

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

DETEKCE A ROZPOZNÁNÍ SPZ AUTOMOBILŮ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ROMAN KOVAŘÍČEK

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

DETEKCE A ROZPOZNÁNÍ SPZ AUTOMOBILŮ

CAR LICENCE PLATE DETECTION AND RECOGNITION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ROMAN KOVAŘÍČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAN VÁŇA

BRNO 2011

Abstrakt

Tato bakalářská práce se zabývá nalezením státní poznávací značky v obraze a jejím rozpoznáním. Dále v krátkosti popisuje historii státních poznávacích značek. Také se věnuje současnému stavu poznávacích značek a jejich problematice. Rozebírá postup segmentace obrazu a následné vyhodnocování vybraných oblastí. Součástí práce je návrh a implementace algoritmů, které řeší samotné nalezení kandidátních oblastí nebo znaků. Konečným krokem je rozpoznání jednotlivých znaků a zobrazení výsledků s podrobnostmi uživateli.

Abstract

This bachelor thesis deals with finding the license plates in the image and pattern recognition. Work describes short history of the state license plates. It deals with also the current state license plates and their problems. It analyzes the process of image segmentation and follow evaluation of selected areas. A part of the work is design and implementation of algorithms that solve candidate search areas or characters. The final step is the recognition of individual characters and display the user with details of the result.

Klíčová slova

Rozpoznávání znaků, SPZ, státní poznávací značka, RZ, registrační značka, optické rozpoznávání znaků, rozpoznávání vzorů, segmentace

Keywords

Optical character recognition, car license plate, registration mark, OCR, pattern recognition, segmentation

Citace

Roman Kovaříček: Detekce a rozpoznání SPZ automobilů, bakalářská práce, Brno, FIT VUT v Brně, 2011

Detekce a rozpoznání SPZ automobilů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Váni.

.....

Roman Kovaříček

13. května 2011

Poděkování

Poděkování patří panu Ing. Janu Váňovi za cenné rady k vytváření odborné práce, a dále pak rodině za velkou podporu během studií.

© Roman Kovaříček, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Teoretický rozbor	4
2.1	Státní poznávací značka	4
2.1.1	Historie ve zkratce	4
2.1.2	Aktuálně používané SPZ a jejich problematika	5
2.2	Rozpoznávání vzorů v obraze	7
2.2.1	Snímání a digitalizace obrazu	7
2.2.2	Vzorkování (sampling)	8
2.2.3	Kvantování	9
2.2.4	Barevné modely	10
2.2.5	Předzpracování (preprocessing)	10
2.2.6	Segmentace obrazu	11
2.2.7	Metody segmentace	11
2.2.8	Úprava nalezených segmentů	15
2.2.9	Popis objektů a jejich klasifikace	17
3	Návrh a prostředí řešení	19
3.1	Pořízení obrazu a jeho předzpracování	20
3.1.1	Volba prahu a prahování	20
3.1.2	Segmentace obrazu na objekty	21
3.2	Nalezení znaků a jejich klasifikace	22
4	Implementace	23
4.1	Nejpoužívanější knihovní funkce a struktury	23
4.2	Segmentace a nalezení kandidátů	24
4.3	Úprava oblastí a rozpoznávání znaků	25
4.3.1	Vyhledání znaků	25
4.3.2	Úprava znaků	25
4.3.3	Rozpoznání znaků	26
4.4	Ovládání programu	27
5	Testování a výsledky	28
5.1	Metodika testování	28
5.2	Závěrečný test a výsledky	28
6	Závěr	32

A	Obsah DVD	34
B	Ukázka správného rozpoznání	35
C	Ukázka špatného rozpoznání	37

Kapitola 1

Úvod

Počítače jsou v dnešní době využívány téměř ve všech oborech lidské činnosti, počítač je vlastně elektronické zařízení, které zpracovává uživatelem zadaná data v připravených programech a výsledkem jsou výstupní data. Jedním z cílů počítače je uživateli usnadnit práci, jednou z takových činností je rozpoznávání v obraze, které je v počítačovém světě nedílnou součástí, je využíváno v mnoha aplikacích, které souvisí se zpracováním obrazu a detekcí objektů. Detekcí můžeme rozpoznat text, barvu pleti, celé osoby nebo jenom jejich obličeje a mnoho dalších objektů jako je například státní poznávací značka (dále SPZ). Mechanismus, sloužící k vyhledání SPZ v obraze, ulehčuje práci nejen policii, využívá se zde hlavně pro identifikaci vozidla páchajícího dopravní přestupek na vozovce, nebo k zachycení možného pachatele. Policie má k dispozici úplnou databázi všech SPZ a je tak možné vyhledat vlastníka vozidla.

Tato práce se tedy zabývá, jak bylo naznačeno, zpracováním obrazu automobilu, nalezením SPZ a následném převedení znaků do textové podoby. Vstupem je fotografie čelní nebo zadní části vozidla, výstupem je textová podoba SPZ vozidla na obrázku.

S krátkou historií, vzhledem a problematikou státních poznávacích značek se seznámíme v kapitole [2.1.1](#). Podíváme se na aktuálně používané podoby a způsoby umístění. Zpracování obrazu a jeho digitalizaci si popíšeme v kapitole číslo [2.2](#). Rozdělíme si vše do několika skupin zpracování a popíšeme si, co která znamená a jak je důležitá pro další vývoj aplikace. Mezi důležité podkapitoly patří segmentace [2.2.6](#), kde budeme obraz rozdělovat do kandidátních oblastí. Ty budou dále zpracovány, než určíme výslednou oblast, která obsahuje poznávací značku. Finální částí bude nalezení a rozpoznání znaků v označené značce.

V kapitole [3](#) si popíšeme návrh celé aplikace, jakými způsoby jsme postupovali při vyhledávání spz oblasti, vyhledávání znaků a jejich rozpoznávání. Poslední kapitolou bude testování [5](#), zde si popíšeme několik testů prováděných na naší aplikaci. Dosažené výsledky zhodnotíme a navrhneme možná vylepšení včetně využitelnosti aplikace.

Kapitola 2

Teoretický rozbor

2.1 Státní poznávací značka

Systém registračních značek je již v pořadí *sedmým* systémem používaným na našem území. Popíšeme si v krátkosti, jak vše začalo a jak se vyvíjela historie značek, nakonec se podíváme na stavbu SPZ a její problematiku. O problematice a historii poznávacích značek jsem čerpal z [9].

2.1.1 Historie ve zkratce

Naše země v roce 1906 spadala ještě pod Rakousko-Uhersko, tehdy byla zavedena první evidence motorových vozidel a vznikla první poznávací značka. První značky byly velice jednoduché, jak je vidět na obrázku 2.1. Povšimněme si bílého podkladu a černého písma. Böhmen (Čechy) bylo pod označením *O*, následovala číselná řada, která se zvětšovala s postupem času a přibýváním značek.

The image shows a historical license plate with the text 'OII 361'. The characters are black and set against a white background. The 'O' is a simple circle, and the 'II' consists of two vertical bars. The numbers '361' are also simple, with the '3' having a curved top and the '1' being a single vertical bar.

Obrázek 2.1: Příklad prvních SPZ v historii.

Značky procházeli různým vývojem. Vznik nového státu, rozdělení na státní celky nebo vyčerpání všech možných kombinací byly důvody ke změnám poznávacích značek a jejich zavedení systému. Po vzniku Československa roku 1918 pokračoval v zemích stejný systém, tedy rakouský.

Počet automobilů vzrůstal a bylo potřeba změnit systém, římské číslice již tvořili značky nepřehlednými. Roku 1932 vzniká první československá značka. Tabulky byly změněny na černý podklad a bílé písmo. V době druhé světové války byly říšské SPZ doplněny takzvaným "věčkem" a razítkem okresu v červené barvě. Po válce byl zachován předválečný stav značek s malými změnami.

Roku 1953 převzala pod správu evidenci vozidel (HS-VB) a vznikl nový druh tabulek. První písmena označovala typ vozidla a druhé okres.

V roce 1960 byli přerozděleny kraje a okresy a systém byl opět změněn. Od tohoto roku vycházejí značky již velice podobné nynějším. Jak vidíme na obrázku 2.2.

Přibližně od roku 1985 jsou vydávány takzvané "staré poznávací značky", které můžeme vidět ještě dnes.



Obrázek 2.2: SPZ vydávané v rozmezí 1960-2001.

2.1.2 Aktuálně používané SPZ a jejich problematika

O aktuálním stavu značek jsem čerpal z [9]. V roce 2001 byla agenda evidence vozidel předána pod ministerstvo dopravy. S převedením agendy došlo také ke změně systému. Prvním krokem je nahrazení názvu SPZ - státní poznávací značka na RZ - registrační značka, druhá větší změna byla ve změně kódování a to podle krajů. Tyto značky jsou vydávány dodnes, a jak můžeme vidět na obrázku 2.3 mají tyto značky určité charakteristické znaky.



Obrázek 2.3: SPZ - ukázka speciálních znaků (SME, STK, pravý horní roh).

Prolisovaný kruhový prostor uprostřed slouží k umístění značek měření emisí (ME) a technické kontroly (TK) vozidla. Dalším znakem je prázdné místo vlevo, kam se od roku 2004 (vstupu do Evropské unie) umísťuje modrý proužek se značkou EU a zkratkou státu - v našem případě CZ (obrázek 2.4).



Obrázek 2.4: Ukázka umístění modrého proužku s CZ.

Značky, kterými se budu v této práci zabývat, jsou na bílém podkladu vyznačeny černým písmem a určeny především pro silniční motorová vozidla a přípojná vozidla. Existuje spousta dalších druhů značek pro speciální vozidla, které by bylo možné do této práce implementovat jako rozšíření. V roce 1992 navíc přibyla možnost umístění jedné dvouřádkové a jedné jednořádkové značky na vozidlo. Značky se vyrábí v předem definovaném rozměru - 520x110 mm a modrý proužek, přidávaný od roku 2004 je široký 55 mm.

Registrační značka (dále jen RZ) se skládá z 5-7 písmen latinské abecedy (bez diakritiky) a několika arabských čísel. Abeceda je využita téměř celá až na znaky CH, G, O, Q, W, které by se mohli plést s jinými. Těmito pravidly vzniká dostatečné množství kombinací, které můžeme používat. Nemůžeme se tedy nikdy setkat se dvěma identickými značkama na dvou různých vozidlech. Znaky jsou ve stejných rozestupech od sebe a vtištěny stejným

druhem písma ve stejné velikosti.

Pro značky platí další určitá pravidla:

1. Vždy musí být umístěna alespoň jedna číslice a písmeno.
2. První písmeno zleva značí vždy kód kraje, z kterého auto pochází.
3. Písmena mohou být libovolně přeházená.

RZ je na vozidle umístěna, viz obrázek 2.5, spodní hranou dolů a přibližně vodorovně k vozovce. Jak stanovuje "Způsob umístění tabulky registrační značky na vozidle (Registr silničních vozidel - K §7 odst. 4 a §79 odst. 1 a 5 zákona z roku 2001)" RZ nesmí být ničím zakryta, nesmí být umístěny nápisy nebo vyobrazení, které by narušily její čitelnost nebo rozlišovací schopnost. Dále stanovuje, že tabulka musí být umístěna na přední a zadní části silničního motorového vozidla a to kolmo ke směru jízdy. Umístění je možné jenom uprostřed, pokud v technickém průkazu není zapsáno vlevo.



Obrázek 2.5: Ukázka umístění RZ.

RZ, neboli SPZ vydané do roku 2001 se liší hlavně strukturou písmen a číslic. Prvním rozdílem je počáteční skupina znaků. Dvě písmena označují okres a třetí sérii. Číslice jsou seskupeny po dvojicích, někdy oddělené pomlčkou. Uprostřed jsou prolisované značky pro SME a STK. Nové značky je možné rozeznat podle proužku EU vlevo, pokud ho značka nemá, lze se odkázat na pomlčky.

Souhrnné pravidla o RZ, které nám usnadní práci:

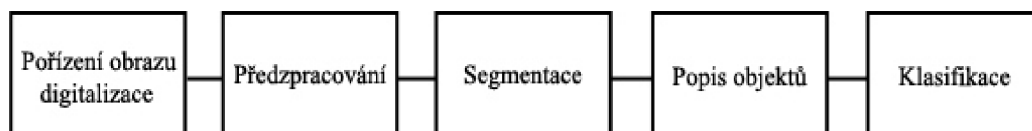
- pevně daná velikost
- stejný druh a velikost písma
- umístění na vozidle
- barva písma a podkladu

Někteří řidiči tyto pravidla však nerespektují a setkáme se v dalších kapitolách i se značkami skoro nečitelnými.

2.2 Rozpoznávání vzorů v obraze

O rozpoznávání vzorů v obraze a dalších podkapitolách jsem čerpal z literatury [5], [6] a [2].

Na fotografii můžeme vidět krajinu, detail rostliny nebo v našem případě fotografii vozidla s poznávací značkou. To vše je odrazem malého kousku reálného světa, neboli obrazovou informací, kterou se rozpoznávání v obraze zabývá. Rozpoznávat vzory nemusíme jenom v obraze. Samotný průběh a zpracování si rozdělíme na několik základních kroků, které jsou nezanedbatelné při zpracování obrazu. Jednotlivé kroky zpracování jsou:



Obrázek 2.6: Kroky rozpoznávání obrazu.

Všechny kroky si zde podrobněji popíšeme a ukážeme si jak jsou důležité k rozpoznávání.

2.2.1 Snímání a digitalizace obrazu

Tato kapitola se zabývá získáním obrazu, zachycení malého kousku reálného světa, a následném uložení do digitální podoby, která nám usnadňuje práci v počítači nebo jiném zařízení či systému. Pořizování obrazu můžeme provádět pomocí různých přístrojů více způsoby. Skener, digitální fotoaparát, kamera nebo družicový snímač dokáží provést digitalizaci obrazové funkce na jednotlivé prvky s různou hodnotou intenzity světla. Provádí se nejčastěji pomocí CCD prvků, které přenášejí elektrický náboj. Jsou tedy schopny převádět optickou veličinu na veličinu elektrickou, například u termokamery se jedná o převod teploty na elektrickou veličinu. Se všemi druhy obrazů je možné provádět laboratorní měření, vyhledávání objektů v obraze nebo převádění na jiné veličiny. Podle knihy [1] popisujeme snímání obrazu jako: "převod optické veličiny na elektrický signál, který je spojitý v čase i úrovni".

V případě digitálního fotoaparátu je záznamovým médiem fotocitlivý snímač. To je zařízení, které převádí světlo na elektrický náboj. Jinými slovy - čím jasnější je světlo, tím větší je náboj. Pořízení snímku může ovlivnit mnoho faktorů, které samotný proces komplikuje, nebo naopak usnadňuje. Některé z hlavních požadavků jsou: ostrost, dostatečné množství detailů, barevná vytíženost a velikost výsledného souboru. Většina policejních radarů pro pořizování snímku je umístěna staticky, například na mostě dálnice, ale přesto nepořizujeme fotografii statické scény, ale jedoucího automobilu. V této práci se nezabýváme fotkami z dálnic, velké dálky nebo bočních pohledů, ale statickou fotografií čelní nebo zadní části vozidla, přesto musíme počítat s některými vlivy, které vznikají při nedokonalém vyfotografování scény. Například špatný kontrast, rozmazání pohybem nebo přesahující náklad na vozidle, který zakrývá tabulku. Takto znehodnocené fotografie můžeme ovlivnit hlavně kvalitou snímacího zařízení a zručností obsluhou na zařízení.

Digitalizaci můžeme vysvětlit mnoha způsoby, každý tento převod analogového vzorku do digitálního popisuje jinými slovy. Převod například jedné textové strany rukopisu ze 14. století do digitální formy můžeme nazývat digitalizací. Není to však náhrada originálu,

jenom vytvoření něčeho nového v mnoha jiných i stejných vlastnotech. Digitalizovat nemusíme jenom statický obraz, ale totéž platí o zvuku, pohyblivém obraze a například také o psaném textu. Podle [2] je digitalizace definována jako:

“Převod vybraných měřitelných fyzikálních veličin digitalizovaného objektu do numerických hodnot, jejich kódování a uložení za účelem pozdějšího vygenerování jiných fyzikálních veličin s cílem umožnit pozdějšímu uživateli fyziologické vjemy nahrazující přímé vnímání originálu.”

Samotnou činností digitalizace rozumíme vzorkování obrazu v matici $M \times N$ bodů a kvantizaci K úrovní. Důležitým krokem je zvolení konstant M, N a K . Pokud zvolíme nízké hodnoty bude rozlišení obrazu příliš malé a ztratíme tím obrazové informace. Naopak zvolíme-li vysoké hodnoty, zachováme mnoho informací o obrazu, ale vzroste jeho rozlišení, tudíž i jeho velikost. Důležitými pojmy jsou: velikost obrazu udávaná v bodech - pixely a rozlišení obrazu v DPI (dots per inch).

Celý proces bychom mohli rozdělit do třech kroků: *vzorkování (sampling), kvantování a kódování.*

2.2.2 Vzorkování (sampling)

Při vzorkování vycházíme z Shannonova teorému, který přesně říká [2]:

“Přesná rekonstrukce spojitého, frekvenčně omezeného, signálu z jeho vzorků je možná tehdy, pokud byl vzorkován frekvencí alespoň dvakrát vyšší, než je maximální frekvence rekonstruovaného signálu.”

Zjišťujeme tedy vzorky v pravidelných intervalech a na základě jednoho čísla přiřadíme hodnotu celé oblasti (ploše vybraného pixelu). Při vzorkování se tedy musíme řídit Shannonovým teorémem, který ve volném překladu říká, že vzorkovací interval musí být větší než dvojnásobek nejvyšší frekvence v signálu. Pokud nebude dostatečně vysoká frekvence, dochází k takzvanému aliasu a tudíž ke ztrátě informace. Pro předejití aliasu se používá antialiasingový filtr, který se umístí před převodník a filtruje vyšší frekvence než odpovídají Shannonovu teorému - antialiasing. Je také potřeba zvolit vhodnou vzorkovací mřížku. Nejvíce používané jsou hexagonální (viz obr. 4) a čtvercové (viz obr. 5), třetí možností jsou rovnostranné trojúhelníky. Vzorkovací mřížka je plošné uspořádání bodů při procesu vzorkování.



Obrázek 2.7: Čtvercová a hexagonální mřížka využívaná ve vzorkování.

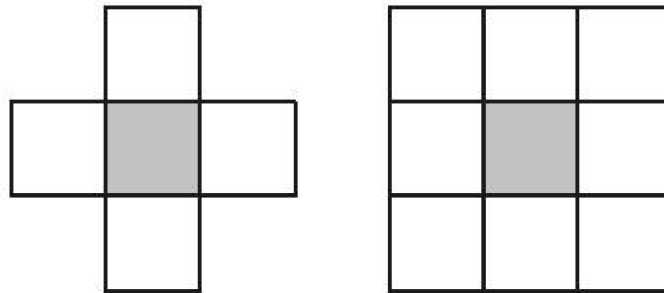
Čtvercová mřížka je snadno realizovatelná, má však nevýhody v měření vzdálenosti a spojitosti objektů. Většinu z těchto problémů řeší hexagonální tvar, který například není vhodný pro Fourierovu transformaci, naopak výborný pro spojitost objektů.

Jednomu vzorku, který získáme z dat, odpovídá nejmenší a již nedělitelný bod - pixel (jedno políčko ve vzorkovací mřížce), který nese obrazovou informaci.

Další vlastnost nadefinujeme jako vzdálenost dvou pixelů, kterou můžeme vyjádřit Euklidovským vztahem pro výpočet vzdálenosti dvou bodů se známými souřadnicemi $x = (x_1, x_2), y = (y_1, y_2)$.

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (2.1)$$

Pokud bychom uvažovali o čtvercové mřížce, lze definovat takzvané sousední body. Jedná se o 4-sousedství a 8-sousedství (viz obr 2.2.2). Jejich vzdálenost od sousedních pixelů je vždy stejná velikost - $D_4 = D_8 = 1$.



Obrázek 2.8: Ukázka 4-sousedství a 8-sousedství.

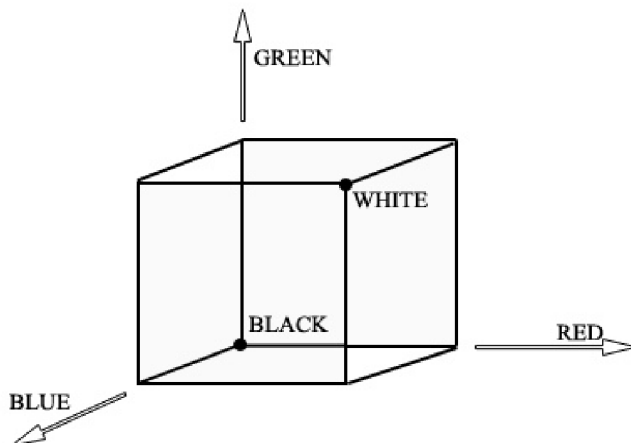
2.2.3 Kvantování

Kvantováním rozumíme přiřazení určité diskrétní hodnoty jasu danému bodu pro zvolený interval. Většinou průměrujeme hodnotu celého intervalu z oboru hodnot. Pokud nebude počet kvantovacích úrovní dostatečně velký, budou na výsledném obraze hrubé přechody. Při kvantování dochází ke ztrátě informací, chybě, které říkáme kvantizační. Chyba se projevuje hlavně v místech, kde rychle přecházíme z odstínu do odstínu. Tuto chybu můžeme částečně eliminovat volbou intervalu. Používá se kvantování **uniformní** - volí konstantní délku intervalu a **neuniformní** - volí proměnnou délku intervalu. Člověk je schopen rozpoznat až 50 úrovní přechodu jasu, měli bychom se snažit, aby kvantizační chyba nedosahovala menších hodnot.

kódování - Signál, který byl kvantovaný se dále kóduje, jednotlivým úrovním (intervalům) se přiřadí binární číslo. Kódování je vlastně část kvantování. Abychom mohli určit přesný odstín, použijeme jeden z již vytvořených barevných modelů, který se liší jenom barevnými složkami.

2.2.4 Barevné modely

Nejpoužívanější barevný model vychází ze tří primárních barev *Red - Green - Blue* (červená, zelená, modrá), viz 2.9. Tento model je aditivní a využívá se hlavně u obrazovek nebo monitorů, zde se jeden bod kombinuje sloučením těchto třech signálů a tvoří jednu z barev. Model lze vyjádřit pomocí jednotkové krychle, kde v počátku $[0,0,0]$ leží černá barva a v protějším rohu bílá barva. V počítačové technice se používá dělení na 256 barev v intervalu $\langle 0 - 1 \rangle$. Což při třech barvách znamená $3 \times 8b = 24b$. Označení pro 24bit barvy je *true colors*, obraz potom může mít 2^{24} barev.



Obrázek 2.9: RGB model.

Mezi další modely bychom mohli zařadit CMY, CMYK, nebo HSV, HLS využívané hlavně v tiskařství. Nebudeme si je dále popisovat, nejsou pro tuto práci důležité.

Grayscale - stupně šedi

Model stupně šedi reprezentuje dvě barvy a to černou a bílou. Možných kombinací je 256, což nám odpovídá 8 bitovému vzorkování. Každý bod - pixel má hodnotu jasu v rozsahu 0 (černá) - 255 (bílá) a nebo lze model zapsat také jako krytí černé barvy, kde 0 % je bílá a 100 % je černá. Přepočítání z plně hodnotného barevného obrázku (RGB) na šedý lze provést převedením všech barev na stupně šedi. Jasová hodnota se potom vypočítá jako:

$$jas = 0.299 * R + 0.587 * G + 0.114 * B \quad (2.2)$$

2.2.5 Předzpracování (preprocessing)

Čerpáno z [8] a [2]. Cílem předzpracování je potlačit nežádoucí jevy, které mohli vzniknout během snímání objektu, přenosu dat nebo digitalizaci. Šum, zkreslení, světelné podmínky a mnoho dalších vlivů, které se snažíme potlačit. Vlivy nemusíme jenom tlumit, ale také naopak je můžeme zvýrazňovat pro účely dalšího zpracování. Aplikujeme mnohé filtry pro zostření obrazu, potlačení šumu či zkreslení. Nebo používáme jednodušší operace, jako jsou: přesuny, ořezávání obrazu, zvětšování nebo zmenšování. Přibližné postupy zpracování

obrazu jsou v tomto pořadí: jasové, geometrické transformace, lokální předzpracování (šum, zkreslení, ...) a konečná restaurace obrazu. Vstupem i výstupem je stále náš obraz, který není nově vytvořenou informací.

V této práci bylo předzpracování využito pouze v podobě adaptivního prahování, kterému se věnuji v následující kapitole.

2.2.6 Segmentace obrazu

O segmentaci a jejích metodách jsem čerpal z [4], [1] a [8].

Segmentace obrazu je v této práci nejdůležitější, nejsložitější a také nejtěžší částí při celkovém zpracovávání a analýze dat z obrazu. Snažíme se rozčlenit obraz do částí, které odpovídají předmětům, oblastím v reálném světě. Jedná se o takovou analýzu, která nalezne objekty v obraze a výsledkem je soubor oblastí, které odpovídají objektům v obraze. Takovým objektem je v našem případě registrační značka automobilu. Segmentaci lze definovat jako [3]:

Segmentace obrazu $f(x, y)$ je jeho dělení na podobrazy R_1, R_2, \dots, R_n tak, že podobrazy splňují následující kritéria:

1. $U_{i=1}^n R_i = f(x, y)$,
2. $R_i \cap R_j = \emptyset, i \neq j$,
3. Každý podobraz splňuje nějaké tvrzení, popř. množinu tvrzení, např:
 - Všechny pixely v podobraze R_i mají stejnou úroveň šedi.
 - Všechny pixely v podobraze R_i se neliší v úrovni šedi více než o předepsanou hodnotu.
 - Standardní odchylka úrovně šedi všech pixelů podobraze R_i je dostatečně malá, apod.

2.2.7 Metody segmentace

A) Statické metody - mezi statické metody řadíme například shlukovou analýzu (podobnost vlastností pixelů) nebo například Kohonenovy mapy využívající soutěžní metodu učení. Mezi nejznámější ovšem patří prahování, které je v této práci využíváno.

Prahování (thresholding) tato metoda je jedna z nejstarších a také nejjednodušších metod segmentace obrazu. Je to převod obrázku ze stupní šedi do binární podoby, černobílého obrazu, kde každý pixel obsahuje buď barvu bílou a nebo černou. Metoda je založena na hodnocení jasové složky každého pixelu. Principem je najít práh, kde jeho hodnota jasu oddělí určitou část od pozadí (zbytku obrazu).

Lze ji tedy zapsat jako transformaci vstupního obrazu f na segmentovaný obraz g s T (předem stanovenou hodnotou neboli prahem):

$$g(x, y) = \begin{cases} 255, & f(x, y) \geq T \\ 0, & f(x, y) < T \end{cases} \quad (2.3)$$

Bodům s hodnotou přesahující určený práh je přiřazena hodnota 255 (bílá barva) a bodům menším než práh 0 (černá barva). Jakou barvu budeme přiřazovat oblasti pozadí

a popředí záleží na nás a na tom jaká oblast nás zajímá v dalším hledání. Prahování lze rozdělit na globální a lokální.



Obrázek 2.10: Rozdíl mezi globálním a lokálním prahováním.

- **Globální prahování** hledá hodnotu prahu T a ta je aplikována na celý vstupní obraz. Zde nastávají problémy při překrývání jasových složek v jednotlivých částech vstupu. Proto je na nerovnoměrně osvětlené obrazy tento postup zcela nepoužitelný, mohlo by se stát, že oblasti budou chybně zpracovány.
- **Lokální prahování** řeší problém nevyváženosti jasových složek v obrázku, pracuje s proměnlivým prahem. Zde hodnota T je dynamicky měněna neboli adaptována pro různé části obrazu.

Důležitým faktorem je správně zvolený práh, tuto hodnotu můžeme stanovovat několika způsoby. Buď pokusy a nebo metodou s automatickým stanovením hodnoty. Automatické určení prahu je v tomto projektu výhodné, protože kdybychom nechali volit uživatele, bylo by to časově náročné.

Zvolení hodnoty prahu je tedy možné udělat několika způsoby, prvním z nich je **experimentální**. Pokud zpracováváme stále stejné obrazy, zkusíme dosazovat různé hodnoty prahu až po nám nejlépe vyhovující výsledek. Další možností je odhadnutí plochy, neboli **procentní**. Pokud víme že v obraze tvoří bílá plocha 30 % zvolíme prahovou hodnotu tak, aby 30 % mělo úroveň jasu menší než T . Nám nejlépe vyhovující metodou je **získání hodnoty z histogramu**, histogramům se věnujeme v kapitole 2.2.8. Z grafu, který nám udává četnost výskytu jednotlivých jasových bodů v obraze, nalezneme hodnotu lokálního minima mezi dvěma maximy.

Mezi výhody prahování patří bezesporu jednoduchost, snadná implementace a malá časová náročnost. Mezi nevýhody zase volba prahu, této části je potřeba se věnovat více.

B) Detekce hran (edge detection) - další metodou, kterou můžeme zařadit mezi segmentovací, jsou detektory hran. Jako hrany můžeme označovat místa, kde body mění hodnotu jasu, nebo kde vzniká přechod mezi objekty. Rozlišujeme potom různé typy hran - skoková, šikmá, čára a střecha. Proces detekce hran rozdělujeme na tři části: filtrování, diferenciace a detekce. Filtrování nám odstraňuje nežádoucí vlivy jako šum, rozmáznutí obrazu a podobné chyby, diferenciací zvýrazníme oblasti v obraze (změny jasu) a detekcí rozpoznáme nejvýznamější oblasti. Hranových detektorů, jako metod existuje velké množství, lze je rozdělit do dvou skupin, metody využívající první derivaci (hledání maxim) a metody upřednostňující druhou derivaci (hledání průchodů nulou).

- **První derivace** - počítáme rozdíl okolních pixelů obrazu, změnu připadající na jednotku délky. Z derivace vyhledáváme maximum, kde je námi hledaná hrana. Výsledná derivace je aproximována pomocí konvoluce (zpracování dvou funkcí) s vhodným jádrem. Mezi nejpoužívanější jádra patří: Robertsův, Sobelův, Robinsonův, Kirschův, Cannyho operátor nebo operátor Prewittové. Hlavní nevýhodou je zde závislost na velikosti objektu a zvolené masky nebo velké citlivosti na šum.

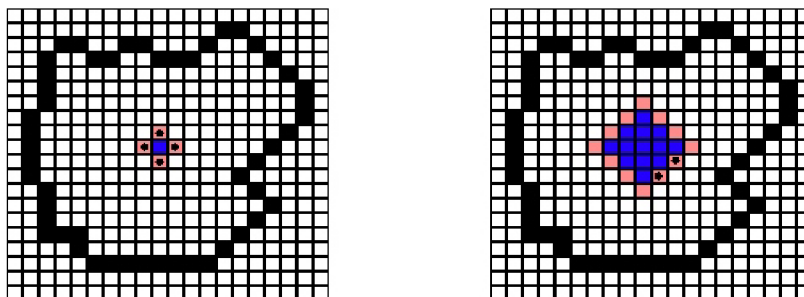
- **Druhá derivace** - hledání průchodu nulou je mnohem jednodušší než vyhledávání maximum, avšak hledáme čistý průchod, ne však hodnotu blížící se nule. Tyto metody dosahují mnohem lepších výsledků než předchozí, které některé námi hledané hrany vůbec neoznačí. Metody pracují s Laplaceovým operátorem, jejich hlavní nevýhodou je ztráta ostrých rohů a nebo spojování do uzavřených smyček.

Dalšími parametry, které můžeme z výsledků získat je směr a síla hrany. Mezi další možné detektory hran bychom mohli uvést například Houghovy transformace, nebo aktivní kontury. Detektorů hran je velké množství a například poslední dvě uvedené jsou časově náročnější na implementaci a i na výpočet než metody založené na hledání derivací.

C) Regiony v obraze (region-based methods) - vše je založeno na zjišťování podobnosti pixelů v nějaké vlastnosti, kterou může být jas, barva, nebo statické okolí pixelu. Tato metoda je výhodnější pro těžko rozpoznatelné obrazy pomocí detektorů hran z důvodu přílišného šumu. Existuje mnoho metod, uvedeme si proto dvě neznámější a to:

- **Rozrůstání oblastí** - založeno na spojování homogeních oblastí. Na počátku definujeme takzvané startovací body (semínka) v obraze, které nám splňují požadované vstupní podmínky. Postupně budeme brát body v okolí semínka (v jeho okolí) a stanovené kritérium nám bude určovat zda pixel přiřadit do oblasti a nebo jej nechat v pozadí. Z popisu můžeme vidět že se jedná o rekurzivní algoritmus, který je implementován ve většině případů pomocí zásobníku. Implementace algoritmu je jednoduchá a rychlá, bohužel vyplňování pomocí semínka nepatří mezi nejrychlejší algoritmy ani mezi nejefektivnější. Body, které byly již testovány jsou zařazovány opět do zásobníku a testovány znovu. Tomu se dá několika úpravami v kódu zabránit a metodu tak urychlit. Určení sousedů, neboli těch semínek které budou do zásobníku vkládány, je důležitým parametrem. Můžeme zvolit 4-sousedství a nebo 8-sousedství viz obrázek [2.2.2](#).

Na obrázku [2.11](#) vidíme jak postupně se rozrůstá oblast, která je započata startovacím bodem a čtyřokolím je vyplňována další oblast, takto by se pokračovalo až do vyplnění po černý okraj. Vše se dá implementovat jednoduchým algoritmem.



Obrázek 2.11: Ukázka semínkového vyplňování. a) startovací semínko b) zachycen třetí krok rozrůstání.

Algoritmus se zásobníkem :

1. Vlož bod $[x,y]$ do zásobníku.
2. Je-li fronta prázdná, tak ukonči, jinak :
 - (a) Vyber semínko z fronty a obarvi jej.
 - (b) Pro všechny sousední body $[x + 1, y], [x, y + 1], [x - 1, y], [x, y - 1]$ znovu zavolej tento algoritmus.

Rekurzivní algoritmus :

1. Otestuj bod $[x,y]$ zda netvoří hranici a nebo je již vyplněný.
 - (a) Pokud ANO, ukonči.
 - (b) Pokud NE, vybarvy bod.
2. Pro všechny sousední body $[x + 1, y], [x, y + 1], [x - 1, y], [x, y - 1]$ znovu zavolej tento algoritmus

Jednou z metod semínkového vyplňování je možnost vypňování řádkově, kdy je vyplňován celý řádek najednou až po hranici a vkládaných bodů do zásobníku je méně.

- Rozdělování a spojování oblastí - principiálně opačný přístup než předchozí metoda. Na začátku je definována jediná oblast (vstupní obraz), ten je dále dělen na menší podoblasti. Nejprve na čtyři kvadranty a v každém kvadrantu je měřena homogenita, podle té se rozhodne zda čtverec dále dělit a nebo ne. Každou oblast která nesplňuje homogenitu rozdělíme na čtyři podoblasti a takto postupujeme dokud není splněno konečné kritérium. Měřením stejnosti oblastí se zabývají další metody.

B) Hybridní metody - nazýváme hybridní pro jejich využívání různých prvků předchozích metod. V krátkosti si řekneme, které metody sem řadíme, protože v této práci nebyly vůbec využity. Jsou to tedy například:

- Watershed Transform (povodí) - je založeno na myšlence, že obraz je topografická mapa, kterou postupně zaplavuje voda. Z počátečních bodů je "voda" vypuštěna a tam, kde by se mohla slít je vytvořena hranice (hráz). Algoritmus je ukončen tehdy, pokud

dosáhneme maxima obrazu. Výsledkem je obraz rozdělený na několik oblastí. Výhodou zde je osvědčený postup a efektivita, nevýhodou je na vstupu zašumělý obraz, který nám vytvoří velké množství oblastí.

- **Neuronové sítě** - základem je zde umělý neuron, který napodobuje neuron v lidském mozku. Tato metoda není založena, jako ostatní, na předem známých kritériích o dané oblasti. V neuronových sítích probíhá takzvané trénování. Existují dvě strategie - učení bez učitele a učení s učitelem. Detailní popisy neuronových sítí v segmentaci obrazu naleznete například v knize [10].

2.2.8 Úprava nalezených segmentů

Čerpáno z [11]. V této části se pokusíme o úpravu již získaných segmentů z předešlé kapitoly. Různé oblasti, které jsme získali mají definovanou velikost, další možné vlastnosti lze získat například pomocí provedení vertikálního nebo horizontálního histogramu, nebo různých početních ukonů. Tyto informace nám dokáží posloužit pro další možné zpracovávání získaných oblastí. Nejprve si vysvětlíme co je histogram a jeho možnosti, dále se podíváme na možné úpravy oblastí, které nám vylepší jejich čitelnost. Tyto metody a algoritmy se využívají i u předzpracování obrazu.

Histogram - nám říká jak velké zastoupení jednotlivých barev je v daném obraze [2]. Toto platí pro barevný obrázek, v případě odstínů šedé se jedná o rozmezí bílé a černé barvy. Histogram může být proveden z horizontální nebo vertikální strany obrazu. Jejich využití je široké, používají se v digitálních fotografiích k určení jasu ve scéně nebo přímo v fotoaparátech. Je také součástí grafických programů na úpravu fotografií, a hlavně se používá při rozeznávání objektů v obraze. Umožňuje také vhodnou volbu prahu pro již zmiňované prahování. Pro jednorozměrný histogram, kde m je počet sloupců a n počet řádků, lze napsat vzorec.

$$\sum_{i=1}^{max} H(i) = m * n \quad (2.4)$$

Jeho hlavní nevýhodou je že nerozpoznává prostorové informace v obraze, tedy pro velký černý čtverec na bílém podkladě bude histogram stejný jako pro pět malých. Stejná nevýhoda platí pro pozici barvy. Ukázka histogramu na obrázku 2.12.

Tento histogram z poznávací značky nám říká, jaké zastoupení jednotlivých jasových složek od bílé do černé je v obraze. Zde vidíme že vysoké hodnoty, tedy bílé barvy a jí velice blízké odstíny, nejsou vůbec zastoupeny, stejně tak čistá černá barva. Můžeme říci, že snímek je tvořen šedými tóny a postrádá kontrast.

Z jiných histogramů můžeme poznat že jsou například podexponované a nebo vykazují přexpozici, případně vysoký kontrast scény. Možnost úpravy velké přexpozice a podexpozice je pomocí histogramu nemožná. Upravením kontrastu obrazu můžeme dosáhnout lepšího rozpoznávání. Slouží nám na to různé nástroje, které se používají již v předzpracování. My si zde ukáže možnost ekvalizace, která nám daný obraz s pomocí histogramu upraví.



Obrázek 2.12: Histogram - zastoupení jednotlivých jasových složek v obraze SPZ.

Ekvalizace - je metoda, která upravuje kontrast obrazu s využíváním histogramu. Mění rozložení intenzit v obraze tak, aby se vyskytovaly v širokém rozmezí. Obraz s celkovým vyšším kontrastem dokáže zvýraznit details, které mají naopak malý kontrast.

Výpočet ekvalizace není náročný a celá metoda se řadí mezi jednodušší. Podívejme se nyní na odvození vzorce pro výpočet [7] :

Vstup: mějme tedy interval jasů ve vstupním obraze $\langle p_0, p_k \rangle$ a histogram $H(p)$

Cíl: cílem je najít monotóní jasovou transformaci $q = \mathcal{T}(p)$, tak aby celý výstupní jasový interval $\langle g_0, g_k \rangle$ byl rovnoměrný pro nový histogram $G(q)$

$$\sum_{i=0}^k G(q_i) = \sum_{i=0}^k H(p_i) \quad (2.5)$$

Ekvalizovaný histogram \approx rovnoměrnému rozdělení.

$$f = \frac{N^2}{q_k - q_0} \quad (2.6)$$

Pro N řádků a N sloupců a žádaný histogram $G(q)$ je konstantní hustota pravděpodobnosti f . Ideálně rovnoměrný histogram získáme pouze pro ideálně spojitý případ, dosadíme tedy do rovnice 2.5.

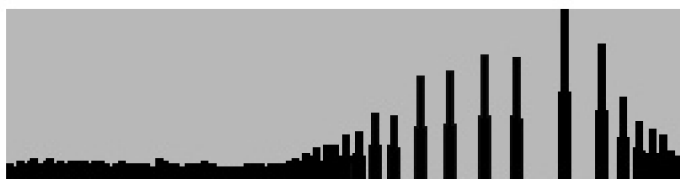
$$N^2 \int_{q_0}^q \frac{1}{q_k - q_0} ds = \int_{p_0}^p H(s) ds \quad (2.7)$$

Po vypočítání dostaneme tedy jasovou transformaci.

$$q = \mathcal{T}(p) = \left(\frac{q_k - q_0}{N^2} \right) * \int_{p_0}^p H(s) ds + q_0 \quad (2.8)$$

Integrál ve vzorci 2.8 se nazývá kumulativní histogram. Ten nám říká že součet hodnot menších nebo rovno p normálního histogramu je každá p -tá položka v kumulativním histogramu. Lze tedy přepsat 2.8 do tvaru z 2.7.

$$q = \mathcal{T}(p) = \left(\frac{q_k - q_0}{N^2} \right) * \sum_{i=p_0}^p H(i) + q_0 \quad (2.9)$$



Obrázek 2.13: Histogram po ekvalizaci.

Z výsledného vzorce vidíme že výsledný ekvalizovaný histogram použije ke svému výpočtu rozměr obrazu $N * N$, interval požadovaných úrovní ve výsledném obraze a původní histogram.

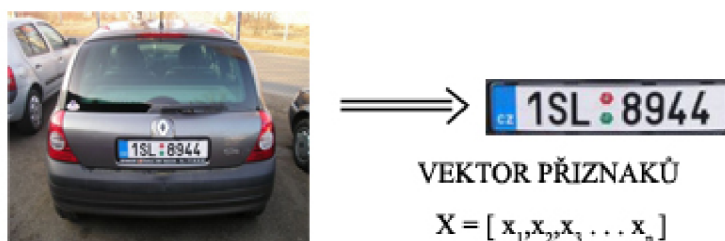
Jak tedy vypadá ekvalizace ve výsledku vidíme na obrázku 2.13, kde jsme použili obrázek uvedený pro histogram normální 2.12.

2.2.9 Popis objektů a jejich klasifikace

Popis nalezených objektů z předešlé segmentace a následné úpravy můžeme provést dvěma základními způsoby.

Prvním je *kvalitativní popis* - popisů existuje více. Narozdíl od kvantitativního zahrnují vlastnosti objektu, tak i jeho strukturu. Celý nalezený objekt je popsán pomocí primitiv a jejich vztahů mezi sebou. Primitivum definujeme jako základní popisný element, který dále slouží k hierarchickému popisu celé části.

Druhým je *kvantitativní popis* - je založen na příznacích. Ty dále sjednocujeme do vektoru příznaků, který nám slouží jako celkový výčet a popis nalezeného objektu. Ukazujeme si tedy vlastnosti objektu, ne jeho strukturu. Využívá se hlavně u statického rozpoznávání. (ukázka na obrázku 2.14). Mezi možné příznaky pro účely rozpoznání RZ (SPZ) můžeme



Obrázek 2.14: Ukázka kvantitativního přístupu a příznaky nalezené SPZ.

zařadit například :

1. x_1 ... velikost stran SPZky $N * M$
2. x_2 ... poměr stran $M : N$
3. x_3 ... poměr bílé a černé barvy v objektu
- ⋮

4. x_n ... vertikální a horizontální histogram

Posledním krokem při zpracování obrazu je klasifikace. Jedná se o zařazení objektů, které jsme našli v obraze do skupiny předem známých tříd.

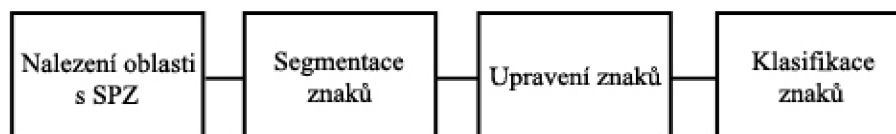
Časy, kdy se tištěné texty museli ručně přepisovat jsou dávno pryč a dneska existuje mnoho nástrojů, které v tomto směru usnadňují lidem práci. OCR (anglicky Optical Character Recognition) neboli optické rozpoznávání znaků je metoda, která převádí psaný, tištěný text do elektronické podoby. Nástrojů a programů je na internetu dostupných velké množství. U každého nástroje záleží v jakém fontu je text nebo v jaké velikosti, zda je text psaný ručně a nebo tištěný tiskárnou. Vytvoření OCR programu, který bude úspěšně rozpoznávat všechna čísla a písmena abecedy, není proto vůbec jednoduché. Při rozpoznávání hraje velkou roli mnoho aspektů, kterými může být samotná příprava obrazu s textem, segmentace jednotlivých znaků, metoda která bude nasazena na samotné rozpoznání.

Software, který je dostupný na internetu (volně, nebo placený) překračuje hranici 95% správně rozpoznávaných znaků, mnohdy se dostává až na 99%. Některé aplikace nabízí uživateli opravu špatně určeného znaku. OCR je označováno za offline rozpoznávání, existuje totiž také dynamické neboli online, které umí určit dynamický pohyb během písma. V této práci využijeme offline přístupu.

Kapitola 3

Návrh a prostředí řešení

Rozhodnutí jakou cestou se vydat za požadovanými výsledky je důležité pro konečný výsledek. Někdy je překážkou čas, jindy zase náročnost a složitost problematiky. V této práci bylo jasnou volbou využití knihovny OpenCV (Open Source Computer Vision Library) ve verzi 2.2. Knihovna má důležité vlastnosti, je svobodná, otevřená, multiplatformní a hlavně je přímo zaměřena na počítačové vidění a zpracování obrazu. Jazyk k implementaci dané práce je jazyk C/C++. Další výhodou knihovny a celé práce je přenositelnost mezi operačními systémy. Celý vývoj probíhal v prostředí GNU/Linux.



Obrázek 3.1: Zjednodušený náčrt přípravy práce.

Výběrat pro (3.1) práci můžeme z mnoha možností metod a algoritmů. Každá metoda má svoje výhody i nevýhody, zde dávám většinou přednost jednoduchosti a rychlosti implementace.

Výběr snímků k testování a k předvedení výsledků, případně obrázků, které naleznete u programu, bylo prováděno z internetových autobazarů.

Před implementací jsem si stanovil předběžné cíle, kterých by bylo dobré dosáhnout pro dobrou ukázkou a experimentování v následující kapitole.

1. Otestování alespoň dvou metod segmentace, výběr a implementace jedné z nich.
2. Dosažení výsledku správně nalezené SPZ oblasti v obraze - alespoň v 85% případů.
3. Dosažení výsledku nalezení znaků v SPZ alespoň v 90% případů.
4. Konečné rozpoznání jednotlivých znaků alespoň nad hranici 60%.
5. Možnost ukládání SPZ výřezu a zobrazení podrobných výpisů programu (parametry).
6. Porovnání námi vytvořeného OCR s jinou již dostupnou open-source aplikací.

3.1 Pořízení obrazu a jeho předzpracování

Obrázky k prvotnímu testování byly většinou nalezeny v internetových autobazarech, k dispozici jsem měl také přeexponované, tmavé nebo šedé fotografie. Vzorky byly také pořizovány pomocí fotografu FUJIFILM FINEPIX Z100 fd s rozlišením 8 Mpix.

Obrázky je potřeba nejprve připravit pro další kroky, které nás čekají. Po experimentech jsem zvolil nevyužití funkce rozmazání obrazu, kterou knihovna OpenCV nabízí pro odtránění šumu z obrazu. Výsledky, které jsem po následném prahování dostával nebyly lepší než při nevyužití této funkce.

3.1.1 Volba prahu a prahování

Prvním krokem je načtení obrázku do připravené struktury, informace o obraze bude potřeba zaznamenávat pro další využívání. U načítání obrázku ze souboru využijeme možnosti nahrání již v šedé škále barev. Obrázky větší jako 800 * 600 px budeme zmenšovat na velikost, která bude vyhovovat zobrazení na monitoru. Informace o zmenšeném obraze opět zaznamenáme pro pozdější přepočítávání souřadnic.

Nejvýhodnější bude zvolit adaptivní prahování. K metodě prahování je potřeba určit práh, který nám oddělí pozadí a popředí v obraze. Z obrazu tedy vytvoříme histogram, který se rozdělí na dvě části. Jedna část bude obsahovat jas s nejmenší četností a druhá jas s největší četností. V programu využijeme hodnoty prahu s největší četností jasu. Po zvolení prahu následuje samotné prahování. Adaptivního prahování využijeme i v dalších částech programu.



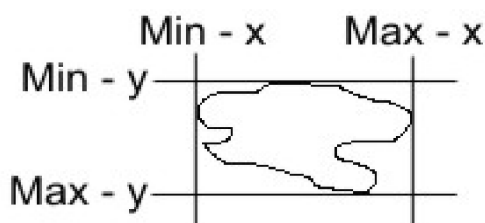
Obrázek 3.2: Ukázka prahovaného obrázku, originál je na obrázku 2.14.

Takto upravená data obrázku budeme ukládat do pole pro lepší práci v dalších částech programu. Originál obrázku si vždy uložíme a používáme kopii.

3.1.2 Segmentace obrazu na objekty

Segmentace obrazu v našem případě je rozdělení na jednotlivé oblasti, které by mohli být naší hledanou SPZ. V programu se zabýváme auty, která jsou vyfocena z přední nebo zadní části v přibližné vzdálenosti 5m.

Po prahování jsme dostali obraz, který je pouze černobílý, barva černá je reprezentována číslem 0 a bílá 255. Na vyhledání vhodných oblastí je použijeme procházení řádků obrazů, dokud nenarazíme na X pixelů za sebou, které jsou bílé. Tento bod dále předáme funkci, která pomocí upraveného semínkového vyplňování zařadí vyhledanou oblast mezi kandidáty a nebo ji vyloučí z hledání. Semínkové vyplňování tedy upravíme, tak aby zaznamenávalo nejkrajnější body na osách X a Y . Budeme ukládat $nejvetsi = [x, y]$ a $nejmensi = [x, y]$ z dané oblasti, jak je vidět na obrázku 3.3. Na implementaci vyplňování použijeme metodu se zásobníkem.



Obrázek 3.3: Ukázka použití souřadnic u nalezených kandidátních oblastí.

Zaznamenanou oblast pomocí souřadnic dále podrobíme testům, zda bude označena za možného kandidáta. Prvním z testů je poměr počtu bílých a černých pixelů v oblasti. Zde je nastavíme přibližné rozmezí mezi 35% až 80%. Vyloučíme oblasti, které jsou příliš bílé, nebo naopak skoro celé černé. Dalším kritériem je poměr stran. Vypočítané strany ze souřadnic by měli být mezi poměrem maximálně 1:8 a minimálně 1:3. Toto kritérium nám vyloučí oblasti naprosto nevyhovující poměru stran. Posledním testem je velikost jednotlivých stran, neboli maximální a minimální přípustná délka v obrazových bodech. Oblast, která splňuje všechna kritéria je zařazena do zásobníku kandidátů SPZ.

Pokud se stane, že v zásobníku je více než jedna oblast je provedeme nalezení nejbližší podobné a ta je dále považována jako výsledná SPZ oblast. Toto vybírání můžeme implementovat pomocí šablony. Šablonu vytvoříme z 10 správně nalezených oblastí. Absolutní hodnota rozdílu mezi šablonou a vstupní oblastí udává číslo odlišnosti, kde nejmenší hodnota je výsledek. Při testování uvidíme zda tato metoda bude mít smysl.

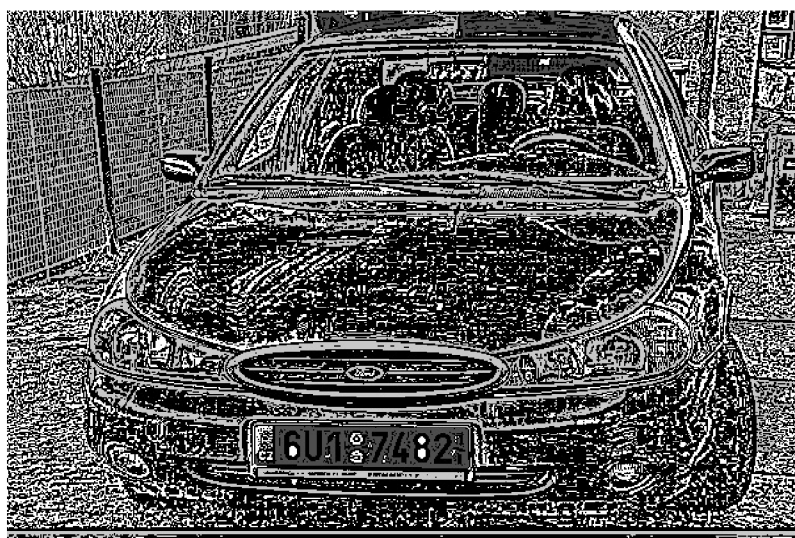
Můžeme zvolit netradiční přístup rozpoznání mezi několika oblastmi. Vycházet z barvy oblasti, kterou mají všechny SPZ stejnou, tedy bílá po okraji. Problém by mohl nastávat až v případě většího naklonění značky na obraze.

Sečtením prvních a posledních 5 řádků oblasti a porovnáním s vytvořenou šablonou řádků. Opět bychom vytvořili šablonu o pěti položkách prvních řádků, posledních a pěti sloupců. Sloupce vlevo bychom nemohli zahrnout do tohoto testu. Důvodem je možný výskyt proužku s CZ značkou, který by znehodnocoval výsledky. Nejmenší rozdíl šablony a



Obrázek 3.4: Prahovaný obrázek, vyznačená nalezená SPZ, světle šedé oblasti jsou ty, které neprošli mezi kandidáty.

vstupu by byl opět výsledkem. Tato metoda se nakonec osvědčila a výsledné nalezení SPZ v obraze je 98%, více se dozvíme o výsledcích v kapitole 5.



Obrázek 3.5: Oblasti, které vyhovují kritériím pro uložení mezi kandidáty.

3.2 Nalezení znaků a jejich klasifikace

Souřadnice určující SPZ oblast je potřeba upravit, abychom s nimi mohli dále pracovat. Následuje vyhledání znaků a potřebná úprava před rozpoznáváním. Více si popíšeme v implementaci 4.3

Kapitola 4

Implementace

V další části se podíváme na implementaci, která proběhla přesně podle návrhu 3. Možnost nahlédnutí do zdrojových kódů je na příloženém DVD na zadní straně desek, jednotlivé funkce jsou okomentovány podrobným popisem. Jak jsem psal v kapitole 3, bude použita knihovna OpenCV v jazyce C/C++.

Popíšeme si zde používané funkce v aplikaci a její ovládání.

4.1 Nejpoužívanější knihovní funkce a struktury

Uvedeme si nejprve nejpoužívanější datové struktury v programu s krátkým popisem, které knihovna obsahuje a my s jejich pomocí si usnadňujeme práci.

IplImage* - struktura pro práci s obrázky, po načtení obrázku obsahuje jeho informace.

Využívá se hlavně jako odkaz na data obrázku. Aplikace využívá některé položky struktury: `nChannels`, `width`, `height`, `imageData` a `widthStep`

CvSize(width, height) - struktura pro definici obrázku, parametry jsou výška a šířka

CvHistogram* - ukazatel na strukturu pro práci s vytvořeným histogramem 2.2.8

Nyní se podíváme na nejpoužívanější funkce z knihovny, uvedeme si i krátký popis s definováním proměnných.

CvHistogram* cvCreateHist(cDims, dims, type, ranges, uniform) - funkce sloužící k vytvoření histogramu, *cDims* - počet dimenzí, *dims* - kolikarozměrný je histogram, *type* - jak jsou data reprezentována (CV_HIST_ARRAY, CV_HIST_TREE), *ranges* - pole rozsahů pro histogram

void cvCalcHist(img, hist, doNotClear, mask) - funkce pro výpočet histogramu z obrázku, *img* - zdrojový obrázek, *hist* - histogram pro uložení výsledku

void cvGetMinMaxHistValue(hist, minVal, maxVal, minIdx, maxIdx) - funkce pro získání minimální a maximální hodnoty výskytu v *hist*

IplImage* cvLoadImage(filename, iscolor) - funkce pro načtení obrázku ze souboru, *filename* - vstup (cesta), *iscolor* - definujeme načtení v požadované barvě, v našem případě využita hodnota CV_LOAD_IMAGE_GRAYSCALE

IplImage* cvCreateImage(size, depth, channels) - funkce pro vytvoření obrázku, *size* - velikost vytvořeného obrázku, použití *cvSize*, *depth* - hloubka (IPL_DEPTH_8U), *channels* - počet kanálů na pixel

void cvResize(src, dst, interpolation) - funkce pro změnu velikosti obrázku, *src* - zdrojový obrázek, *dst* - cíl změny, *interpolation* - interpolační metoda, zde používáme CV_INTER_LINEAR

void cvCopy(src,dst) - funkce využitá ke kopírování obrázku ze zdroje *src* do *dst*, další parametry nevyužíváme

void cvAdaptiveThreshold(src, dst, max_value, Tmet, thresholdType, blockSize) - funkce slouží k prahování 2.2.7, konkrétně k adaptivnímu, *src* - zdrojový obrázek, *dst* - cíl změny, *max_value* - maximální hodnota v *src*, použita v *thresholdType*, *Tmet* - metody adaptivního prahování (MEAN_C,GAUSSIAN_C), *thresholdType* - typ prahování, musí být buď CV_THRESH_BINARY nebo CV_THRESH_BINARY_INV, *blockSize* - velikost sousedních pixelů používaných pro počítání

void cvSetImageROI(img, rect) - funkce slouží k výřezu oblasti definovanou pomocí *rect* (*CvRect*) z obrázku *img*

void cvEqualizeHist(src, dst) - funkce ekvalizuje *src* do *dst*, o ekvalizaci pojednáváme zde 2.2.8

void cvReleaseImage(img) - funkce pro uvolnění dat z *img*

void cvReleaseHist(hist) - funkce pro uvolnění dat z *hist*

int cvNamedWindow(name, flags) - funkce, která vytvoří okno s obrázkem, *name* - jméno okna, *flags* - příznak k otevření, používá se CV_WINDOW_AUTOSIZE

void cvMoveWindow(name, x, y) - funkce pro určení místa na obrazovce, kde se dané okno zobrazí, *name* - jméno okna, které má být umístěno, *x,y* - souřadnice místa umístění, počítáno od levého horního rohu

void cvShowImage(name, img) - funkce pro zobrazení obrázku *img* do okna s daným názvem *name*

int cvSaveImage(fileName, img) - funkce pro uložení aktuálních dat obrázku do souboru, *fileName* - jméno a cesta pro uložení, *img* - obrázek, který má být uložen

4.2 Segmentace a nalezení kandidátů

Jak jsme psali v návrhu 3.1.1 prahování bylo implementováno pomocí funkce *cvAdaptiveThreshold*. Výběr prahu pro tuto funkci byl zvolen pomocí histogramu, kde využíváme nalezení hodnoty s největší četností jasu. Upravené semínkové vyplňování nám vyhledá oblasti a my je pomocí zmiňovaných testů označíme za kandidáty a nebo ne.

Pro výběr z mnoha kandidátních oblastí byla použita metoda podobnosti v řádcích a sloupcích. Metoda šablony celé oblasti byla zavádějící a každá testovaná oblast vycházela jinak. Nedostali jsme se ani nad 50% nalezených SPZ značek v obraze.

4.3 Úprava oblastí a rozpoznávání znaků

Státní poznávací značku nám reprezentují čtyři body, které jsou předány k další úpravě. Nyní použijeme kopii originálu obrázku a vyřízneme oblast znovu. Oblast je dále znovu prahována s jinými hodnotami. Dosáhneme tak lepšího rozpoznávání. Tato volba se osvědčila v lepších výsledcích, než použití jednou prahovaného obrazu. Obraz je zmenšen, nebo zvětšen na velikost 128 * 32 px pro lepší zpracování.

Dalším krokem je úprava okrajů oblasti. Program vezme první tři řádky a otestuje jejich velikost pomocí součtu řádku s maximálně přípustnou hodnotou. Stejně se provede pro spodní řádky a sloupce na pravé straně. Levou stranu je potřeba ořezat důkladněji, jak již bylo zmíněno, může obsahovat modrý proužek (po prahování černý).



Obrázek 4.1: a) SPZ před úpravou b) SPZ po úpravě ořezáním okrajů.

4.3.1 Vyhledání znaků

Ořezaná oblast je z pohledu výšky "rozpůlena" a jsou postupně brány body po celém řádku. Tyto body jsou testovány a pokud narazíme na černou barvu, je odeslána souřadnice k přesné lokalizaci znaku.

Vyhledání pozice znaku je provedeno osvědčenou semínkovou metodou, zaznamenáváním největších a nejmenších krajních bodů. Jestli umístíme daný znak v souřadnicích do zásobníků znaků rozhodneme ještě podle testu, který určí jestli splňuje minimální kritéria. Otestujeme zda není příliš malý, tím vyloučíme případné malé černé části, nebo STK a SME oblasti. Na obrázku 4.3 jsou ohrazené znaky k lepší demonstraci.



Obrázek 4.2: Ohrazené nalezené znaky v SPZ



Obrázek 4.3: Ohrazené nalezené znaky v SPZ (mírně natočené).

4.3.2 Úprava znaků

Konečným výsledkem vyhledávání znaků je zásobník o souřadnicích. Pro rozpoznání je potřeba znak upravit tak, aby jsme jej mohli plnohodnotně porovnávat s předem připravenými šablonami. Po testování jsem zvolil velikost jednoho znaku na 25 * 12 px. Jestliže je některý menší jak 23 a nebo 10 px je zvětšen do definované velikosti, větší naopak. Roztažený znak neodpovídá své podobě a je tedy znehodnocen. Proto jsou znaky 7,1 nebo I podle definované výšky přesakovány, co se týče zvětšování a zmenšování.

je tedy založeno na součtu rozdílů horizontálního a vertikálního histogramu znaku a histogramů šablony znaku. Každý předpřipravený znak porovnáme se všemi šablonami. Vyjimku tvoří čtyři číslice vpravo, které porovnáme pouze s číslicemi.

Další volbou je zobrazení přesnosti určení jednotlivých znaků, neboli procentuelní vyjádření přesnosti označení. Šablony jsou vytvořeny v souboru *hist_znaky.txt*, který je ve složce *src*. Každá šablona byla vytvořena z cca 5-10 znaků toho druhu. Hodnoty byly sečteny a zprůměrovány počtem.

Výsledek je vytisknut na standartní výstup, podle přidanych parametrů při spuštění jsou vypisovány další hodnoty.

4.4 Ovládání programu

V této bakalářské práci nebylo úkolem vytvořit jednoduchou aplikaci s GUI rozhraním. Vytvořený program je možné, po úspěšném přeložení, spouštět přes příkazovou řádku s několika parametry viz 4.6. Základní použití je: `./spz [-h] [-s jmeno obrazku] [-i] [-c] cesta/jmeno obrazku`

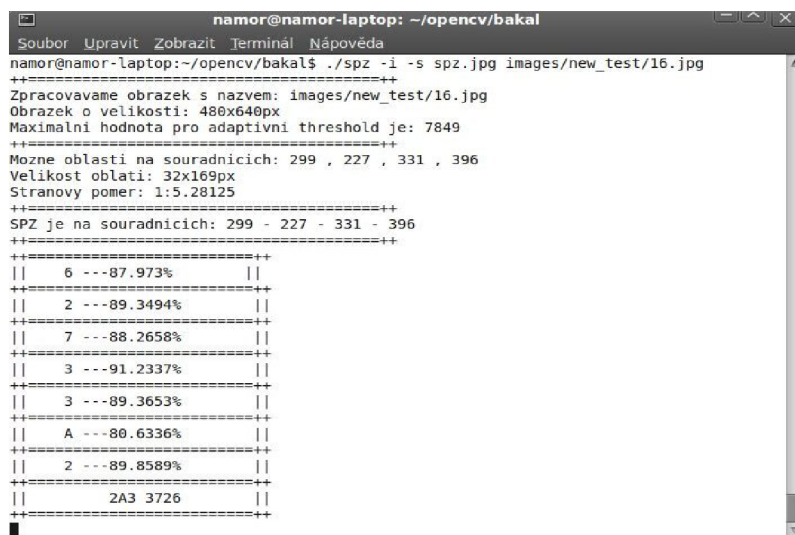
-h vytisknutí nápovědy, jak s programem zacházet a jaké jsou parametry

-s slouží k uložení SPZ výřezu do aktuální složky, za tímto parametrem musí následovat název a cesta pro uložení výřezu

-i zobrazení podrobného výpisu programu a těmi jsou:

- informace o načteném obrázku,
- možné kandidátní oblasti a jejich atributy,
- nalezená SPZ oblast,
- výpis přesnosti určení znaku

-c zobrazení pouze SPZ bez jakéhokoliv výpisu, možnost pro vložení aplikace do programu



```
namor@namor-laptop: ~/opencv/bakal
Soubor Upravit Zobrazit Terminál Nápověda
namor@namor-laptop:~/opencv/bakal$ ./spz -i -s spz.jpg images/new_test/16.jpg
++++
Zpracovavame obrazek s nazvem: images/new_test/16.jpg
Obrazek o velikosti: 480x640px
Maximalni hodnota pro adaptivni threshold je: 7849
++++
Možne oblasti na souradnicich: 299 , 227 , 331 , 396
Velikost oblati: 32x169px
Stranovy pomer: 1:5.28125
++++
SPZ je na souradnicich: 299 - 227 - 331 - 396
++++
|| 6 ---87.973% ||
++++
|| 2 ---89.3494% ||
++++
|| 7 ---88.2658% ||
++++
|| 3 ---91.2337% ||
++++
|| 3 ---89.3653% ||
++++
|| A ---80.6336% ||
++++
|| 2 ---89.8589% ||
++++
|| 2A3 3726 ||
++++
```

Obrázek 4.6: Ukázka výpisu v příkazové řádce.

Kapitola 5

Testování a výsledky

Cílem této kapitoly je určit jednotlivé míry spolehlivosti při vyhledávání v obraze a rozpoznávání.

5.1 Metodika testování

Jak jsem uváděl v předchozích kapitolách, zkoušel jsem experimentovat se způsoby segmentace, použití šablony při vybírání z kandidátních oblastí pravé značky. Docházelo k horším výsledkům, než u implementovaných metod. Nejlepších výsledků jsem dosáhl navrženým programem, ikdyž výsledky nejsou ideální a způsoby implementace dokonalé. Navrženou aplikaci jsme testovali přibližně na 120 vzorcích automobilů postupně při vývoji.

5.2 Závěrečný test a výsledky

Uvedu zde výsledky závěrečného testu, kde byly použity **nově pořízené fotografie, tedy takové, které nebyly použity při vývoji.**

Test proběhl na 50 vzorcích s těmito výsledky:

Vzorek	Oblasti v obraze		Znaky v kandidátovi		SPZ porovnání	
	Kandidátní oblasti	Nalezena SPZ	Znaků nalezeno	Rozpoznané + jistota	Výsledek SPZ	Originál SPZ
1.jpg	1	ANO	7	7 - 86,8%	2H7 2462	2H7 2462
2.jpg	1	ANO	7	5 - 79,8%	Z39 5737	2S9 5737
3.jpg	1	ANO	7	7 - 84,3%	1AC 6563	1AC 6563
4.jpg	1	ANO	7	5 - 77,8%	505 8376	5U5 8316
5.jpg	1	ANO	7	7 - 81,8%	9A3 0456	9A3 0456
6.jpg	1	ANO	7	7 - 82,4%	4P3 6321	4P3 6321
7.jpg	2	ANO	7	7 - 86,2%	3T5 4183	3T5 4183
8.jpg	1	ANO	7	6 - 83,6%	338 4546	3S8 4546
9.jpg	1	ANO	7	6 - 78,8%	406 9857	4S6 9857
10.jpg	2	ANO	7	6 - 83,5%	162 4770	1B2 4770

Tabulka 5.1: Ukázka prvních 10 testovaných vzorků.

V tabulce 5.2 je výčet prvních 10 testovaných vzorků. V prvním sloupci je název fotografie, všech 50 testovaných fotografií je k dispozici na DVD ve složce *images/test*. Druhý sloupec nám ukazuje počet kandidátních oblastí, které jsem v obraze našel a z kterých tedy vybírám potenciální SPZ. Vidíme také jestli byla značka nalezena. V dalším sloupci je počet znaků, které jsem našel, a které jsem rozpoznal správně. U správně rozpoznávaných je k dispozici procentuální hodnota jistoty rozpoznání všech znaků. Poslední dva sloupce ukazují výstup z aplikace s porovnáním originálu SPZ.

Vzorek	Oblasti v obraze		Znaky v kandidátovi		SPZ porovnání	
	Kandidátní oblasti	Nalezena SPZ	Znaků nalezeno	Rozpoznané + jistota	Výsledek SPZ	Originál SPZ
15.jpg	2	ANO	7	4 - 81,9%	RXL 1671	RKE 1661
22.jpg	4	ANO	7	5 - 84%	601 7492	6U1 7482
27.jpg	2	NE	0	0 - 0%	??? ????	8A7 4485
32.jpg	1	ANO	7	4 - 81,6%	7XX 5870	2AA 5870
36.jpg	1	ANO	6	4 - 80,8%	-0U 2636	UHE 2636
48.jpg	1	ANO	7	4 - 75,3%	8K0 6786	AK0 6488

Tabulka 5.2: Ukázka malých odchylek od ostatních testovaných vzorků.

Tabulka 5.2 ukazuje vzorky s větší odchylkou, než je v ostatních případech. Ve vzorcích 15,32,36 a 48 vidíme pouhé 4 znaky rozpoznané správně. V 27 vzorku, jako v jediném z 50 testovaných jsem nenalezl SPZ oblast. Za povšimnutí stojí také fotografie s číslem 22, kde jsem našel 4 kandidátní oblasti. Špatně rozpoznané znaky jsou způsobeny nedokonalostí šablon, s kterými srovnávám vstupní znaky.

Další neuvedené výsledky z 50 testovaných fotografií jsou k nahlédnutí na konci této kapitoly.

Shrnutí z **50 testovaných vzorků** z pohledu segmentace:

- - úspěšně nalezeno **49 oblastí obsahujících SPZ**, tedy **98% úspěšnost**
- - 1x špatně nalezena SPZ oblast
- - 2x špatně oříznuta SPZ oblast zprava (CZ proužek)
- - nejvíce kandidátních oblastí v jednom vzorku - 4
- - nejméně kandidátních oblastí v jednom vzorku - 1
- - průměr kandidátních oblastí na jeden obrázek je 1,34

Nyní se podíváme na výsledky z pohledu nalezení a rozpoznání znaků. Celkový počet znaků je 343 (7 znaků v SPZ * 49 vzorků). V jednom případě SPZ oblast nebyla nalezena, proto 49 vzorků. Nebude tedy započítán do celkové úspěšnosti rozpoznání na znak.

Shrnutí:

- - úspěšně nalezeno bylo 347 znaků z 350 možných, tedy **99,14% úspěšnost**
- - 3 znaky nenalezeny, důvodem je špatně oříznutý obrázek z pravé strany

- - 16x rozpoznány všechny znaky správně - 30%
- - 10x rozpoznáno pouze 6/7 znaků - 20%
- - 20x rozpoznány pouze 5/7 znaků - 40%
- - 4x rozpoznáno pouze 4/7 znaků - 8%
- - 1x nebyla nalezena SPZ
- - průměrné procentuální vyjádření jistoty ozančení znaků je 81.6%
- - průměrné procentuální vyjádření jistoty ozančení znaků je 81.6%
- - z celkového počtu **343 znaků bylo 288 rozpoznáno správně**, 55 špatně
- - **úspěšnost rozpoznání na jeden znak činí 83,96%**

Z výsledků uvedených v krátkých výpisech je zřejmé, že jsem splnil podmínky, které jsem si předem stanovil **3**. Navrhovaných 85% pro nalezení SPZ oblasti jsem splnil. Nalezení znaku v oblasti jsem předpokládal alespoň v 90% případů, tuto podmínku jsem také splnil. Jednotlivé znaky rozpoznávám na 83,96% na znak v SPZ oblasti, navrhovaných 60% jsem překonal o 20%.

Dalším stanoveným cílem bylo otestování alespoň dvou metod segmentace. Segmentaci jsem zkoušel také pomocí detektoru hran, ovšem výsledky byly mnohem horší než při použití adaptivního prahování. Test byl proveden pouze na prahování, podrobné výsledky z druhé metody nejsou součástí této práce.

Možnost ukládání SPZ výřezu do souboru a podrobné výpisy při spuštění aplikace je možné nastudovat z **4.4**.

Posledním cílem bylo otestování rozpoznávání znaků na jiné dostupné open-source aplikaci. Zde jsem zvolil aplikaci GOCR, která je k dispozici na <http://jocr.sourceforge.net/>. Vstupem pro stažený a aplikovaný program byl výřez SPZ. Aplikaci jsem spustil a předal jí v parametru náš výřez. Pro čisté výřezy (znaky na bíle ploše, pozadí celé bíle bez černých skvrn) aplikace rozeznávala každý znak bez problémů. Pokud na její vstup byl dosazen výřez, který byl v pozadí s černými skvrnami, znaky byly spojené s černými pixely, nebo byl celý výřez nakloněný, probíhalo rozpoznávání špatně. Znaky nebyly rozpoznány vůbec a nebo jenom částečně. Skvrny byly poznávány jako podtržítka nebo tečky. Aplikace nabízí, pomocí parametrů při spouštění, mnoho možností a úprav při rozpoznání. Zkoušel jsem i tuto možnost a došel jsme tedy k závěru, že pro každý obrázek by se musely nastavovat jedinečné parametry s jinými hodnotami. Tato aplikace by se pravděpodobně nedala využít v mojí práci k rozpoznávání znaků.

Mezi nedostatky mnou vytvořené aplikace můžeme zařadit například obrazy pořízené z větší vzdálenosti než je přibližně 8m. V takovém případě není SPZ nalezena a následně rozpoznána. Dalším nedostatkem, který byl zjištěn v testování jsou boční pohledy automobilu. Větší naklonění značky než cca 20 stupňů vede k nenalezení některých znaků, případně celé SPZ oblasti. Oba případy je možné řešit dokonalejším způsobem vyhledávání oblastí a znaků.

Vzorek	Oblasti v obraze		Znaky v kandidátovi		SPZ porovnání	
	Kandidátní oblasti	Nalezena SPZ	Znaků nalezeno	Rozpoznané + jistota	Výsledek SPZ	Originál SPZ
11.jpg	1	ANO	7	6 - 83,9%	152 4770	1B2 4770
12.jpg	1	ANO	7	5 - 82,5%	13L 8974	1SL 8944
13.jpg	1	ANO	7	5 - 81,5%	3TB 1648	3T8 1646
14.jpg	1	ANO	7	7 - 87%	CVI 6649	CVI 6649
16.jpg	1	ANO	7	7 - 88%	2A3 3726	2A3 3726
17.jpg	2	ANO	7	6 - 84,5%	464 3049	4B4 3049
18.jpg	2	ANO	7	5 - 83,2%	AUJ 0071	AHJ 0031
19.jpg	2	ANO	7	5 - 79,4%	363 6587	3B3 6582
20.jpg	1	ANO	7	7 - 81,8%	3T4 6805	3T4 6805
21.jpg	2	ANO	7	5 - 80,8%	7A0 1978	2AB 1978
23.jpg	1	ANO	7	7 - 87,3%	7A7 7371	7A7 7371
24.jpg	2	ANO	7	7 - 82%	3C4 5556	3C4 5556
25.jpg	1	ANO	7	7 - 84,6%	4H3 7070	4H3 7070
25.jpg	1	ANO	7	7 - 84,6%	4H3 7070	4H3 7070
26.jpg	1	ANO	7	5 - 80,6%	7X0 8070	4U0 8070
28.jpg	1	ANO	7	7 - 82,6%	3T0 2735	3T0 2735
29.jpg	1	ANO	7	7 - 82,3%	3T0 2735	3T0 2735
30.jpg	1	ANO	7	5 - 77,4%	7A8 0814	1AB 0814
31.jpg	2	ANO	7	5 - 80%	10C 4187	1AC 4157
33.jpg	2	ANO	7	6 - 81%	376 7978	3T6 7978
34.jpg	1	ANO	7	5 - 80,7%	B03 3060	8S3 3060
35.jpg	1	ANO	7	6 - 75,4%	893 3060	8S3 3060
37.jpg	1	ANO	7	5 - 80,7%	JIV 1790	JIH 1490
38.jpg	1	ANO	7	6 - 81,7%	2P1 9397	2P1 9392
39.jpg	2	ANO	7	5 - 77,3%	4PZ 8960	4P2 8968
40.jpg	1	ANO	6	5 - 79,1%	-X2 2506	5U2 2506
41.jpg	2	ANO	7	6 - 79%	2Z8 4879	2Z8 4819
42.jpg	1	ANO	7	7 - 84%	6T7 3365	6T7 3365
43.jpg	2	ANO	7	5 - 78,2%	4CX 5389	OCK 5389
44.jpg	1	ANO	7	5 - 78,5%	862 0663	5B2 0663
45.jpg	1	ANO	7	6 - 82%	B7C 1917	BZC 1917
46.jpg	1	ANO	7	7 - 81,3%	4J0 6896	4J0 6896
47.jpg	3	ANO	7	5 - 78,6%	784 8820	7B4 8920
49.jpg	1	ANO	6	5 - 81,6%	-X3 5424	2K3 5424
50.jpg	1	ANO	7	5 - 80,2%	156 6839	1B6 6809

Tabulka 5.3: Zbytek testovaných vzorků.

Kapitola 6

Závěr

Cílem této práce bylo seznámit se s problematikou státních poznávacích značek, prostudovat metody zabývající se zpracováním obrazu a rozpoznáváním vzorů. Vybranou metodu aplikovat pro detekci SPZ a pro její následné rozpoznání (převedení do textové podoby). Implementaci otestovat a vyhodnotit výsledky celé aplikace. Tento úkol se podařilo splnit.

Problematikou poznávacích značek se zabývám v kapitole 2, zpracování obrazu a rozeznávání vzorů popisují v kapitole 2.2. Z těchto kapitol jsem sestavil návrh 3 pro aplikaci rozeznávající značky. Návrh byl dále implementován. Použité funkce z knihovny OpenCV, výběr metod a ovládání programu jsou popsány v implementaci 4. Pro konečný test bylo získáno 50 nových vzorků, které nebyly již dříve otestovány. Konečný výsledek lze posoudit v kapitole 5.

Dosáhl jsem 98% úspěšnosti nalezení SPZ v obraze, 99,14% úspěšnosti nalezení znaků v SPZ a konečného rozpoznání 83,96% na jeden znak. Využití vyvinuté aplikace je přesto minimální a z výsledků vidíme, že jako rozpoznávací jednotka pro policii by aplikace neuspěla. Nabízí se tedy možnost využití aplikace na závoře hlídaného parkoviště apod. Mohla by být využita např. na kontrolu automobilů, které mají přístup do dané oblasti.

Jako každý vědecký objev je dále vylepšován i tuto aplikaci by bylo možné dále rozvíjet a dosáhnout spolehlivějších výsledků. Celý systém by bylo možné rozšířit tak, aby dokázal rozeznávat další značky vyskytující se v České republice. Jeho využití bychom tímto rozšířili. Dalším možným vylepšením by byla optimalizace navrhnutých algoritmů, neboli zrychlení aplikace. Nabízí se také možnost rozšíření o určení kraje, z kterého auto pochází.

Literatura

- [1] Dobeš Michal: *Zpracování obrazu a algoritmy v C#*. BEN-Technická literatura, 2008, iSBN 978-80-7300-233-6.
- [2] Hlaváč, V., Šonka, M.: *Počítačové vidění*. Praha: Grada. Edice Nestůjte za dveřmi., 1992, iSBN 80-85424-67-3.
- [3] Španěl a Ing. Vítězslav Beran, I. M.: *Obrazové segmentační techniky*.
http://www.fit.vutbr.cz/info/statnice/psani_textu.html, 2005-12-10 [cit. 2011-05-03].
- [4] Kalová, I.: *Segmentace a detekce geometrických primitiv*.
<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/mpov.html.cs>, 2005-12-10 [cit. 2011-05-03].
- [5] Lačezar Ličev a Štěpán Sojka: *Rozpoznávání zájmových bodů a objektů na snímcích*.
http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2009/sbornik/Lists/Papers/002.pdf, 2009.
- [6] Prof. Ing. Václav Matoušek, CSc.: *Rozpoznávání a klasifikace*.
<http://www.kiv.zcu.cz/studies/predmety/uir/predn/P4/FThema4.pdf>, 16. - 23. 3. 2005.
- [7] Prof. Ing. Hlaváč Václav, C. a. M. B. H.: *Digitální zpracování obrazu. Jasové a geometrické transformace*. Strana 7-8.
- [8] Vala, T.: *Rozpoznávání SPZ z jednoho snímku*. Diplomová práce, Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, 2006.
- [9] WWW stránky-Filip Zelený: *Poznávací značky silničních vozidel v Českých zemích - historie a vývoj*. <http://www.k-report.net/spz/>, 2007-01-11.
- [10] Šíma, Jiří - Neruda, Roman: *Teoretické otázky neuronových sítí*. Matfyzpress, 1996, iSBN 80-85863-18-9.
- [11] Černý, M.: *Rozpoznávání registračních značek motorových vozidel*. Diplomová práce, UNIVERZITA PARDUBICE, Fakulta elektrotechniky a informatiky, 2010.

Příloha A

Obsah DVD

- **bin** - binární přeložené soubory
- **src** - zdrojové soubory aplikace
- **images** - obrázky pro test uváděný v kapitole 5 a obrázky používané při vývoji
- **doc** - zdrojové soubory dokumentace a tohoto pdf vytvořené v L^AT_EXu
- **files** - pomocné soubory pro ukládání různých datových struktur
- **makefile** - soubor pro přeložení projektu
- **test.sh** - spustitelná dávka testu, příkaz pro spuštění ./test

Příloha B

Ukázka správného rozpoznání



Obrázek B.1: Ukázka správně rozpoznaných SPZ.



Obrázek B.2: Ukázka správně rozpoznaných SPZ.



Obrázek B.3: Ukázka správně rozpoznaných SPZ.

Příloha C

Ukázka špatného rozpoznání



Obrázek C.1: Ukázka rozpoznání několika znaků SPZ.