmi oblastí RZ. Jsou mezi nimi často detekované hrany [2] nebo barva v kombinaci s velikostí oblasti [12]. Pokud chceme takový systém použít na různých kamerách, je možné algoritmus zkalibrovat a od té chvíle dokáže fungovat se statickými parametry.

Vzhledem k možnostem pohybu fotoaparátu v prostředí, musí být řešení na mobilní platformu obecnější nebo obsahovat restriktce při jeho použití. V rozsahu této práce je vhodné zvolit kompromis mezi těmito možnostmi a umožnit tak uživateli pohyb, ale jen v omezených vzdálenostech a úhlech od oblasti RZ. Tyto omezení je možné vidět na obrázku 2.1 a bude s nimi počítáno v dalších krocích této práce.



Obrázek 2.1: Ukázka úhlů, ze kterých je vhodné fotit.

## 2.2 Přesnost výsledku

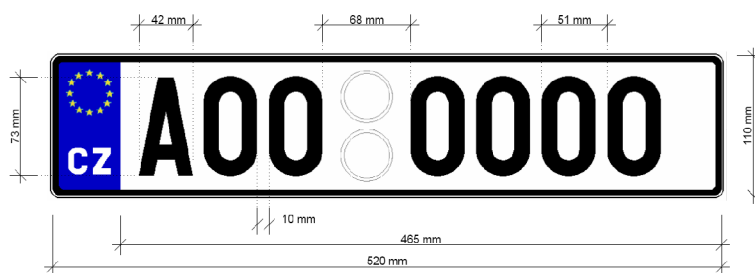
Kromě omezení přináší mobilní platforma i určité možnosti. Je samozřejmé, že vyvíjené řešení nebude pracovat autonomně. Na druhé straně fotoaparátu bude vždy člověk, který si je vědom správného výsledku rozpoznání RZ. Je pravděpodobné, že tento výsledek bude také potvrzovat. Chybné výsledky rozpoznání tedy není nutné zahazovat, například ve strachu z mylného pokutování vozidla, ale je možné uživateli nabídnout v případě nejistoty nějaké možnosti. To znamená, že vyvíjené řešení se bude v každém případě snažit o co nejpřesnější výsledky, ale ty méně pravděpodobné uživateli alespoň nabídne. Díky tomu nebude nutné fotit vozidlo znovu, což by celý proces dělalo poněkud frustrujícím.

## 2.3 Rychlost zpracování

V kamerových systémech je častá nutnost pro zpracování snímků v reálném čase nebo minimálně takovou rychlostí, aby ve chvíli, kdy projede další vozidlo, byl systém připraven. V aplikaci na mobilní platformě je rychlost zpracování druhořadá. Cílová rychlost zpracování jednoho snímku bude dostačující, pokud nepřekročí určitý práh, kdy by byla nepříjemná pro používání aplikace. Tento práh je u každého jiný, takže byl zvolen „vhodně“ okolo 2 sekund. V případě použití v praxi by bylo vhodné tento práh zjistit na sadě skutečných uživatelů. Cílem této práce budou zmíněné 2 sekundy s tím, že bude upřednostňována přesnost před rychlostí, avšak ne za každou cenu.

## 2.4 Formát registračních značek

Tato práce se zaměřuje na nejčastější typy RZ autobusů, osobních a nákladních vozidel, což je dle vyhlášky Ministerstva dopravy 243/2001 Sb. typ 101 [7]. Formát těchto značek má podobu vyobrazenou na obrázku 2.2 od roku 2001. Významné vlastnosti jsou šířka a výška respektive poměr stran pro účely detekce, a rozmístění znaků a pole evropské unie pro rozpoznání obsahu RZ. Velkou výhodou při filtrování falešných pozitivů je znalost na kterých pozicích mohou být písmena a kde nesmí. Na druhé pozici je vždy písmeno, na první a třetí pozici může být číslo i písmeno a zbylé čtyři pozice jsou vždy obsazeny čísly.



Obrázek 2.2: Ilustrace formátu RZ, převzato z [9].

Vzhledem k použití klasifikátorů pro rozpoznání znaků není nutné brát v úvahu emisní známky nebo pomlčku na RZ vydaných před rokem 2001. Tyto starší RZ mají také mírně odlišné rozprostření znaků a obsahově povinné písmena na všech třech levých pozicích.

## Kapitola 3

# Použité metody pro rozpoznání objektů

V této kapitole budou popsány použité technologie pro rozpoznání a klasifikaci objektů v obraze. Budou diskutovány důvody jejich výběru v kontextu s vlastnostmi detekovaného subjektu a případy použití v praxi.

Registrační značka je složena ze silně kontrastních částí a jako celek má přesně daný tvar. Způsob její detekce navíc musí být tolerantní k různým vzdálenostem od objektu, jak bylo popsáno v kapitole 2. Z těchto důvodů byla zvolena detekce hran jako prvotní charakteristika vstupních dat. Z ní je možné jít mnoha cestami. Je možné využít statistické metody díky velkému množství hran v okolí znaků [5, 11], najít hrany daného tvaru pomocí Hughovy transformace nebo detekovat jednotlivé objekty na základě spojených hran (využito zde). Pro samotnou detekci hran byl zvolen Cannyho detektor 3.1 vzhledem k jeho celkově lepším výsledkům (viz [6]) a schopnosti ignorovat šum, kterého ve venkovním prostředí kolem vozidel může být hodně.

Dalším netriviálním prvek algoritmu, který je nutné popsat, je rozpoznání a klasifikace znaků. Pro tento úkol se často používají metody založené na porovnání se šablonou [5], s využitím neuronových sítí nebo SVM<sup>1</sup>. Zde budou využity kaskádové klasifikátory 3.2 reprezentující objekty pomocí LBP<sup>2</sup> příznaků. Tyto příznaky jsou schopny zachytit velice malé detaily a fungují dobře i v různých světelných podmínkách (funguje i u špinavých RZ). Vytváří rychleji výkonnější klasifikátory než Haarovy obdélníkové příznaky. Vzhledem k nutnosti klasifikace velkého množství znaků, které si jsou navíc dosti podobné, jsou LBP příznaky vhodnějším kandidátem.

### 3.1 Cannyho detektor hran

Cannyho detektor hran [4] je označován za optimální detektor hran, který svojí přesností předčil všechny své předchůdce. Toho dosáhl díky úspěšné eliminaci chyb jiných detektorů a sledováním těchto cílů:

- nízký počet chybně označených nebo neoznačených hran,
- body hran jsou co nejpřesněji lokalizovány,

---

<sup>1</sup>SVM – Support Vector Machines.

<sup>2</sup>LBP – Local Binary Patern.

- pouze jeden výsledek pro jednu hranu.

Jedná se o kritické vlastnosti při detekci RZ, protože Cannyho algoritmus tímto zajišťuje, že budou nalezeny maximálně dvě kandidátní oblasti u RZ – jedna, vnitřní, ohraničující přechod mezi pozadím RZ a plastovým rámem nebo polem evropské unie, a druhá, vnější, odpovídající přechodu mezi rámem a nárazníkem vozidla. Vnitřní oblast je samozřejmě preferovaná, protože už je oříznutá přesně okolo bílého pozadí.

Samotný algoritmus nejdříve vyčistí snímek od šumu pomocí Gaussova filtru, poté získá velikost gradientu v každém bodě aplikováním 3x3 Sobelova operátoru (viz obrázek 3.1) pro horizontální a vertikální směr. Výsledná velikost  $G$  je poté vypočtena:  $|G| = |G_x| + |G_y|$ .

-1	0	+1
-2	0	+2
-1	0	+1

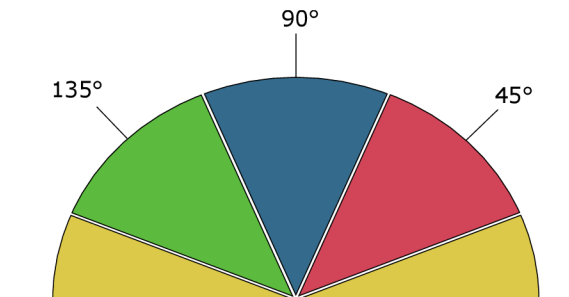
(a)  $G_x$

+1	+2	+1
0	0	0
-1	-2	-1

(b)  $G_y$

Obrázek 3.1: Sobelovy operátory pro horizontální (vlevo) a vertikální (vpravo) směr

Dalším krokem je výpočet směru hrany v každém bodě:  $\theta = \tan^{-1}(G_y/G_x)$ . Tento úhel musí být poté převeden do obrazu, kde každý pixel má pouze omezený počet sousedů, kam může hrana pokračovat, takže výsledné úhly musí být rozděleny do 4 tříd reprezentující jednotlivé směry (viz obrázek 3.2).



Obrázek 3.2: Ilustrace rozdělení úhlů při rasterizaci hran.

V okolí hranových bodů jsou dále ponechány jen lokální maxima, pro ztenčení výsledné hrany, což je nutné při hledání spojitých hran objektů. Nakonec je aplikováno prahování v rozmezí dvou hodnot, mezi kterými mohou velikosti gradientu kolísat.

## 3.2 Kaskáda klasifikátorů

Kaskáda klasifikátorů je rychlý způsob detekce objektů v obraze. Původně byl vyvinut pro detekci obličejů autory Viola a Jones [10], ovšem příznaky reprezentující detekovaný objekt je možné vytvořit nad čímkoli.

Svou rychlost algoritmus získává díky několika inovacím, které přináší. Zprvė využívá integrální obraz pro reprezentaci obrazových dat, kdy každý pixel obsahuje sumu pixelů

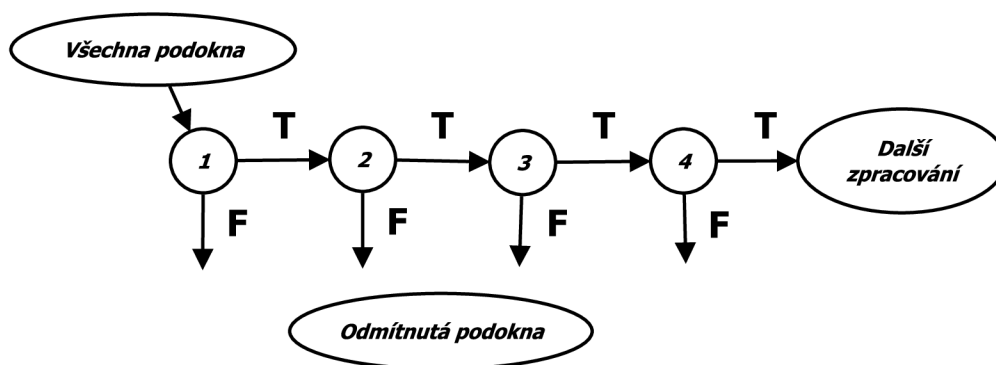
lokalizovaných od něj vlevo a nahoru. Toto je možné zapsat vztahem:

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

Snímek je tedy možné převést do integrální reprezentace jedním průchodem a poté vypočítat příznaky v konstantní časové složitosti.

Zadruhé jsou pomocí algoritmu AdaBoost<sup>3</sup> vybírány jen některé příznaky, ze kterých se skládá klasifikátor. Většina rysů objektu je tedy ignorována, aby bylo možné identifikovat ty nejdůležitější.

A nakonec jsou při detekci objektů pomocí klasifikátorů aplikovány nejdříve ty jednodušší, které rychle odstraní většinu negativních podoken. Poté jsou povolány ty sofistikovanější, aby přesně určily pozice hledaných objektů. Odtud tedy pramení slovo *kaskáda* klasifikátorů. V praxi se jedná o rozhodovací strom (viz obrázek 3.3).

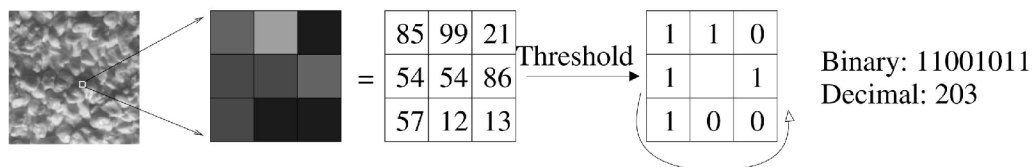


Obrázek 3.3: Rozhodovací strom představující kaskádu.

<sup>3</sup>Viz <http://en.wikipedia.org/wiki/AdaBoost>.

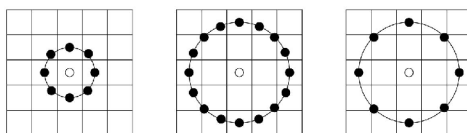
## Příznaky LBP

Operátor přiřadí každému pixelu hodnotu na základě jeho okolí tak, že okolní pixely vyprahuje se svojí hodnotou, přiřadí jim tak binární číslo 0 nebo 1 a tedy 8 okolních pixelů dává tomuto středovému 8-bitovou hodnotu (znázorněno na obrázku 3.4). Histogram vytvořený z ohodnocení jednotlivých pixelů pak může být použit jako popis rysů textury.



Obrázek 3.4: Hodnocení LBP operátorem, převzato z [1].

Aby bylo možné pracovat s texturami v různých velikostech, tento základní operátor byl rozšířen o práci s většími okolími a také o interpolaci bodů okolí do kruhu se stejnými vzdálenostmi mezi nimi. Takto je možné kruh a počet bodů v něm libovolně měnit (znázorněno na obrázku 3.5).



Obrázek 3.5: Příklady kruhových okolí, převzato z [1].



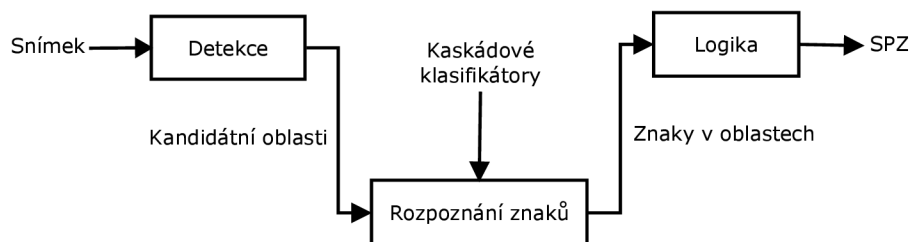
## Kapitola 4

# Popis navrženého algoritmu

V této kapitole bude podrobně popsán navržený způsob detekce a rozpoznání RZ. Návrh bere v potaz vlastnosti řešení vyplývající z použití mobilní platformy, které byly diskutovány v kapitole 2. Prioritami tedy jsou:

1. nalezení výsledku pokud možno přesně, u špatně určených výsledků alespoň nabídnout ten správný
2. rychlost zpracování jednoho snímku okolo 2 sekund,
3. tolerance RZ vyfocených z úhlu.

Samotný algoritmus se skládá ze tří částí znázorněných na schématu 4.1. Ve vstupním snímku jsou nejdříve nalezeny kandidátní oblasti na základě jejich tvaru. V těchto oblastech jsou následně nalezeny znaky, jejichž pozice a hodnota udává ohodnocení celé oblasti. Na základě tohoto ohodnocení je určena jedna z oblastí jako nejpravděpodobější registrační značka.



Obrázek 4.1: Schéma algoritmu.

Zde je možné vidět zaměření algoritmu na přesnost, protože každý jeho krok nehledá jeden přesný výsledek, ale má poměrně vysokou toleranci a jako výstup bere všechny oblasti nebo znaky, které do této tolerance spadají. Pokud by bylo vyvíjeno řešení nad videem z kamerových systémů, kde je potřeba zpracování v reálných rychlostech, každá část algoritmu by musela přesně určit výsledek, aby nedocházelo ke zbytečnému zpracování falešných pozitivů.

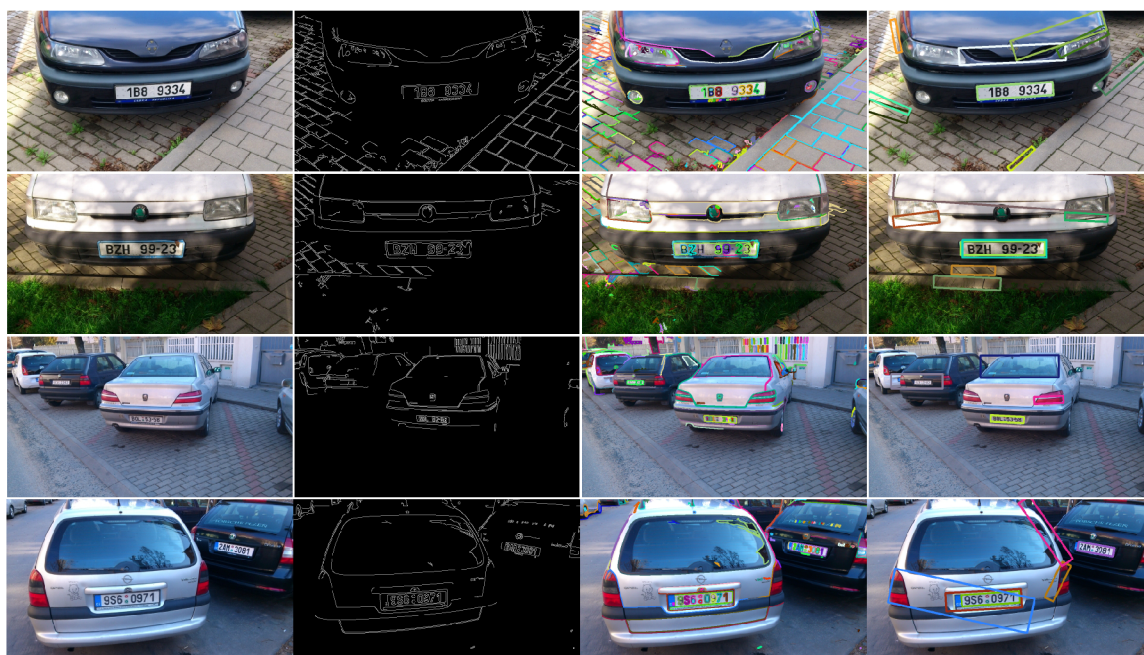
Řada existujících řešení (například [5]) obsahuje navíc ještě jednu část algoritmu – segmentaci znaků. Ta navazuje přímo na detektor RZ a umožňuje brzké rozlišení správných od špatných oblastí na základě rozprostření znaků. Pokud segmentace funguje, je to velice

silný nástroj pro zrychlení a zpřesnění rozpoznání, protože výrazně snižuje velikost plochy, na které se musí hledat znaky. Její implementace 4.4 ovšem nebyla dostatečně úspěšná, takže byla využita pouze pro automatizaci vyřezávání znaků do sad, které byly potřebné ke strojovému učení klasifikátorů 5.1. Bez segmentace jsou klasifikátory aplikovány na celé kandidátní oblasti, což z tohoto procesu vytváří časově nejnáročnější činnost napříč algoritmem (viz tabulka 6.6), avšak přináší lepší výsledky.

## 4.1 Detekce kandidátních oblastí

Zmenšený vstupní snímek je nejdříve vyhlazen, aby při následné detekci hran byly nalezeny jen ty nejostřejší hrany. Hranice mezi bílým pozadím registrační značky a jejím černým umělohmotným rámem mezi ně v naprosté většině patří. Tato hranice zaniká až u značek ve velké vzdálenosti nebo u značek v netypických rámech a s nárazníkem světlé barvy. Pro nalezení hran je použit Cannyho detektor (viz 3.1) s velikostí matice 3, spodním prahem 100 a horním prahem 255. Aby bylo možné rozpoznat hranice jednotlivých objektů, jsou jimi nataženy kontury, které odpovídají souvislým hranám. Takto nalezené hranice objektů je dále možné uzavřít do obdélníku reprezentujícího nalezenou oblast, zvláště vzhledem k obdélníkovému tvaru hledaného objektu. V případě potřeby jsou nalezené oblasti v obdélnících otočeny, aby ležely horizontálně.

Nyní přichází na řadu filtrace nalezených objektů. Jejím hlavním parametrem je poměr stran, který je známý ze sekce 2.4. Vzhledem k tomu, že registrační značky mohou být vyfoceny z mírných úhlů, tento poměr stran se samozřejmě zkresluje, takže je na místě určitá tolerance. Na obrázku 4.2 můžete vidět vizuální výstupy z jednotlivých kroků.



(a) Zdrojové snímky (b) Extrahované hrany (c) Kontury (d) Kandidátní oblasti

Obrázek 4.2: Výstupy jednotlivých kroků detekce.

Tímto postupem vzniká okolo sedmi kandidátních oblastí při filtrování všech oblastí, jejichž poměr stran nespadá do tolerance 33%.

## 4.2 Rozpoznání znaků

Tato část algoritmu má za úkol nalézt všechny znaky odpovídající fontu registračních značek v kandidátních oblastech. K tomu využívá naučené kaskádové klasifikátory (viz sekce 3.2) jednotlivých znaků (viz 5.1). V ideálním případě by tyto klasifikátory neomylně určily pouze znaky registračních značek. Po mnoha experimentech se strojovým učením, se ovšem ukázalo, že bude nutné se vypořádat s falešnými pozitivy, které můžete vidět na obrázku 4.3.



Obrázek 4.3: Ukázka false positivů u oblastí vyfocených z různých vzdáleností.

Registrační značky vyfocené z větších vzdáleností na obrázku (b) a (d) obsahují menší množství falešných pozitivů oproti těm zblízka (a) a (c), což souvisí s velikostí znaků, na kterých se trénovaly klasifikátory. Experimentováno bylo také s klasifikátory z větších i menších znaků. S většími znaky se ovšem nedalo nalézt znaky v menších RZ a s menšími znaky zase vznikalo velké množství falešných pozitivů, protože se již ztrácely informace o detailech mezi znaky.

## 4.3 Logika selekce výsledku

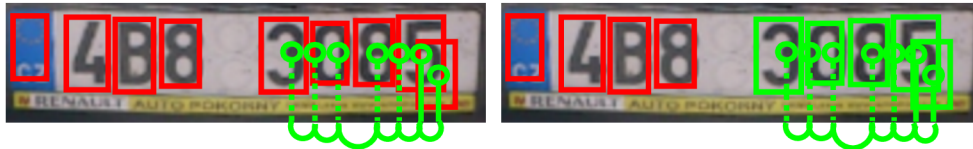
Vzhledem k množství falešných pozitivů, zmíněných v minulých sekcích, je potřeba navrhnout rozhodovací logiku, která dokáže vybrat z nabízených kandidátních oblastí a jejich znaků takový výsledek, jehož vlastnosti nejvíce odpovídají registrační značce. Je nutné, aby taková logika brala v úvahu i možnost, že žádnou z navrhovaných oblastí registrační značka není. Ohodnocení oblastí navrhovaného algoritmu probíhá na základě znaků které obsahuje. O znacích jsou známy tyto informace:

- pozice
- velikost
- hodnota
- priorita (bude vysvětleno v podsekcí 4.3)

K rozpoznání registrační značky jsou využity pozice a velikosti znaků, a to tak, že jsou vyhledány znaky s podobným rozprostřením, jako na RZ (viz 2.4). Vzhledem k tomu, že oblasti obsahují z pravé strany jako první nebílý objekt poslední znak (hledaný subjekt), ale z levé strany pole evropské unie (potencionální falešný pozitiv), jsou pozice znaků a mezery mezi nimi testovány s počátkem na pravé straně. Stejně jako implikuje obrázek 4.4, každý

znak je postupně označen za správný poslední znak RZ a k němu se hledá předposlední. Se získanou mezerou mezi znaky a šířkou znaků jsou odhadnuty pozice zbylých pěti. Je tedy kritické pro úspěšnost algoritmu, aby oba poslední znaky registrační značky byly nalezeny klasifikátory.

Rozdíl mezi nalezenými pozicemi (a) a (b) na obrázku 4.4 je v množství znaků, které tyto pozice pokrývají. Z výstupu minulého kroku vyplývá, že většina znaků leží přímo na nebo okolo reálných znaků ve snímku. Čím větší procento znaků tedy odhadnuté pozice pokrývají, tím je vyšší šance, že se jedná o správné pozice. To je ještě umocněno faktem, že oblasti reprezentující RZ přichází z detekce oříznuty poměrně přesně.



(a) Pozice odhadnuté z falešného pozitivu



(b) Pozice odhadnuté ze znaků RZ

Obrázek 4.4: Ilustrace nalezení pozic.

## Priority znaků

Kaskádové klasifikátory jsou výjimečně užitečné v rozpoznávání objektů jako je obličej, oči, postava, atd. Předmětem tohoto algoritmu jsou ovšem znaky, které se mohou lišit naprostými detaily. Vzhledem k tomu, že jsou klasifikátory trénovány nad pozitivy o velikosti 12x18 pixelů, vzniká u některých z nich množství záměn s jinými. Po nalezení správných pozic na správné oblasti tedy není nutné, že se zde nachází označení pouze jednoho znaku. Například mezi B a 8 nebo 5 a 6 dochází k pravidelným záměnám. Aby bylo možné určit o který znak se jedná bez dodatečného aplikování klasifikátorů nebo jejich sjednocení, byl implementován systém priorit, který závisí na tom, že jeden znak často falešně označuje druhý, ale druhý neoznačuje ten první. Pokud tedy například 6 označí oblast za 6, označení klasifikátorem 5 již není na této pozici uvažováno, protože se ukázalo, že 5 někdy označuje symbol 6. Tímto způsobem by teoreticky mohlo dojít k pravidelným záměnám, systém byl ovšem testován (viz 6.4) a pokud se znaky nezaměňují v cyklu, priority by tento problém měly vyřešit.

Při tomto procesu pomáhá i fakt, že na některých pozicích RZ mohou být pouze čísla nebo písmena (viz 2.4). Jen na minimu pozic mohou být obě. Tím se výrazně sníží množství potenciálních záměn. Pořadí aplikace kaskádových klasifikátorů tedy určuje priority symbolů a je následující:  $3 > 4 > 5 > 2 > 9 > 8 > 6 > R > A > E > N > 0 > 7 > 1 > S > Z > T > V > U > J > M > B > H > L$ .

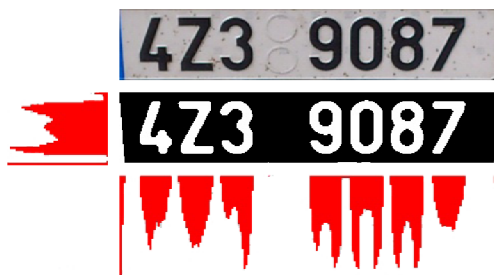


## 4.4 Segmentace znaků

Učení kaskádových klasifikátorů vyžaduje co možná největší množství vzorků snímků subjektu a příprava – vyřezávání, je při manuálním zpracování zdlouhavá práce. Pro účely automatizace tohoto procesu byl implementován algoritmus segmentace znaků.

### Projekce pixelů

Algoritmus využívá projekci bílých pixelů z černo-bílých snímků. Vytváří tak dva histogramy, jeden pro řádky a jeden pro sloupce. Tyto procesy jsou tedy nazývány horizontální a vertikální projekcí. Nejprve je provedena vertikální projekce a oříznutí horního a spodního okraje kolem největší nenulové posloupnosti. Tím jsou vyčištěny od šumu mezery mezi znaky pro následnou horizontální projekci. Ta vyhledává znaky v pozicích podobných registrační značce a začíná zprava kvůli již popisovanému poli evropské unie na levé straně RZ.



Obrázek 4.5: Horizontální a vertikální projekce pixelů.

## Kapitola 5

# Implementace

Implementace navrhovaného algoritmu a jeho testování proběhlo na platformě Windows. Jelikož je nutné v budoucnu vše přenést na platformu Android, byl zvolen programovací jazyk C++ a dále zejména otevřená, multiplatformní knihovna OpenCV [3], která abstrahuje funkce z oblasti počítačového vidění a zpracování obrazu. Prostředí Qt umožnilo vyvíjení algoritmu a implementaci testovací aplikace na Windows. Dále také umožňuje vytvoření uživatelského rozhraní pro případnou Android aplikaci, takže přenesení řešení na mobilní platformu by nemělo být příliš složité.

### 5.1 Trénování klasifikátorů

Kaskádové klasifikátory, jako páteř rozpoznání registrační značky, byly trénovány pomocí prostředí implementovaného v OpenCV. Jedná se o program `opencv_traincascade`, který očekává na svém vstupu snímky hledaného objektu – pozitivy a snímky pozadí, od kterých je objekt nutné rozlišit – negativy. Více praktických informací o procesu vytváření kaskádových klasifikátorů pomocí OpenCV, je možné najít ve skvělém tutorialu od Naotoshi Seo [8].

#### Pozitivní snímky

Pozitivní snímky byly vytvořeny automatickou segmentací znaků z datové sady a jejich následným zmenšením na 12x18 pixelů. Za pomoci podpůrného programu OpenCV byly tyto pozitivy transformovány do různých úhlů, které jsou podporovány aplikací, aby se jejich celkové množství pohybovalo okolo 400 snímků. Pokud jsou tyto pozitivy vytvořeny z alespoň 20 různých snímků, výsledné klasifikátory jsou dostatečně kvalitní. Některých písmen není v datové sadě dostatek, protože byla nafocena pouze v Brně, takže jejich klasifikátory nejsou v momentální chvíli naučeny a nejsou také zahrnuty ve výsledcích (viz 6.3).

#### Negativní snímky

Negativní snímky byly vytvořeny vyřiznutím oblastí registračních značek, a protože se v nich nesmí objevovat hledané objekty, byl z negativů každého klasifikátoru vymazán znak, pro který se tento klasifikátor učí. Celkový počet takto použitých negativů byl 280.

## Učení klasifikátorů

Po připravení pozitivních a negativních snímků je možné následovně použít program pro naučení kaskádových klasifikátorů:

```
opencv_traincascade
    -data Cascades
    -vec A_samples.vec
    -bg Negatives_RZ“negs.txt
    -numPos 394
    -numNeg 280
    -numStages 15
    -featureType LBP
    -w 12
    -h 18
    -minHitRate 0.999
    -maxFalseAlarmRate 0.5
```

Více o parametrech a použití těchto programů je možné se dočíst v jejich krátké dokumentaci<sup>4</sup>.

## 5.2 Aplikace pro testování

Příložená aplikace byla vytvořena pro účely anotace datové sady, přípravy sad znaků na učení klasifikátorů a především testování vytvořeného algoritmu. Umožňuje otevření složky, prohlížení jednotlivých fotek a provést nad nimi zmíněné činnosti.

### Anotace

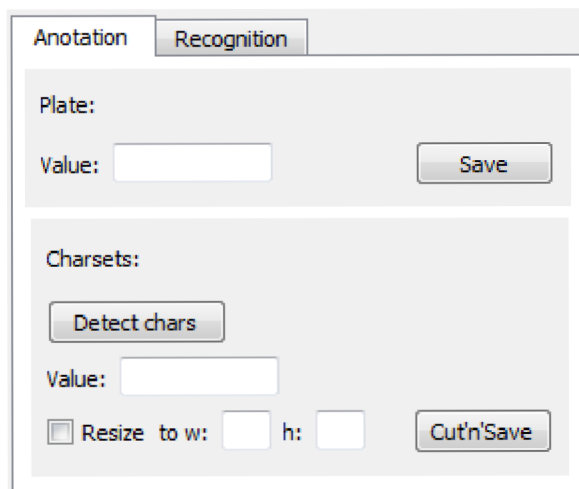
Obsah RZ vložený do pole *Plate:Value* je po kliknutí na *Save* uložen do souboru *anotace.txt*, který je vytvořen pro celou aktuálně otevřenou složku. Po anotaci všech snímků ve složce tedy tento soubor obsahuje na každém řádku jméno souboru s přiřazeným obsahem RZ. Algoritmus pro automatické vyřezávání znaků je oddělen od anotace RZ, protože nefunguje ve všech případech. Pole *Charsets:Value* je tedy vhodné vyplnit podle výsledku segmentace, který se zobrazí po kliknutí na tlačítko *Detect chars*. Falešné pozitivivy ve výsledcích segmentace je dobré nahradit znakem pomlčky, aby nedošlo k posunutí všech znaků a tedy chybné anotaci znaků. Kliknutím na *Cut'n'Save* jsou znaky vyříznuty a uloženy do složek */charsets/char*.

### Rozpoznání

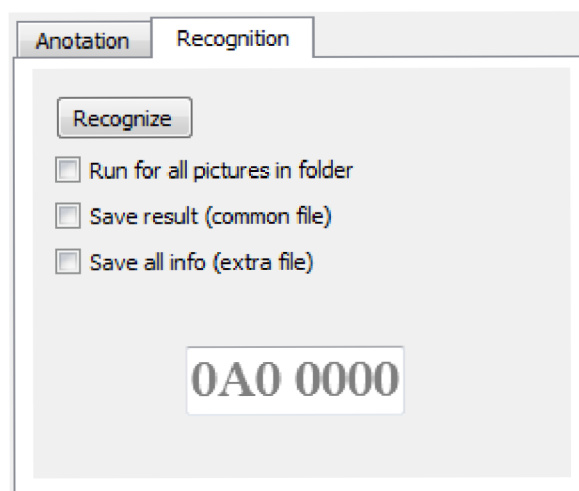
Karta *Recognition* slouží pro testování implementovaného algoritmu. Po kliknutí na *Recognize* je spuštěn na aktuálním snímku. Kandidátní oblasti jako výsledky detekce jsou zobrazeny na pravé straně a rozpoznáný obsah RZ je zobrazen v poli *Result*. Další doplňující informace jako rychlost zpracování, počet kandidátních oblastí a zamítnuté znaky na jednotlivých pozicích RZ, jsou zobrazeny v poli *Info* a mohou být uloženy do speciálních souborů pro každý snímek.

---

<sup>4</sup>[http://docs.opencv.org/doc/user\\_guide/ug\\_traincascade.html](http://docs.opencv.org/doc/user_guide/ug_traincascade.html)



Obrázek 5.1: Možnosti aplikace pro anotaci.



Obrázek 5.2: Možnosti aplikace pro rozpoznání RZ.



## Kapitola 6

# Testování a analýza výsledků

V této kapitole budou popsány výsledky dosažené implementovaným algoritmem a jeho částmi. Pro tento účel byla vytvořena testovací sada 6.1. Dále budou tyto výsledky analyzovány a pokud možno nalezeny aspekty řešení, které jsou přínosné nebo naopak chybné.

### 6.1 Datové sady

Výsledky prezentované dále byly zjištěny nad testovací sadou nacházející se na přiloženém disku. Všechny snímky byly pořízeny autorem této práce. Testovací sada je rozčleněna podle vzdálenosti a úhlu od registrační značky do složek v tabulce 6.1 a ukázky jednotlivých sad je možné vidět na obrázcích 6.1.

Sada	Počet snímků	Vzdálenost od RZ	Úhel od RZ
A	42	$< 2m$	$< 15^\circ$
B	115	$< 4m$	$< 15^\circ$
C	23	$\geq 4m$	$< 15^\circ$
D	44	různé	$\geq 15^\circ$

Tabulka 6.1: Testovací sady.

Kromě této testovací sady určené pro testování implementovaného algoritmu, byla pořízena ještě datová sada zaměřená na jiná písmena než je jen B, aby bylo možné vytvořit kvalitní kaskádové klasifikátory pro znaky jako je například J, M nebo N, které se také hojně vyskytují v okolí Brna. Tato sada je pouze částečnou součástí testovací sady a také je přiložena k této práci.

Pro účely strojového učení byly pomocí automatizované segmentace vytvořeny také sady jednotlivých znaků. V tabulce 6.2 můžete vidět jejich počty a také jestli toto množství bylo dostatečné k vytvoření klasifikátoru.



(a) Sada A

(b) Sada B

(c) Sada C

(d) Sada D

Obrázek 6.1: Ukázky z jednotlivých složek testovací sady.

Znak	Počet v sadě	Klasifikátor	Znak	Počet v sadě	Klasifikátor
0	67	ANO	H	16	ANO
1	94	ANO	I	7	NE
2	35	ANO	J	26	ANO
3	100	ANO	K	7	NE
4	99	ANO	L	12	ANO
5	87	ANO	M	26	ANO
6	82	ANO	N	11	ANO
7	84	ANO	P	7	NE
8	101	ANO	R	13	ANO
9	111	ANO	S	17	ANO
A	38	ANO	T	21	ANO
B	76	ANO	U	5	ANO
C	12	NE	V	4	ANO
D	11	NE	X	3	NE
E	20	ANO	Y	3	NE
F	3	NE	Z	65	ANO

Tabulka 6.2: Sady znaků pro strojové učení.

## 6.2 Výsledky detektoru

První z testovaných částí je detektor kandidátních oblastí, který byl navržen tak, aby propustil všechny objekty obdélníkového tvaru. V tabulce 6.3 je vidět, že s celkovou úspěšností 96,9% byla implementace úspěšná. Je ovšem nutné počítat s tím, že kromě oblastí RZ tato

část také propouští velké množství falešných pozitivů. Tato skutečnost se může projevit v navazujících krocích.

	Sada				Celkem
	A	B	C	D	
Počet RZ	42	115	23	44	224
Počet nalezených RZ	41	110	22	44	217
Počet nenalezených RZ	1	5	1	0	7
Úspěšnost	97,6%	95,7%	95,7%	100%	96,9%

Tabulka 6.3: Úspěšnost detekce RZ.

### 6.3 Výsledky rozpoznání znaků

Rozpoznání znaků a tím pádem celé RZ je závislé na kvalitě kaskádových klasifikátorů. V tabulce 6.2 je možné vidět, že byly vytvořeny i klasifikátory z velice malých sad některých znaků. To mělo docílit možnosti rozpoznání většího množství RZ, ale bylo kontraproduktivní. V tabulce 6.4 je možné vidět úspěšnost rozpoznání jednotlivých znaků na správně rozpoznávaných RZ. Některé z nich (zejména písmena) doplatily na nízkou kvalitu jejich klasifikátorů a buď dosáhly nízké úspěšnosti nebo často zastupovaly místo jiným znakům. To je částečně způsobeno i jejich zmíněnými prioritami. Tuto chybu je možné zpozorovat například na písmeně B, které je často zastoupeno písmenem M. Už z tabulky je patrné, že M a B se navzájem zastupovaly, ale M dostalo vyšší prioritu. U čísel, které oproti písmenům dominují co se týče jejich množství na RZ, není vidět pravidelná chybovost. Jejich úspěšnost by tedy mělo být možné zvýšit vytvořením kvalitnějších klasifikátorů.

Znak	0	1	2	3	4	5	6	7	8	9	A	B	E	H	J	L	M	N	R	S	T	U	V	Z	∅	∑	Úspěšnost
0	81	1	1	0	1	0	2	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	11	100	81.0%
1	1	99	0	1	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	126	78.5%
2	1	0	100	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	15	122	81.9%
3	1	0	1	111	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	16	132	84.0%
4	1	0	0	2	125	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	12	142	88.0%
5	1	1	0	0	0	89	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	106	83.9%
6	3	0	0	0	0	1	87	0	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	112	77.6%
7	5	0	2	0	0	1	0	114	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	7	130	87.6%
8	2	1	0	0	0	0	1	1	113	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	10	132	85.6%
9	1	1	0	0	1	0	0	0	0	117	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	137	85.4%
A	0	0	0	0	1	0	0	0	2	0	21	0	0	0	3	0	0	0	0	0	0	0	0	0	5	32	65.6%
B	0	0	0	0	0	0	0	0	0	0	1	77	2	0	0	0	10	0	0	0	0	1	0	0	12	103	74.7%
E	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	3	19	84.2%
H	0	0	0	0	0	0	0	0	0	0	0	3	0	1	1	0	3	0	0	0	0	2	0	0	1	11	9.0%
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	6	28	78.5%
L	2	0	1	0	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	1	11	54.5%
M	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	10	1	0	0	0	0	0	1	3	17	58.8%
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	1	8	87.5%
R	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	4	0	0	0	0	0	1	7	57.1%
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	16	100.0%
T	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	21	90.4%
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	4	100.0%
V	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	3	66.6%
Z	0	0	0	0	0	0	0	0	0	0	0	4	0	0	1	0	0	0	0	0	0	1	0	39	4	49	79.5%

Tabulka 6.4: Úspěšnost a záměny rozpoznání jednotlivých znaků.

## 6.4 Celkové výsledky rozpoznání RZ

Konečné výstupy celého algoritmu, tedy jestli byla RZ rozpoznána nebo ne, případně do jaké míry, můžete vidět v tabulce 6.5. Po velmi dobrých výsledcích detekce dochází k velkému poklesu úspěšnosti, především u snímků datasetu D (vyfoceny z úhlu).

		Sada				Celkem
		A	B	C	D	
Počet RZ		41	110	22	44	217
Počet správně rozpoznaných znaků RZ	7	28 68,3%	74 67,3%	15 68,2%	8 18,2%	125 57,6%
	$\geq 4$	13 31,7%	28 25,5%	6 27,3%	22 50%	69 31,8%
	$< 4$	0 0%	8 7,3%	1 4,5%	14 31,8%	23 10,6%

Tabulka 6.5: Úspěšnost rozpoznání RZ.

Po analýze sady D docházím k závěru, že je to chyba především klasifikátorů písmen, protože právě dataset D je složen ze snímků s největším množstvím znaků A-Z. Registrační značky na těchto snímcích jsou také zkresleny úhlem, pod kterým jsou vyfoceny, na což nereaguje dostatečně přesně třetí část algoritmu – logika. Kombinace těchto drobných odchylek s častými chybami kaskádových klasifikátorů, které jsou páteří celého algoritmu, vytváří vysoké míry chybovosti.

Zbylou vlastností algoritmu, kterou je nutné analyzovat, je rychlost zpracování jednoho snímku. V kapitole 2 byl zvolen cíl 2 sekund. V tabulce 6.6 je vidět, že toho bylo docíleno minimálně na osobním počítači s procesorem Intel Core i3-2120 s frekvencí 3.30GHz a 8GB RAM. Rozdíly mezi jednotlivými sadami jsou způsobeny velikostí plochy, nad kterými je nutné hledat znaky. Ta je totiž u snímků vyfocených z větší vzdálenosti mírně menší.

	Sada				Celkem
	A	B	C	D	
Detekce oblastí	266,5	286,45	282,47	250,36	271,45
Rozpoznání znaků	1174,97	1172,77	984,43	995,77	1081,99
Logika – nalezení pozic	0,54	0,53	0,65	0,36	0,52
Celý systém	1442	1459,75	1267,56	1246,5	1353,95

Tabulka 6.6: Průměrná rychlost(ms) zpracování jednoho snímku jednotlivými částmi a celým systémem.

# Kapitola 7

## Závěr

Cílem této práce bylo vytvořit algoritmus pro detekci a rozpoznání RZ v obraze. Řešení jsem zaměřil na mobilní platformu, kde by umožňovalo jednoduchou a efektivní anotaci RZ. Po celou dobu vyvíjení jsem se držel předem daných priorit a podmínek a snažil se myslet na využití v praxi.

Při testování nad získanou testovací sadou o velikosti 224 snímků jsem overil funkčnost jednotlivých částí algoritmu. Detektor funguje velmi dobře s celkovou úspěšností 96,9%. Rozpoznání znaků není chybné, není ovšem dostatečně přesné. Klasifikátory některých znaků jednoduše nejsou kvalitní pro praktické použití. Logika pracuje správně, avšak je mírně nepřesná u RZ vyfocených z úhlu.

Z toho vyplývá, že implementovaná metoda je funkční, avšak není ještě dostatečně přesná pro přenos na mobilní platformu. Je potřeba výrazně zvětšit datovou sadu, aby bylo možné vytvořit kvalitnější klasifikátory, na kterých metoda závisí, a důsledněji také overit její funkčnost. S mírnými úpravami logiky a snížením množství falešných pozitivů by řešení bylo použitelné na mobilní platformě a mohlo by sloužit pro průběžné testování a zároveň rozšiřování datové sady. Poté by našlo využití u českých policistů, u kterých by sloužilo k automatické anotaci dokumentovaných vozidel.

# Literatura

- [1] Ahonen, T.; Hadid, A.; Pietikainen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Pattern Analysis and Machine Intelligence*, ročník 28, č. 12, 2006: s. 2037–2041, ISSN 0162-8828.
- [2] Anagnostopoulos, C.-N. E.; Anagnostopoulos, I. E.; Psoroulas, I. D.; aj.: License Plate Recognition From Still Images and Video Sequences: A Survey. *IEEE Intelligent Transportation Systems*, ročník 9, č. 3, 2008: s. 377–391, ISSN 1524-9050.
- [3] Bradski, G.: OpenCV. *Dr. Dobb's Journal of Software Tools*, 2000.
- [4] Canny, J.: A Computational Approach to Edge Detection. *IEEE Pattern Analysis and Machine Intelligence*, ročník PAMI-8, č. 6, 1986: s. 679–698, ISSN 0162-8828.
- [5] Dobrovolný, M.: Rychlý algoritmus rozpoznání registračních značek vozidel. *Perner's Contacts*, , č. 15, 2009: s. 35–46, ISSN 1801-674X.
- [6] Maini, R.; Aggarwal, H.: Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, ročník 3, č. 1, 2009: s. 1–11, ISSN 1985-2304.
- [7] Česká republika: Typy tabulek s registrační značkou platné dle 243/2001 Sb. 243/2001 [online]. <http://www.mdcz.cz/NR/rdonlyres/5C50406D-6CEE-4F03-BFD5-E29575BA674F/0/2regzna.pdf>, 2006, [cit. 2015-05-13].
- [8] Seo, N.: Tutorial: OpenCV haartraining [online]. <http://note.sonots.com/SciSoftware/haartraining.html>, 2008, [cit. 2015-05-13].
- [9] Smékal, D.: *Detekce státní poznávací značky pro dohledové systémy*. Diplomová práce, Brno: Vysoké učení technické. Fakulta elektrotechniky a komunikačních technologií, 2013.
- [10] Viola, P.; Jones, M.: Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference*, ročník 1, IEEE, 2001, ISBN 0-7695-1272-0, s. 511–518.
- [11] Wai-Yiu, H.; Chi-Man, P.: A Macao license plate recognition system based on edge and projection analysis. In *Industrial Informatics (INDIN), 8th IEEE International Conference*, IEEE, 2010, ISBN 978-1-4244-7298-7, s. 67–72.

- [12] Ying, W.; Yue, L.; Jingqi, Y.; aj.: An Algorithm for License Plate Recognition Applied to Intelligent Transportation System. *IEEE Intelligent Transportation Systems*, ročník 12, č. 3, 2011: s. 830–845, ISSN 1524-9050.

# Dodatek A

## Obsah CD

Seznam složek na přiloženém CD:

- *text* – technická zpráva v pdf a její zdrojové soubory
- *src* – testovací aplikace a její zdrojové soubory pro prostředí Qt
- *datasets* – pořízené datové sady
- *plakat*
- *video*