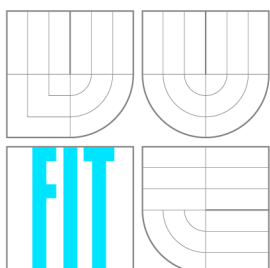


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



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

# SYSTÉM AUTOMATICKÉHO PŘÍSTUPU NA PARKOVIŠTĚ POMOCÍ ROZPOZNÁNÍ REGISTRAČNÍ ZNAČKY

SYSTEM FOR AUTOMATIC PARKING ACCESS BASED ON LICENSE PLATE RECOGNITION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PATRIK VÁCLAVEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAKUB ŠPAŇHEL

BRNO 2016

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

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

**Zadání bakalářské práce**

Řešitel: **Václavek Patrik**

Obor: Informační technologie

Téma: **Systém automatického přístupu na parkoviště pomocí rozpoznání registrační značky**

**System for Automatic Parking Access Based on License Plate Recognition**

Kategorie: Zpracování obrazu

**Pokyny:**

1. Prostudujte základy zpracování obrazu. Zaměřte se zejména na problematiku detekci a rozpoznání registrační značky v obraze.
2. Vyberte vhodnou metodu a navrhnete systém automatického přístupu na parkoviště.
3. Experimentujte s vaší implementací a případně navrhnete vlastní modifikace metod.
4. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje. Zvažte další pokračování v rámci diplomové práce.
5. Vytvořte stručný plakát nebo video prezentující vaši bakalářskou práci, její cíle a výsledky.

**Literatura:**

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

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

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

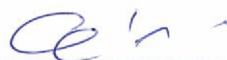
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo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

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

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
612 06 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Cílem této práce bylo navrhnout a implementovat systém pracující v reálném čase, schopný detekovat příjezd vozidla k terminálu parkoviště, rozpoznat registrační značku a automaticky rozhodnout o jeho vpuštění. Systém pro detekci příjezdu automobilu využívá algoritmus Gaussian Mixture Model. Pro spolehlivou lokalizaci registrační značky jsou použity dvě metody. V první jsou extrahovány Maximálně Stabilní Extrémní Regiony (MSER), zatímco ve druhé je použita transformace Top–Hat. Pro určení, zda se u nalezených regionů jedná o registrační značku, je využito klasifikátoru Support Vector Machine (SVM). Klasifikace znaků je provedena použitím umělé neuronové sítě. Pro implementaci byla použita knihovna OpenCV. Extrakce MSER byla díky navržené optimalizaci zrychlena až sedmkrát. Při lokalizaci registrační značky se podařilo dosáhnout úspěšnosti 92,47% a spolehlivost klasifikace dosahuje 90,03%.

## Abstract

Goal of this thesis was to design and implement system operating in real time, which manages to detect incoming vehicle to the car park terminal, recognize its licence plate and automatically decide on its admission. System uses the Gaussian Mixture Model algorithm for detection of incoming vehicle. For reliable localization of licence plate are used two methods, the first one uses of extraction of Maximally Stable Extremal Regions (MSERs), the second one uses of Top–Hat transformation. Support Vector Machine (SVM) algorithm is used to decide, whether is the found area a licence plate. Character classification is performed using artificial neural network. For implementation was used library OpenCV. Thanks to optimalization is the extraction of MSERs accelerated up to seven times. The accomplished success rate in case of licence plate localization is 92,47% and in case of classification of characters is 90,03%.

## Klíčová slova

rozpoznání registrační značky, nalezení registrační značky, zpracování digitálního obrazu, detekce, klasifikace, extrakce

## Keywords

licence plate recognition, licence plate localization, digital image processing, detection, classification, extraction

## Citace

VÁCLAVEK, Patrik. *Systém automatického přístupu na parkoviště pomocí rozpoznání registrační značky*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Špaňhel Jakub.

# System automatického přístupu na parkoviště pomocí rozpoznání registrační značky

## Prohlášení

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

.....  
Patrik Václavek  
18. května 2016

## Poděkování

Rád bych poděkoval svému vedoucímu za ochotu a nepostradatelné rady, které mi pomohly při řešení. Také bych rád poděkoval panu Michalu Hájkovi, provozovateli portálu 3260.cz, za poskytnutí svých snímků registračních značek. V neposlední řadě děkuji rodině a přátelům za podporu a pomoc při vytváření datové sady.

© Patrik Václavek, 2016.

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



# Obsah

<b>1 Úvod</b>	<b>2</b>
<b>2 Teorie nalezení a rozpoznání RZ</b>	<b>3</b>
2.1 Vlastnosti registračních značek . . . . .	3
2.2 Metody lokalizace RZ na snímku . . . . .	4
2.3 Metody klasifikace znaků . . . . .	8
<b>3 Použité metody nalezení a rozpoznání RZ</b>	<b>10</b>
3.1 Metoda GMM pro detekci pohybu . . . . .	10
3.2 Metoda extrakce MSER . . . . .	11
3.3 Transformace Top-Hat . . . . .	12
3.4 Lineární klasifikátor SVM . . . . .	13
3.5 Vícevrstvá umělá neuronová síť a algoritmus zpětného šíření chyb . . . . .	14
<b>4 Návrh systému automatického přístupu na parkoviště</b>	<b>17</b>
4.1 Analýza problému . . . . .	17
4.2 Základní skladba systému . . . . .	18
4.3 Detekce příjezdu vozidla . . . . .	19
4.4 Nalezení registrační značky . . . . .	20
4.5 Rozpoznání registrační značky . . . . .	24
<b>5 Implementace a vyhodnocení dosažených výsledků</b>	<b>27</b>
5.1 Vytvořený program . . . . .	27
5.2 Datová sada . . . . .	28
5.3 Trénování algoritmů strojového učení . . . . .	28
5.4 Vyhodnocení dosažených výsledků . . . . .	30
<b>6 Závěr</b>	<b>33</b>
<b>Literatura</b>	<b>34</b>
<b>Přílohy</b>	<b>36</b>
Seznam příloh . . . . .	37
<b>A Matice záměn klasifikace znaků</b>	<b>38</b>
<b>B Manuál</b>	<b>39</b>
B.1 Kompilace programu . . . . .	39
B.2 Použití programu . . . . .	39

# Kapitola 1

## Úvod

Systémů fungujících na principu rozpoznání registrační značky existuje velmi mnoho a nalézájí své uplatnění po celém světě v nejrůznějších oblastech lidské činnosti. Takovéto systémy jsou v dnešní době využívány například u mýtných bran a pro nejrůznější bezpečnostní, monitorovací či statistické účely. Systém automatického vpouštění na parkoviště tak může mít díky rozpoznání registrační značky mnoho dalších praktických využití. Příkladem může být jeho využití na placených parkovištích pro automatický výpočet dlužné částky na základě doby parkování. Tato práce se zaměřuje na vpouštění vozidel na parkoviště na základě seznamu povolených registračních značek.

Cílem této práce bylo navrhnout a implementovat plně automatický systém schopný v reálném čase na základě vstupu ze stacionární kamery detekovat příjezd automobilu, rozpoznat jeho registrační značku a rozhodnout o jeho vpuštění. Práce je tak především zaměřena na spolehlivé nalezení a následné rozpoznání registrační značky.

Navržený systém využívá algoritmus *Gaussian Mixture Model* pro detekci příjezdu automobilu. Díky tomu je také schopný redukovat prohledávanou část snímku pouze na oblast, ve které je detekován pohyb. Pro nalezení registrační značky jsou za účelem spolehlivosti využity dvě metody. První metoda využívá extrakce *Maximally Stable Extremal Regions*, zatímco druhá metoda využívá transformaci *Top-Hat*. Pro rozhodnutí, zda se u nalezených oblastí jedná o registrační značku, je využit algoritmus strojového učení *Support Vector Machine*, a pro klasifikaci znaků je využita umělá neuronová síť.

V kapitole 2 jsou stručně popsány existující práce zabývající se problematikou lokalizace a rozpoznání registračních značek. Použité algoritmy počítačového vidění jsou podrobněji probrány v kapitole 3. Navržený systém včetně konkrétního popisu fungování jednotlivých metod je popsán ve 4 kapitole. V kapitole 5 je stručně popsána implementace, datová sada a především jsou zde zhodnoceny dosažené výsledky.

## Kapitola 2

# Teorie nalezení a rozpoznání RZ

Tato kapitola pojednává o používaných metodách lokalizace a rozpoznání registračních značek. Je zde představeno několik různých prací zabývajících se touto problematikou a každá z nich je stručně popsána. Nejprve je však nutné zmínit nejpodstatnější vlastnosti registračních značek.

### 2.1 Vlastnosti registračních značek

Zákoník České republiky rozlišuje hned několik typů registračních značek. Co se rozměrů týče, je dle vyhlášky Ministřestva dopravy č. 343/2014 k roku 2016 k dispozici celkem 7 různých velikostí. S výjimkou dočasných registračních značek, jsou všechny značky vyrobeny ze slitin lehkých kovů s již prolisovanými znaky. Nejčasteji používanou velikostí registračních značek pro většinu motorových vozidel je 520 x 110 mm (viz obrázek 2.1). Tento rozměr je běžně používán také na Slovensku. Všechna dále zmíněná fakta se budou vztahovat k tomuto typu registrační značky.



Obrázek 2.1: Typické rozměry registrační značky v České republice - 520x110 mm [10]

Značky se v drtivé většině umísťují do plastových rámečků na vozidle, umístěných na přední i zadní části vozidla. Značka by se měla nacházet v podélné ose vozidla, pokud to však konstrukce vozidla neumožňuje, může se nacházet i vlevo od ní. Značka by měla být vždy kolmo ke směru jízdy a přibližně vodorovně k vozovce. Jak dále vyhláška zmiňuje, značka by měla být za nesnížené viditelnosti čitelná na 40 metrů.

Od června roku 2004 se značky vyrábí s modrým pruhem, který je umístěn v levé části registrační značky. Ten obsahuje logo evropské unie a dvoupísmenný kód státu, jak lze vidět na obrázku 2.1. Značky jsou také mezi třetím a čtvrtým znakem opatřeny kruhovým prostorem pro umístění nálepky o technické kontrole. Dále je po obvodu registračních značek veden 4 mm široký pruh, jehož barva se shoduje s barvou znaků na značce. Oba tyto prostory jsou též prolisovány.

Vyhláška také rozlišuje několik barevných variant registračních značek (viz obrázek 2.2). Tabulky značek běžných motorových vozidel mají černé znaky na bílém podkladu. Naproti

tomu tabulky značek pro zvláštní motorová vozidla mají černé znaky na žlutém podkladu, pro historická vozidla zelené znaky na bílém podkladu a značky pro diplomatické účely modré znaky na bílém podkladu. Podklad všech značek je navíc reflexní.



Obrázek 2.2: Speciální typy registračních značek v České republice. Převzato z [10]

Znaky registračních značek jsou tvořeny velkými písmeny latinské abecedy a arabskými číslicemi. Každá značka je tvořena nejméně 5 a maximálně 8 znaky. Zatímco čísla mohou být použita všechna v rozsahu 0 až 9, písmena jsou omezena kvůli podobnosti, např. číslice 0 a písmena O nebo Q. Povolená písmena tedy jsou A, B, C, D, E, F, H, I, J, K, L, M, N, P, R, S, T, U, V, X, Y, Z [10].

## 2.2 Metody lokalizace RZ na snímku

Problematika lokalizace registračních značek na snímku je aktuálním tématem počítačového vidění již od 90. let minulého století. Za tuto dobu bylo vyvinuto hned několik více či méně efektivních technik. V této kapitole bude stručně představeno několik z nich. Většinu běžných algoritmů pro lokalizaci registrační značky lze rozdělit podle těchto tří principů, na kterých fungují: analýza barev, detekce hran a porovnávání se šablonou.

### 2.2.1 Předzpracování obrazu

Pro zvýšení přesnosti lokalizace registračních značek musí být nejprve použity některé techniky pro předzpracování obrazu. Takovýchto technik již bylo představeno mnoho a v následujících odstavcích jsou stručně vysvětleny některé z nich.

První technikou je převod obrazu na stupně šedi [1, 2, 4, 16, 7, 11]. Tato technika je využita v případech, kdy algoritmus pro lokalizaci registrační značky potřebuje znát informace o jasů v obraze. Pro odstranění šumu z obrazu je například používán mediánový filtr [2], který nastavuje hodnoty pixelů na medián hodnot okolních pixelů.

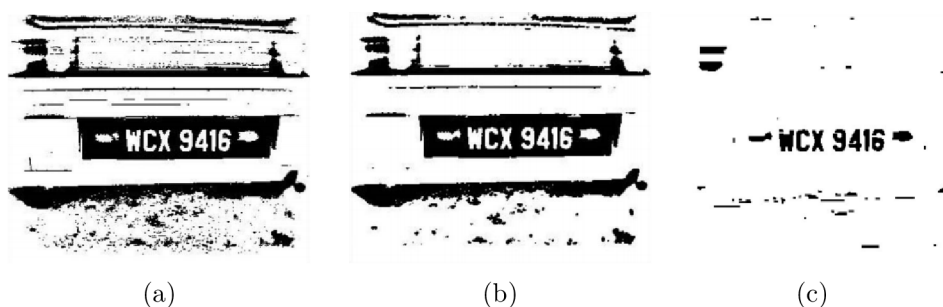
V článku [4] je použita technika ekvalizace histogramu, která zlepšuje kontrast obrazu. Tím zlepšuje i kvalitu obrazu a výsledky získané z detekce hran. Další technikou je binarizace, která se také využívá pro převod obrazu na stupně šedi. Může být například využita pro zvýraznění znaků na registrační značce nebo potlačení detailů pozadí. Pro binarizaci obrazu existuje mnoho metod. Například metoda adaptivního prahování použitá v práci [1] segmentuje obraz na regiony, a pro každý pak spočte vlastní práh. Hlavní nevýhodou

prahování je ovšem to, že občas vrátí neužitečné výsledky a jsou při něm ztraceny některé důležité informace o obraze.

Detekce hran [1, 16] je dalším krokem předzpracování obrazu. Ta detekuje ostré změny v jasů obrazu. Vertikální hrany znaků registrační značky mohou být zvýrazněny například operátorem Sobel [1, 4]. Dalším takovým operátorem pro detekci hran je například Canny [1]. Hlavní nevýhodou využívání těchto operátorů je to, že při použití na komplexnějších obrázcích mohou být zvýrazněny i nežádoucí hrany. Toto chování může být kompenzováno jejich použitím v kombinaci se základními morfologickými operacemi [2, 16, 11], které dokáží rychle a přesně eliminovat nežádoucí hrany. Často používanou morfologickou transformací je například Top–Hat [11], která zvýrazní oblast registrační značky tím, že potlačí pozadí a jiné nežádoucí oblasti.

### 2.2.2 Lokalizace RZ založená na detekci hran

Al-Ghaili a kol. [1] použili metodu nalezení a zpracování registračních značek v reálném čase, která je založena na hledání vertikálních hran. Nejprve je snímek převeden do stupňů šedi a poté binarizován pomocí jejich vlastního vylepšeného adaptivního prahování, kterému stačí jediný průchod obrázkem. Na obrázku 2.3a lze vidět jeho výstup. Následně jsou vyfiltrovány nežádoucí přímky s tloušťkou 1px, čímž je z obrázku redukován šum (viz obrázek 2.3b).



Obrázek 2.3: Na obrázku (a) je vidět výstup adaptivního prahování před filtrací nežádoucích přímek. Na obrázku (b) je vidět výsledek po vyfiltrování nežádoucích přímek. Na obrázku (c) je pak vidět výsledek detekce hran a následného zvýrazňování [1].

Na výsledném obrázku tak vznikly černé a bílé regiony, u kterých je následně provedena detekce vertikálních hran. Pro tento účel byl implementován vlastní algoritmus, který dosahuje 5x až 9x vyšších rychlostí než metoda Sobel. Poté přichází na řadu procedura, která po řádcích prochází obraz a porovná vždy tři sousední pixely a vyplňuje bílou oblast mezi černými pixely. Takto zvýrazněné oblasti jsou brány jako výsledné oblasti s registrační značkou, jak lze vidět na obrázku 2.3c.

### 2.2.3 Lokalizace RZ využitím morfologických operací

N. K. Babu a kol. [2] využili pro nalezení registrační značky morfologické operace. Snímek je nejprve převeden na šedotónový a následně je aplikována transformace pro roztažení kontrastu obrazu. V takto předzpracovaném obraze jsou pak hrany detekovány použitím Laplaceovy transformace. Takto detekované regiony jsou nakonec extrahovány a dále zpracovány pro detekci znaků.

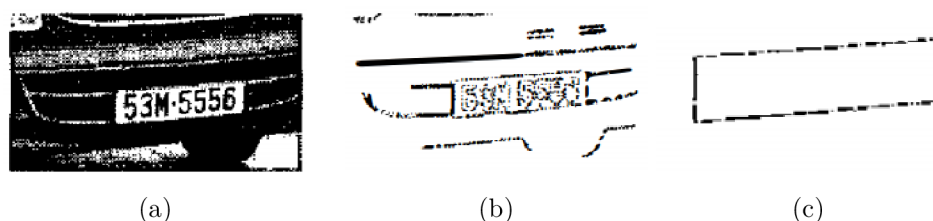
Joe C. H. Poon a kol. [11] využili morfologické operace pro lokalizaci registrační značky již v roce 1995. Po převedení na šedotónový je na obraz nejprve aplikována transformace



Top–Hat, která zvýrazní znaky v registrační značce. Poté je každý znak převeden na obdélník použitím opakované dilatace a minimalizace. Následně je detekována vzdálenost mezi znaky. Použitím morfologické operace *otevření* jsou pak detekovány takové horizontální přímky, které mezi sebou mají stejnou vzdálenost, jako je vzdálenost mezi znaky, a jsou na stejné horizontální úrovni. Tyto přímky jsou poté propojeny a jsou vybrány regiony mezi nimi. Tyto regiony jsou následně vyfiltrovány na základě jejich úrovně jasu. Zbylé regiony jsou vyprahovány na základě hodnot jejich histogramů. Ještě jsou vyhlazeny vnější okraje regionů, které jsou poté brány jako nalezené oblasti se značkami.

#### 2.2.4 Lokalizace RZ kombinací Houghovy transformace a detekce kontur

Pro lokalizaci registrační značky použil T.D. Duan a kol. [4] kombinaci Houghovy transformace a detekce kontur. Pro samotnou lokalizaci jsou zde zavedeny dva moduly, jeden pro předzpracování obrazu a druhý pro lokalizaci registrační značky. Modul pro předzpracování obrazu slouží k úpravě obrazu tak, aby v něm vynikly hrany. To je provedeno tak, že je obraz z kamery nejprve převeden na šedotónový, následně normalizován, a je provedena ekvalizace histogramu. Následně je aplikován Sobelův operátor a obraz je binarizován. Takto předzpracovaný obraz je poté předán druhému modulu pro lokalizaci (viz obrázek 2.4a).



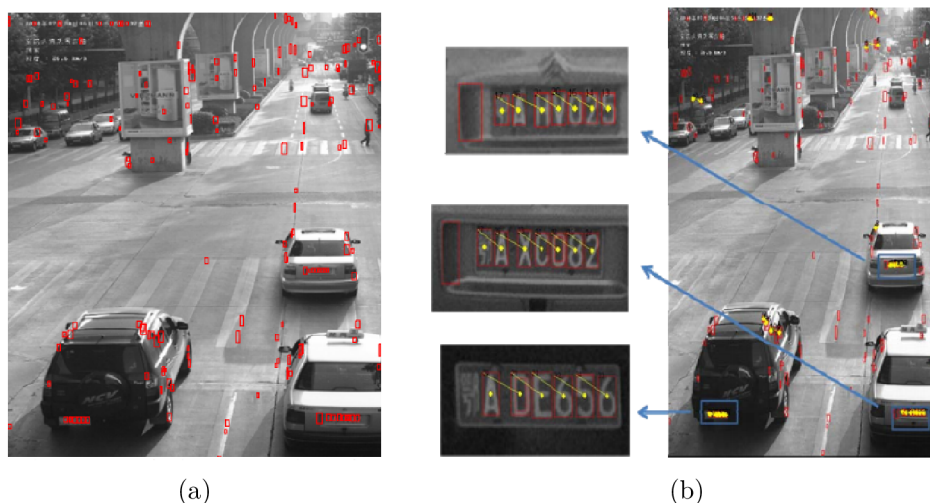
Obrázek 2.4: Na obrázku (a) je výstup modulu pro předzpracování obrazu. Na obrázku (b) je zobrazen výstup algoritmu pro detekci kontur. Na obrázku (c) je vykreslena kontura vyhodnocená jako RZ. Převzato z [4].

V modulu pro lokalizaci registrační značky je na obraz nejprve aplikován algoritmus pro detekci kontur pro nalezení uzavřených oblastí (viz obrázek 2.4b). Nalezené kontury jsou vyfiltrovány pomocí několika metod tak, aby zbyly pouze takové, které odpovídají vlastnostem registrační značky (viz obrázek 2.4c). Algoritmus pro detekci kontur je však nespolehlivý u obrázků s nízkou kvalitou. V těchto případech algoritmus vrací neuzavřené oblasti, a kontury tak nejsou nalezeny. Dalším krokem je v takto nalezených konturách vyhledání dvojice rovnoběžných přímků na základě kterých je poté rozhodnuto, zda se jedná o registrační značku či nikoliv. Tímto ověřením však mohou projít i jiné objekty, které jsou pak mylně pokládány za registrační značku. Další nevýhodou je potom výpočetní náročnost Houghovy transformace, proto se nehodí pro použití v real-time aplikaci.

#### 2.2.5 Lokalizace RZ pomocí extrakce MSER regionů

Zcela odlišný postup zvolili Bo Li a kol. [7], kteří pro své řešení použili algoritmus pro extrakci MSER (*Maximally Stable Extremal Regions*). Po jednoduchém předzpracování obrazu, kdy je obraz převeden na šedotónový, je aplikován algoritmus extrakce MSER. Ten

je lehce poupraven tak, aby výstupem byly pouze ty regiony, u kterých lze dle jejich vlastností předpokládat, že by se mohlo jednat o znaky na registrační značce. Příklad takového výstupu lze vidět na obrázku 2.5a.



Obrázek 2.5: Na obrázku (a) jsou červeně vykresleny *bounding-boxy* nalezených MSER regionů. Na obrázku (b) jsou zobrazeny skupiny regionů odpovídající registrační značce. Převzato z [7].

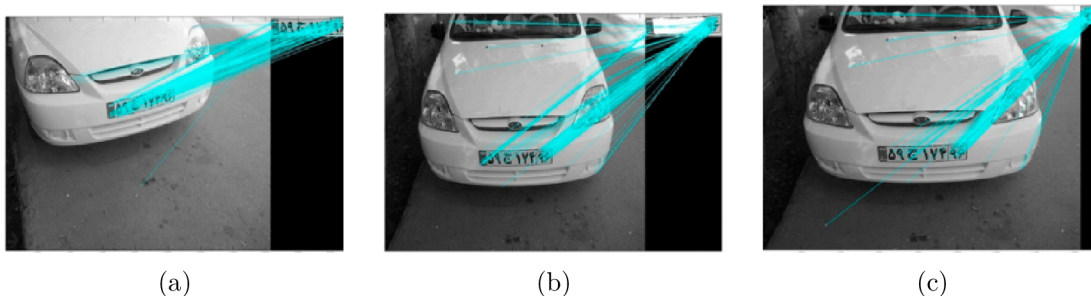
U takto získaných MSER jsou nejprve zjištěny vzájemné geometrické vztahy. To je provedeno tak, že je každý region porovnán s každým, na základě čehož je vytvořen vzájemný geometrický a jasový vztah. Každý takovýto geometrický vztah obsahuje vlastnosti jakými jsou například euklidovská vzdálenost, horizontální a vertikální vzdálenost, rozdíl velikosti jejich *bounding-boxů* (tj. nejmenší možné ohraničující obdélníky), a další. Vzájemný vztah jasu určuje, zda jsou oba regiony tmavé nebo světlé. Na základě těchto vztahů jsou poté hledány takové skupiny regionů, které odpovídají rozložení znaků na registrační značce. Příklad takto nalezených skupin je ukázán na obrázku 2.5b. Výsledkem jsou poté *bounding-boxy* takovýchto skupin.

## 2.2.6 Metody založené na extrakci SIFT příznaků

Méně obvyklou metodu zvolili M. Zahedi a S. M. Salehia [16], kteří pro lokalizaci registrační značky využili extrakce SIFT (*Scale invariant feature transform*) příznaků. Pro použití této metody bylo nejprve potřeba pořídit datovou sadu vzorků registračních značek, ze kterých byly extrahovány klíčové body. Tyto klíčové body byly umístěny do databáze, která později slouží k porovnání s nalezenými příznaky. Při lokalizaci registrační značky je obraz transformován do velké kolekce příznakových vektorů, které jsou nezávislé na velikosti, rotaci a částečně i na jasu obrazu. V databázi jsou poté hledány shodné SIFT příznaky s těmi, které byly extrahovány z obrazu. Hledání probíhá na základě porovnávání euklidovských vzdáleností příznakových vektorů.

Ze shodujících se příznaků je vybrána podmnožina klíčových bodů (ukázáno na obrázcích 2.6). Na základě velikostí této podmnožiny a podmnožiny neúspěšně porovnaných SIFT příznaků je následně vypočteno, s jakou pravděpodobností se jedná o hledaný objekt. Pro zvýšení efektivity celého algoritmu je na obraz aplikována detekce hran, díky které je zúžen počet možných oblastí s registrační značkou.





Obrázek 2.6: Na obrázku (a) je zobrazeno úspěšné rozeznání registrační značky pomocí kompletního vzorku. Na obrázku (b) je ukázáno úspěšné rozeznání za použití překrytého vzorku. Na obrázku (c) je ukázáno použití částečného vzorku taktéž vedoucího k úspěšnému rozeznání. Převzato z [16].

## 2.3 Metody klasifikace znaků

Stejně jako pro lokalizaci registrační značky, existuje také mnoho metod řešící problém klasifikace znaků. Vítaným faktem je to, že registrační značky mají jednotný typ a velikost písma, což celý problém zjednodušuje. Následující kapitoly se budou zabývat některými existujícími metodami řešící problém klasifikace znaků registrační značky.

### 2.3.1 Klasifikace založená na metodě HOG

HOG (*Histogram of Oriented Gradients*) je algoritmus, který dokáže převést obrázek na příznakový vektor. HOG funguje tak, že vypočte histogram gradientní orientace na husté mřížce rovnoměrně rozmístěných buněk a pro zvýšení přesnosti využívá lokální normalizace kontrastu překrývajících se bloků. Tím dokáže efektivně popsat tvar a vzhled objektu. Vzhledem k tomu, že HOG vypočítává histogramy na lokalizovaných buňkách, je tato metoda nezávislá na geometrických i fotometrických transformacích [7].

Algoritmus, který navrhli a implementovali Bo Li a kol. [7], postupuje tak, že jsou nejprve jednotlivé znaky normalizovány. Toho je dosaženo tím, že jsou převedeny na šedotónové a transformovány do jednotného rozměru 20x40 pixelů. Následně je na takto normalizované znaky aplikována metoda HOG s nastavenou velikostí buněk 5x10 pixelů, velikostí bloků 2x2 a s 9-kanálovým histogramem. Z této metody je tak obdržen příznakový vektor o 324 dimenzích.

Pro klasifikaci je nejprve odděleno několik vzorových znaků, které slouží jako trénovací sada. Pro každou třídu znaku je vypočtena střední hodnota  $\mu_i$  příznakových vektorů v trénovací sadě. Takto je získána sada středních hodnot  $\{\mu_1, \mu_2, \dots, \mu_C\}$ , kde  $C$  je počet tříd znaků. Každý další znak je pak klasifikován tak, že je nejprve vypočten HOG metodou jeho příznakový vektor  $\mu$  a následně je odvozena jeho třída  $c$ . To probíhá na základě toho, ke které střední hodnotě třídy má příznakový vektor  $\mu$  nejbližší

$$c = \arg \min_{i=1,2,\dots,C} \|\mu_i - \mu\| \quad (2.1)$$

### 2.3.2 Klasifikace použitím SVM

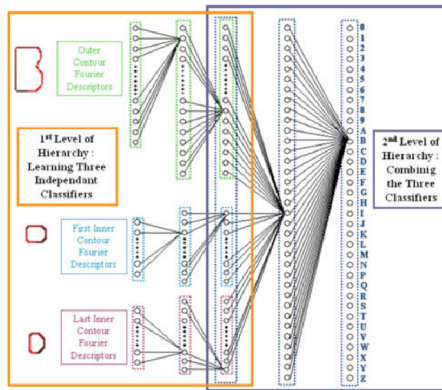
Klasifikátor SVM (*Support Vector Machine*) použili ve své práci Liu Yongchun a Yang Jing [15]. Každý znak je nejprve binarizován operací LoG (*Laplacian of Gaussian*), která znak ztenčí a také eliminuje nežádoucí šum a další rušivé vlivy z prostředí. Binarizovaný znak je také normalizován na jednotnou velikost 18x32 pixelů. Následně je extrahován příznakový vektor znaku. Výpočet tohoto vektoru probíhá na základě struktury znaku a vlastností pixelů. Struktura znaku je vypočtena akumulací několika jeho vlastností, jakými jsou například výchylka inflexního bodu nebo šířka obrysu. Z vlastností pixelů je pro obraz vypočten vektor, který odpovídá odstínu a jasu každého pixelu. Získaný příznakový vektor je ještě redukován na 50 dimenzí algoritmem PCA (*Principal Component Analysis*). Takto zpracovaný příznakový vektor je poté předán klasifikátoru. Pro zvýšení rozpoznávací rychlosti a úspěšnosti je pro algoritmus SVM použita jádrová funkce RBF (*Radial Basis Function*), která je vyjádřena následovně:

$$k(x_i, x_j) = \exp \left\{ - \frac{\|x_i - x_j\|^2}{\delta^2} \right\} \quad (2.2)$$

Pro každou dvojici tříd znaků je natrénován SVM klasifikátor. Při klasifikaci je pak příznakový vektor vyhodnocen všemi klasifikátory a třída s nejvyšší pravděpodobností je zvolena jako výsledná.

### 2.3.3 Klasifikace umělou neuronovou sítí

V další práci, kterou zde zmíním, se Nicolas Thome a kol. [13] zaměřili na klasifikaci znaků pomocí umělé hierarchické neuronové sítě. Pro každý znak jsou nejprve extrahovány Fourierovy deskriptory z jeho vnitřních i vnějších kontur. Nastavením nízké frekvence deskriptorů je získán popis tvaru objektu, který je zároveň odolný proti šumu. Následně je pro klasifikaci použita neuronová síť, jejíž architektura je znázorněna na obrázku 2.7.



Obrázek 2.7: Hierarchická neuronová síť pro klasifikaci znaků. Převzato z [13].

Na první úrovni hierarchie jsou tři perceptrony, které mají za úkol rozeznat vnější konturu a až dvě vnitřní kontury. Druhá úroveň zpracovává výstup tří neuronových sítí v první úrovni a rozhoduje o tom, s jakou mírou pravděpodobnosti patří vstupní znak do které třídy.

## Kapitola 3

# Použité metody nalezení a rozpoznání RZ

Při hledání registračních značek ve videosekvencích se často používá detektor pohybu, aby algoritmus pro lokalizaci značky nemusel analyzovat každý snímek a běžet tak neustále. V této práci jeho použití také umožňuje efektivněji analyzovat pouze aktivní oblast obrazu. Pro lokalizaci registrační značky je použit algoritmus extrakce *Maximally Stable Extremal Regions* a algoritmus Top-Hat. Pro ověření, zda je nalezený region opravdu registrační značkou, je použit lineární klasifikátor *Support Vector Machine*. Následně je pro klasifikaci jednotlivých znaků použita umělá neuronová síť.

### 3.1 Metoda GMM pro detekci pohybu

Pro detekci příjezdu automobilu v obraze ze stacionární kamery je použit GMM (*Gaussian Mixture Model*). Jedná se o jednu z nejpoužívanějších rekurzivních technik pro detekci pohybu. GMM k modelování jasů jednotlivých pixelů používá distribuční funkce normálního rozdělení, označované také jako Gaussovo rozdělení. V průběhu času jsou získávány hodnoty každého pixelu na základě vážené MoG (*Mixture of Gaussians*). Takovýto model pozadí je schopný vyrovnat se s rozmanitostí mnoha praktických situací a vede k dobrým výsledkům při opakovaných pohybech na pozadí, jakými jsou například pohyby listů stromů [3, 18].

Mějme  $x_t$  reprezentující jas libovolného pixelu snímku v čase  $t$ . Za dobu  $T$  pak mohou být intenzity tohoto pixelu vyjádřeny jako  $\{x_1, x_2, \dots, x_t, \dots, x_T\}$  a lze je brát jako směs Gaussovských rozložení. Pro tento pixel pak může být vytvořen GMM s  $K$  prvky vyjádřený následovně:

$$P(x_t) = \sum_{j=1}^K w_j \phi(x_t; \mu_j, \sigma_j) \quad (3.1)$$

kde  $w_j$  je váha jasů pixelu v čase  $j$  a

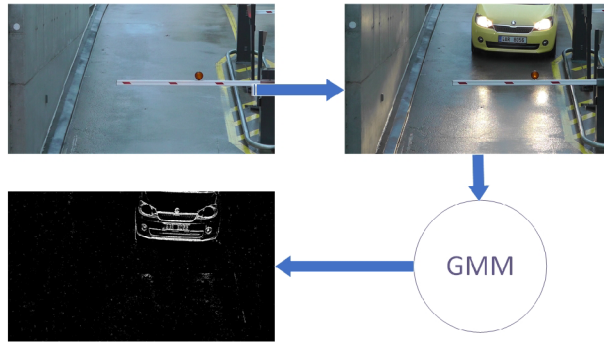
$$\phi(x_t; \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{1}{2}(x_t - \mu_j)^2 \sigma_j^{-2}} \quad (3.2)$$

je Gaussova funkce hustoty pravděpodobnosti se středem  $\mu_j$  a standardní odchylkou  $\sigma_j$ . Váha  $w_j$  je nezáporná a platí, že  $\sum_{j=1}^K w_j = 1$ .

Dalším krokem je rozhodnutí, zda pixel  $x_t$  náleží pohybujícímu se objektu nebo pozadí. V případě GMM se obvykle pixely pohybujících se objektů vyznačují nízkou vahou  $\hat{w}_j$  a velkou odchylkou  $\hat{\sigma}_j^2$ . Nejprve je v GMM vzestupně seřazeno  $K$  prvků na základě hodnoty  $\frac{\hat{w}_j}{\hat{\sigma}_j^2}$ . Poté je zvoleno prvních  $M$  prvků jako pohybující se objekt, kde  $M$  je vyjádřeno jako

$$M = \arg \min_m \left( \sum_{j=1}^m \hat{w}_j > T_m \right), m < K \quad (3.3)$$

kde  $T_m$  je práh, na základě kterého je rozhodnuto, které pixely budou brány jako pixely pohybujícího se objektu. Algoritmus GMM je také schopný odlišit stíny a odrazy světla od pohybujícího se objektu, ty je tak možné ignorovat - jak je tomu například na obrázku 3.1.



Obrázek 3.1: Ukázka výstupu GMM při příjezdu automobilu v dešti.

## 3.2 Metoda extrakce MSER

Při extrakci MSER (*Maximally Stable Extremal Regions*) jsou hledány maximálně stabilní extrémní regiony v obraze. Regiony jsou definovány svou intenzitou a intenzitou na své vnější hranici. Za stabilní lze potom považovat ty, které se napříč různými úrovněmi prahování mění málo. Prahování je prováděno nad všemi úrovněmi intenzity, typicky v rozsahu intervalu  $S = \{0, 1, \dots, 255\}$  [9].

Extrémní jsou ty regiony, které mají buď vyšší nebo nižší intenzitu než je intenzita všech ohraničujících pixelů. To lze vyjádřit matematickým zápisem

$$\forall p \in Q, \forall q \in \partial Q : I(p) > I(q) \vee I(p) < I(q) \quad (3.4)$$

kde  $Q$  je region,  $\partial Q$  je vnější hranice regionu a  $I$  je obrázek. Extrémní regiony jsou vybírány tak, že pixely jsou nejprve seřazeny podle intenzity, a poté jsou vloženy do obrazu. Rychlost řazení je v této části algoritmu, za předpokladu že je intenzita v rozsahu intervalu  $\langle 0, 255 \rangle$ , lineární  $O(n)$  s  $n$  počtem pixelů v obraze. Následně je pomocí tzv. *union-find* algoritmu získán list spojených složek a jejich oblastí. Rychlost *union-find* algoritmu použitého v článku [9] je  $O(n \log \log n)$ , tedy téměř lineární.

Z tohoto procesu je získána struktura obsahující oblast každé pospojované složky. Každá tato oblast je reprezentována jako funkce intenzity. Sloučením dvou složek zanikne ta menší z nich, a její pixely jsou přepokopírovány do té větší. Nakonec jsou vybrány prahové hodnoty

na základě takových úrovní intenzit, které jsou lokálními minimy oblasti složky. Z těch jsou poté získány maximálně stabilní extrémní regiony. Extrémní region  $Q_{i^*}$  je maximálně stabilní, pokud

$$q(i) = |Q_{i \pm \Delta}| / |Q_i| \quad (3.5)$$

má lokální minimum v  $i^*$ .  $\Delta$  určuje, napříč kolika úrovněmi intenzity musí být region stabilní, aby mohl být považován za maximálně stabilní. Platí  $\Delta \in S$ .



Obrázek 3.2: Na obrázku vlevo je zobrazen vstup algoritmu MSER. Na obrázku vpravo jsou bíle vykresleny extrahované maximálně stabilní extrémní regiony.

Jak z výše uvedeného vyplývá, tak metoda MSER je rychlá především při práci nad černobílými obrázky, u kterých se úroveň jasu pohybuje v intervalu  $\langle 0, 255 \rangle$ . Její výhodou je také zachování stability při monotóní změně jasu obrazu [7]. Na obrázku 3.2 je ukázán příklad výstupu extrakce MSER regionů.

### 3.3 Transformace Top–Hat

Transformace Top–Hat dokáže ve snímku zvýraznit malé prvky a objekty a při správném nastavení dokáže rozlišovat vertikální a horizontální hrany. Toho lze využít právě při detekci registračních značek, jelikož obsahují znaky u kterých lze předpokládat, že se dají v rámci snímku považovat za malé a mají velké množství vertikálních hran. Pro tyto účely je nejvýhodnější variantou tzv. Black Top–Hat.

Top-hat využívá základní matematické morfologické operace a uplatňuje se na šedotónových snímcích. Morfologické operace jsou realizovány jako relace obrazu  $f(x, y)$  s jinou menší množinou  $B(u, v)$ , tzv. strukturním elementem. První morfologickou operací, kterou tato transformace používá, je operace dilatace. Dilatace skládá body dvou množin pomocí vektorového součtu a používá se k zaplnění děr [5]. Definiční vztah dilatace vypadá takto:

$$f \oplus B = \max_{(u,v) \in B} (f(x - u, y - v) + B(u, v)) \quad (3.6)$$

Dalším krokem je operace eroze, která je aplikována na výstup dilatace se stejným strukturním elementem  $B$ . Operace eroze je duální k operaci dilatace. Eroze se používá pro zjednodušení struktury objektů. Definiční vztah eroze vypadá takto:

$$f \ominus B = \min_{(u,v) \in B} (f(x + u, y + v) - B(u, v)) \quad (3.7)$$



Dilatace následovaná erozí se nazývá morfologické uzavření. Tato operace vyhladí obrysy, zaplní díry a spojí tak objekty, které jsou blízko u sebe. Definiční vztah morfologického uzavření lze vyjádřit takto:

$$f \bullet B = (f \oplus B) \ominus B \quad (3.8)$$

Pro dovršení Top–Hat transformace už zbývá jenom porovnání výsledného obrázku s tím původním. Porovnávání snímků se provádí jako odečítání úrovní jasu každého pixelu původního obrázku od každého pixelu výsledného snímku. Top–Hat transformaci  $TH$  lze tedy vyjádřit následovně:

$$TH = f \bullet B - f \quad (3.9)$$

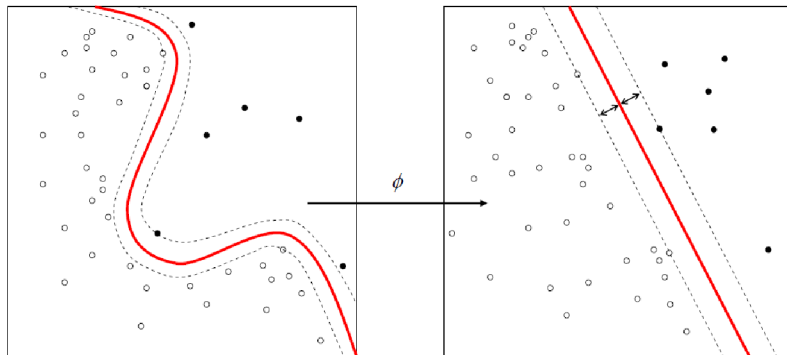
### 3.4 Lineární klasifikátor SVM

Support Vector Machine (dále jen SVM) je široce využívaný klasifikátor fungující na principu statistického strojového učení [14]. Pro účely této práce se bude tato kapitola dále věnovat pouze lineárnímu SVM, které prvky rozděluje do dvou tříd, kde jedna je pozitivní a druhá negativní.

Hlavní myšlenkou je optimální rozdělení objektů pomocí nadroviny. Základním předpokladem proto je, že jsou od sebe objekty obou tříd lineárně oddělitelné. Právě jednou z velkých výhod SVM je, že dokáže pracovat s transformovaným příznakovým prostorem. Jinými slovy, tyto dvě třídy nemusí být lineárně oddělitelné v původním prostoru, ale mohou být oddělitelné v transformovaném prostoru (viz obrázek 3.3). SVM zvládá tuto transformaci implicitně pomocí tzv. jádrových funkcí [6].

$$y(x) = w^T \phi(x) + b \quad (3.10)$$

Jakmile jsou data lineárně separovaná, SVM hledá optimální nadrovinu. Tato nadrovina se poté používá pro klasifikaci a je ve formě normálového vektoru  $w$  a skalárního prahu  $b$ , neboli biasu. Nalezená nadrovina bezchybně odděluje data a má co největší vzdálenost od trénovacích bodů obou tříd. Vstupní vektor je pak klasifikován lineární funkcí vyjádřené rovnicí 3.10, kde  $\phi(x)$  představuje transformaci příznakového prostoru. Trénovací body, které jsou nejbližší k této nadrovině, se nazývají *support vectors*, neboli podpůrné vektory [17]. Viz obrázek 3.3 vpravo.



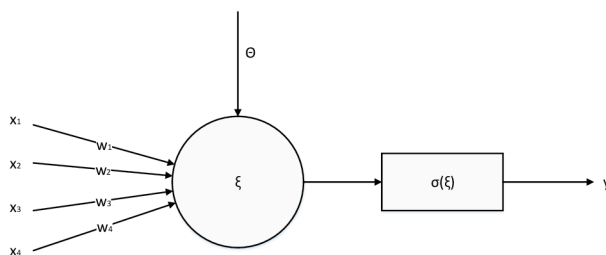
Obrázek 3.3: Levá část obrázku znázorňuje objekty obou tříd v původním prostoru. Pravá část obrázku ukazuje prostor po transformaci a nalezenou optimální nadrovinu [17].

### 3.5 Vícevrstvá umělá neuronová síť a algoritmus zpětného šíření chyb

Obecně je umělá neuronová síť matematický algoritmus se snahou o napodobení fungování lidského mozku. Takovýto algoritmus se především snaží napodobit nejdůležitější vlastnosti mozku, jakými jsou schopnost se učit, uchovávat znalosti a schopnost generalizovat. Umělé neuronové sítě našly své uplatnění napříč nejrůznějšími obory a jejich problematika je velmi rozsáhlá. Proto zde bude omezena pouze na oblast vícevrstevných neuronových sítí používající algoritmus zpětného šíření chyb (angl. error backpropagation). Informace v této kapitole jsou čerpány z [12, 8].

#### 3.5.1 Neuron

Mozek je v zásadě sítí vzájemně propojených neuronů, které si mezi sebou předávají informaci, kterou nějakým způsobem transformují. Každý neuron může mít  $n$  vstupů, ale pouze jediný výstup  $y$ . Model umělého neuronu je znázorněn na obrázku 3.4.



Obrázek 3.4: Model umělého neuronu.

Neuron má vstupy  $x_1, x_2, x_3, \dots, x_n$ , které jsou ohodnoceny svými synaptickými vahami  $w_1, w_2, w_3, \dots, w_n$ . Všechny vstupy jsou váženě sečteny s vahami a je přičten konstantní práh neuronu  $\Theta$ . Tím je získán vnitřní potenciál neuronu  $\xi$ . Aby signál mohl projít neuronem, musí vnitřní potenciál neuronu  $\xi$  překročit hodnotu prahu  $\Theta$ . Vnitřní potenciál může být tedy vyjádřen následovně:



$$\xi = \sum_{i=1}^N (w_i x_i) - \Theta \quad (3.11)$$

Potenciál  $\xi$  je následně transformován aktivační přenosovou funkcí  $\sigma$ . Výstup této funkce je zároveň výstupem neuronu a lze ho vyjádřit jako

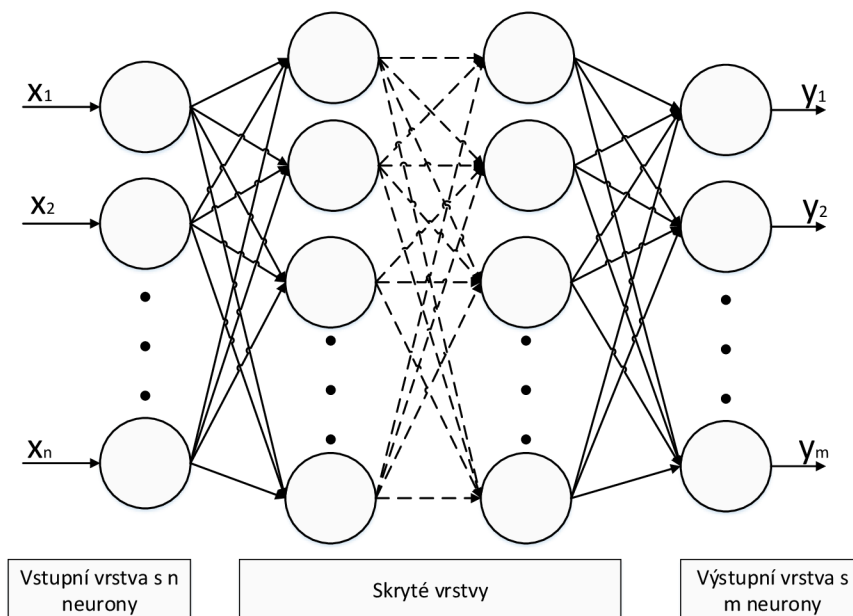
$$y = \sigma(\xi) \quad (3.12)$$

Aktivační přenosová funkce  $\sigma$  je obecně nelineární a transformuje hodnotu vnitřního potenciálu neuronu. Obvykle je od ní očekávána její diferencovatelnost, což umožňuje využití učících algoritmů, které jsou založené na gradientních metodách. Typickým příkladem aktivačních přenosových funkcí jsou logistická sigmoida (viz 3.13) a hyperbolický tangens, a jsou často používány s volitelnou strmostí  $s$ .

$$\sigma(\xi) = \frac{1}{1 + e^{-s\xi}} \quad (3.13)$$

### 3.5.2 Topologie umělé neuronové sítě

Topologii neuronové sítě určuje počet a vzájemné propojení neuronů. Obecně jsou neurony propojeny tak, že výstup z jednoho neuronu je vstupem několika jiných neuronů. Topologie sítě využívající algoritmus zpětného šíření chyb je vícevrstvá a dopředná. Dopředná síť je taková, kde je informace předávána z neuronů nižší vrstvy do neuronů vyšší vrstvy a mezi neurony nejsou žádné zpětné vazby.



Obrázek 3.5: Topologie neuronové sítě algoritmu zpětného šíření chyb [8].

Algoritmus zpětného šíření chyb rozlišuje tři typy vrstev (viz obrázek 3.5). Nejnižší vrstva je vstupní, nejvyšší vrstva je výstupní a mezi nimi se nachází skrytá vrstva. Neurony jsou v takové síti propojeny tak, že výstup jednoho neuronu nižší vrstvy je vstupem všech neuronů vyšší vrstvy a síť tak tvoří úplný orientovaný bipartitní graf. Úkolem vstupní vrstvy je přijmutí vstupního vektoru a delegovat ho do první skryté vrstvy, přičemž počet neuronů vstupní vrstvy je ekvivalentní s počtem dimenzí vektoru. Naproti tomu výstupní vrstva slouží ke zpracování výsledku neuronové sítě. Při použití umělé neuronové sítě jako klasifikátoru je počet neuronů výstupní vrstvy obvykle roven počtu rozlišovaných tříd.

Pro ovlivňování výsledků a kvality neuronové sítě pak slouží skryté vrstvy, jejichž počet a počet jejich neuronů je obvykle volen dle specifického účelu neuronové sítě a je často hledán experimentálně. Pro tento účel existují různé heuristiky.

### 3.5.3 Učení umělé neuronové sítě s učitelem

Učení umělé neuronové sítě, tzv. adaptace, je proces, při kterém jsou upravovány váhy vstupů a hodnoty prahů neuronů za účelem snížit celkovou chybu neuronové sítě na minimum. Učení s učitelem znamená, že jsou algoritmu adaptace na začátku předány správné vzory a výstupy. Z těch je pak počítán rozdíl mezi správným a skutečným výstupem a z toho i celková chyba sítě. Tento typ učení se nazývá akumulované učení, ve kterém se váhy mění až po vyhodnocení reakce na všechny vzory.

Celková chyba sítě udává odchylku mezi hodnotami správných výstupů a hodnotami dosažených výstupů. Pro všechny správné výstupy je pro  $k$ -tý vzor dopočítávána chyba  $E_k$  jako

$$E_k = \frac{1}{2} \sum_{i=1}^m (d_{ik} - y_{ik})^2 \quad (3.14)$$

kde  $m$  je počet výstupních neuronů,  $d$  požadovaná hodnota výstupu z neuronu a  $y$  je skutečný výstup z neuronu. Celková chyba neuronové sítě  $E_c$  se poté počítá jako součet chyb  $E_k$  všech správných vzorů. Vyjádřeno vzorcem  $E_c = \sum_{k=1}^p E_{kp}$ , kde  $p$  je počet správných vzorů.

Úprava váhy  $k$ -tého neuronu  $i$ -tého vstupu v  $t$ -té iteraci pak probíhá na základě delta pravidla

$$w_{ik}(t) = w_{ik}(t-1) - \mu \frac{\delta E_k}{\delta w_{ik}} \quad (3.15)$$

kde  $\mu$  je koeficient učení. Pokud má koeficient učení příliš nízkou hodnotu, celková chyba konverguje k požadované maximální chybě pomalu nebo jí nedosáhne nikdy. V opačném případě, pokud je koeficient příliš vysoký, nemusí být dosaženo konvergence  $E_c$ . Hodnota koeficientu učení se obvykle určuje experimentálně.

## Kapitola 4

# Návrh systému automatického přístupu na parkoviště

Rozpoznávání registračních značek je aktuálním a poměrně dlouho řešeným problémem počítačového vidění. Své uplatnění nalézá v mnoha bezpečnostních, monitorovacích i statistických systémech po celém světě. Tato práce je zaměřena na rozpoznání registrační značky vozidla přijíždějícího ke vstupnímu terminálu parkoviště a rozhodnutí o jeho vpuštění. Vstupem systému je tedy obraz z vhodně umístěné stacionární kamery a systém by měl být v reálném čase schopný detekovat přijíždějící vozidlo, a následně nalézt a rozpoznat jeho registrační značku. Pro tento účel je navržen systém, který se zaměřuje na detekci a rozpoznání registračních značek o nejpoužívanější velikosti 520x110 mm (kapitola 2.1).

### 4.1 Analýza problému

Velký vliv na úspěšnost celého systému má v první řadě kvalita vstupního obrazu. Proto velmi záleží na použité technice, především pak na kvalitě záznamových zařízení. Mezi nejdůležitější parametry patří rozlišení a ostrost obrazu, expoziční čas uzávěrky a světelnost optické soustavy. Pro zvýšení úspěšnosti také mohou být použity kamery s infra přísvitem, jelikož registrační značky jsou reflexivní (viz 2.1) a infračervené světlo odráží. Dalším důležitým parametrem je také umístění záznamového zařízení. Se změnšujícím se úhlem mezi registrační značkou a záznamovým zařízením se snižuje jak úspěšnost lokalizace registrační značky, tak i úspěšnost segmentace a klasifikace znaků. Ideální úhel je  $90^\circ$ , tedy když jsou registrační značka a záznamové zařízení kolmo k sobě. Vhodné umístění kamery také může napomoci eliminaci ruchu v pozadí a zefektivnit tak detekci příjezdu automobilu.

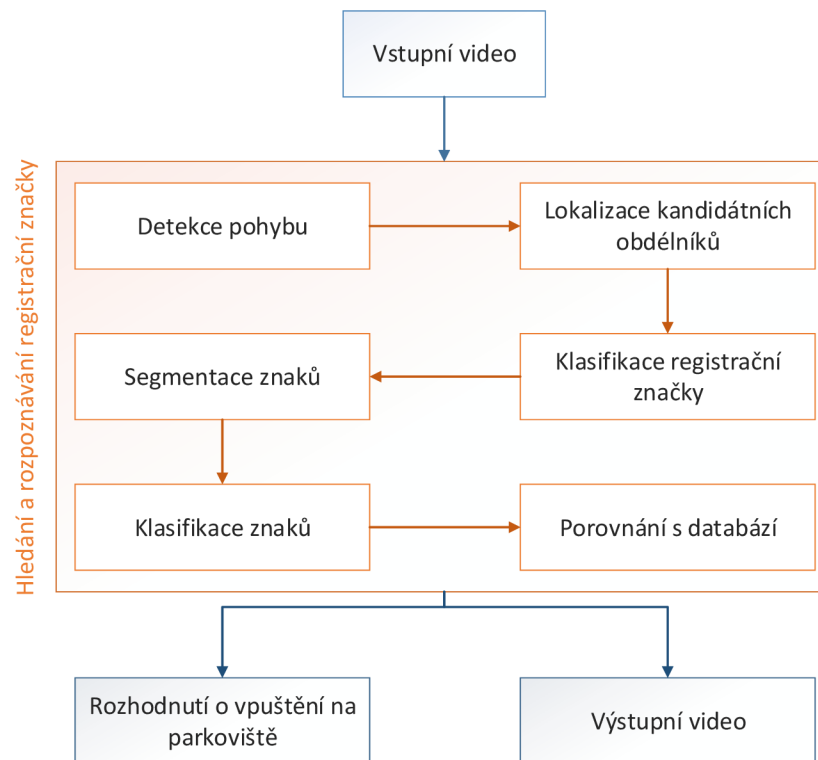
Dalším faktorem, který může ve venkovním prostředí ztížit rozpoznávání registrační značky, je počasí a změna denní doby. Se změnou denní doby si musí poradit především záznamové zařízení, které by se mělo dokázat přizpůsobit světelným podmínkám prostředí. Rušivým elementem však mohou být i jiné zdroje světla. Například oslnění registrační značky nebo záznamového zařízení sluncem, nebo nevhodně vržený stín. U počasí je potřeba vypořádat se především s deštěm nebo slabším sněžením. V případech silně snížené viditelnosti však obvykle dochází k selhání takovýchto systémů.

## 4.2 Základní skladba systému

System se skládá ze čtyř základních částí, které na sebe navazují. Každá z těchto částí obstarává část funkcionality, základní dekompozice je následující:

- Detekce pohybu
- Nalezení registrační značky
- Klasifikace znaků
- Porovnání s databází

Vstupem systému je videosekvence, ve které je detekován pohyb. Ve chvíli, kdy se objeví vozidlo v obraze, je vybrána aktivní oblast s vozidlem. V této oblasti jsou poté nalezeny kandidátní regiony, které by mohly odpovídat registrační značce. U těch je následně rozhodnuto, zda se jedná či nejedná o registrační značku. U regionů představujících registrační značku jsou poté segmentovány znaky. Pokud počet znaků odpovídá platné registrační značce, je provedena klasifikace těchto znaků. Následně je výsledek klasifikace vyhodnocen databází, na základě které je rozhodnuto, zda má být vozidlo vpuštěno na parkoviště. Poté dojde k otevření závory po dobu intervalu  $T$ , po který neprobíhá detekce pohybu ani poznávacích značek. Po uplynutí tohoto intervalu je závora opět zavřena a spuštěna detekce. Tento proces je znázorněn na obrázku 4.1 a podrobně popsán v následujících kapitolách.



Obrázek 4.1: Schéma fungování systému automatického vpuštění na parkoviště.

### 4.3 Detekce příjezdu vozidla

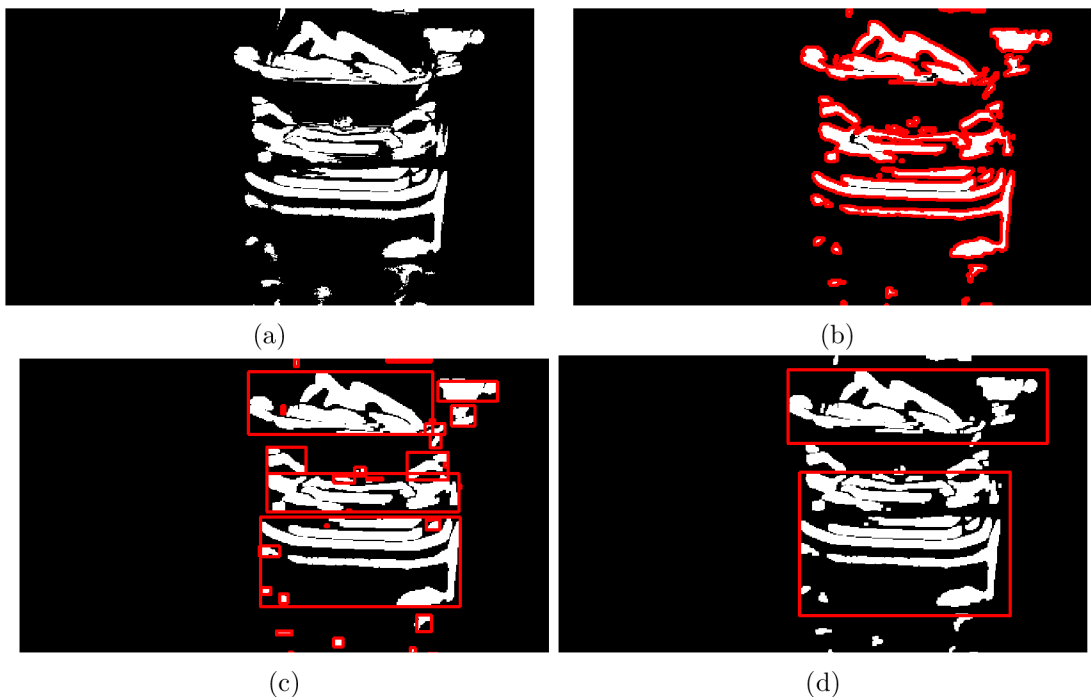
Algoritmus pro lokalizaci registrační značky je poměrně výpočetně náročný a jde o nejpomalejší část aplikace (viz tabulka 5.1). Z tohoto důvodu je využit algoritmus pro detekci pohybu, díky kterému je spuštěna lokalizace registrační značky pouze v potřebný moment. Další výhodou jeho použití je také schopnost vymezit pouze aktivní oblast videa a dále pracovat pouze s touto oblastí. Díky tomu není zpracováván celý snímek, což vede ke zvýšení rychlosti běhu aplikace.

Zpracovávané video je převzorkováno na přibližně 5 FPS. Tím jsou zlepšeny výsledky detekce pohybu, a systém tak není příliš zahlcen lokalizací registrační značky. Díky snížení FPS je tak zpracováván každý  $x$ -tý snímek, kde  $x$  lze vyjádřit tímto vztahem:

$$x = \frac{\text{FPS vstupního videa}}{5} \quad (4.1)$$

Před samotnou detekcí pohybu je každému snímku nejprve změněna velikost tak, aby nepřesahoval šířkou ani výškou 600 pixelů. Tato hodnota byla zvolena experimentálně jako kompromis mezi spolehlivostí detekce a rychlostí algoritmu. Každý snímek je také rozostřen jednoduchým filtrem *Normalized Box* za účelem eliminace detailů, které by mohly být pro algoritmus detekce pohybu rušivé. Následně je na takovýto snímek aplikován algoritmus GMM, který je podrobněji probrán v kapitole 3.1. GMM je nastaveno tak, aby vyhodnocovalo pohyb na základě 3 předchozích snímků. Tato hodnota je také odvozena experimentálně. Algoritmus GMM odstraní stín pod vozidlem a částečně i světlo vržené světlometry příjíždějícího vozidla.

Z algoritmu GMM je získána binární mapa pixelů objektů na popředí (viz 4.2a). Pro eliminaci nežádoucího šumu, který nevyhnutelně vzniká vnějšími vlivy (například deštěm či snežením), jsou na tuto binární mapu následně aplikovány morfologické operace eroze a dilatace. Tím jsou eliminovány drobné irelevantní objekty v binární mapě, ve které jsou poté detekovány kontury (zobrazeny na obrázku 4.2b). Pro každou nalezenou konturu je nalezen nejmenší ohraničující obdélník (viz 4.2c). Příliš malé obdélníky jsou vyfiltrovány a zbylé překrývající se obdélníky jsou sloučeny (viz 4.2d). U výsledných oblastí se pak předpokládá, že obsahují příjíždějící vozidlo a je v nich hledána registrační značka.



Obrázek 4.2: Na obrázku (a) je zobrazena binární mapa popředí získaná z algoritmu GMM. Na obrázku (b) jsou zvýrazněny kontury v binární mapě. Na obrázku (c) jsou vykresleny nejmenší ohraničující obdélníky kontur. Výsledek filtrace a sloučení těchto obdélníků je zobrazen na obrázku (d).

## 4.4 Nalezení registrační značky

U systému pro rozpoznávání registrační značky pro vpouštění na parkoviště je mnohem důležitější jeho přesnost a úspěšnost, než jeho rychlost. Z tohoto důvodu musí být kladen velký důraz na úspěšnou lokalizaci registrační značky. Proto byly navrženy dvě odlišné metody. První z nich je metoda využívající extrakci MSER, zatímco druhá metoda využívá transformaci Top-Hat. Obě tyto metody jsou podrobněji popsány v následujících podsekcích.

Nejprve je potřeba stanovit některé konstanty, ze kterých budou později vycházet některé výpočty. První konstantou je minimální povolený rozměr  $M$  kandidátního obdélníku. Ten byl experimentálně stanoven na šířku 60 pixelů a výšku 13 pixelů, jelikož znaky ve značkách s nižším rozlišením mají nízkou úspěšnost klasifikace. Jelikož jsou vstupem této části systému oblasti, které byly vyhodnoceny detektorem pohybu jako aktivní, je také nutné definovat maximální velikost takovýchto oblastí (šířku i výšku). Tato hodnota ovlivňuje rychlost i úspěšnost celého procesu a byla rovněž zvolena experimentálně, a to na hodnotu 600 pixelů.

### 4.4.1 Předzpracování obrazu

Před zahájením lokalizace registrační značky je nejprve nutné originální obraz převést do stupňů šedi. Pokud velikost oblasti překračuje výškou nebo šířkou maximální velikost 600 pixelů, je velikost výřezu oblasti zmenšena.



#### 4.4.2 Metoda využívající MSER

První navržená metoda, která se pokouší lokalizovat registrační značky na snímku, využívá extrakce MSER (kapitola 3.2). Přesto, že se algoritmus extrakce MSER dokáže efektivně adaptovat zpracovávanému snímku, tak musí být správně nakonfigurován. Při konfiguraci byl experimentálně zvolen parametr  $\Delta$  na hodnotu 3 a také maximální odchylka velikosti oblasti od velikosti podřazených oblastí na 10%. Minimální možná velikost plochy oblasti je nastavena na velikost plochy obdélníku  $M$ , který je definován výše.

Důležitým parametrem je také maximální velikost plochy oblasti  $A_m$ , jelikož značně ovlivňuje rychlost algoritmu extrakce MSER. Pokud je zpracován snímek o velikosti 600 pixelů a hledáme registrační značku o poměru stran  $\sim 4.72$  (kapitola 2.1), musíme zohlednit oblasti o obsahu až  $\sim 76000$  pixelů. Experimentováním se ukázalo (kapitola 5.4.1), že doba zpracování takto velkých oblastí trvá až příliš dlouho. Proto jsem navrhl způsob optimalizace, kterým lze rozdělit extrakci MSER do dvou částí tak, aby každá z nich proběhla co nejrychleji.

Nejprve se na základě šířky obrázku  $I_w$ , minimální šířky registrační značky  $M_w$  a poměru stran  $r$  registrační značky (včetně chyby  $e$ ) vypočte maximální velikost plochy oblasti  $A_m$  jako

$$A_m = I_w * M_w / (r + r * e) \quad (4.2)$$

Chyba byla zvolena experimentálně na 5%,  $A_m$  tak bude  $\sim 7253$  pixelů. Po vypočtení tohoto parametru jsou extrahovány MSER ze snímku. Následně je snímek zmenšen, kdy je jeho nová velikost vypočtena na základě poměru zmenšení  $s$  vypočteného jako

$$s = I_w / \sqrt{I_w * M_w} \quad (4.3)$$

Zmenšením snímku jsou všechny nalezené regiony při první extrakci zmenšeny pod úroveň  $M$ , tudíž při druhé extrakci nebudou zvažovány. Z takto zmenšeného snímku jsou opět extrahovány MSER za použití stejných parametrů. Tyto regiony jsou pak přetransformovány na původní velikost na základě poměru  $s$ . Na obrázcích 4.3a a 4.3b je zobrazen případ, kdy je registrační značka detekována až při druhé extrakci MSER po zmenšení snímku, jelikož značka je v originálním snímku příliš velká.

#### 4.4.3 Metoda využívající transformace Top–Hat

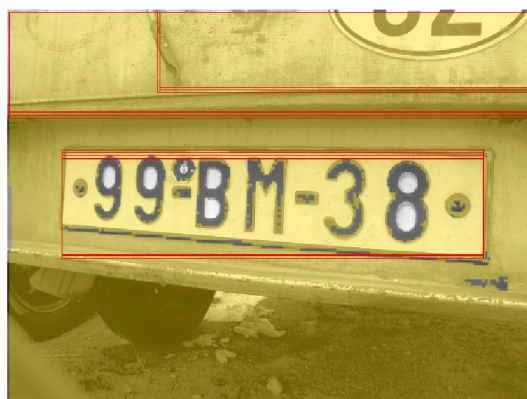
Druhá použitá metoda využívá transformaci Top–Hat. Tato metoda využívá toho, že registrační značka obsahuje velké množství vertikálních hran, které jsou blízko u sebe a na kontrastním pozadí. Díky tomu mohou být nalezeny i ty značky, jejichž obrys není zřetelně rozeznatelný a kde by mohla metoda využívající extrakce MSER selhat. Transformace Top–Hat sestává z chronologické posloupnosti následujících morfologických operací: dilatace, eroze a subtrakce od původního snímku (viz 3.3).

Pro dilataci a erozi je jako strukturální element zvolen obdélník o rozměrech  $9 \times 3$  pixelů. Poté je na černobílý obrázek aplikována operace dilatace a následně operace eroze. Tím je dosaženo morfologického uzavření, díky kterému jsou odstraněny detaily z obrazu, obraz je vyhlazený a všechny blízké objekty jsou spojeny. Na obrázku 4.4b je vidět výsledek morfologického uzavření.





(a)



(b)

Obrázek 4.3: Na obrázcích je ukázán výstup extrakcí MSER (žlutě) a zvýrazněny kandidátní obdélníky (červeně). Na obrázku (a) je zobrazen výsledek první extrakce, na obrázku (b) výsledek druhé extrakce po zmenšení snímku.



(a)



(b)

Obrázek 4.4: Na obrázku (a) je vstupní snímek. Na obrázku (b) je výsledek morfologického uzavření.

Pro dovršení transformace Top-Hat je výsledná eroze porovnána s původním černobílým snímkem, čímž jsou na původním snímku zvýrazněny všechny detaily, které morfologické uzavření odstranilo. Na obrázku 4.5a lze vidět zvýrazněné znaky na registrační značce.



(a)

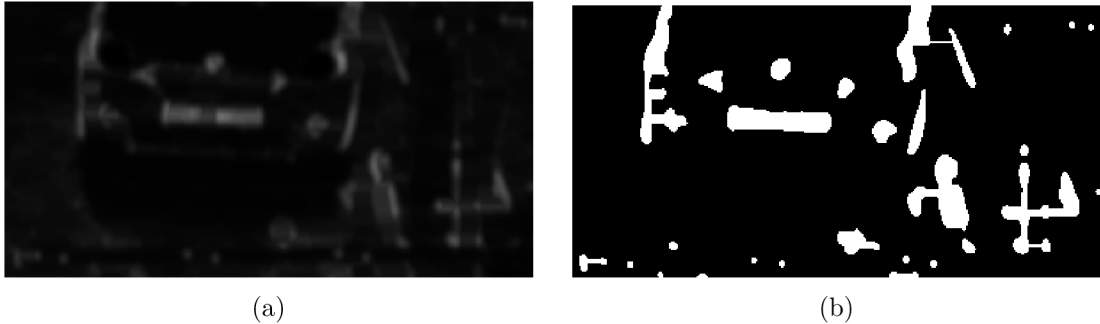


(b)

Obrázek 4.5: Obrázek (a) je výsledek transformace Top-Hat. Na obrázku (b) je ukázán výsledek použití Sobel operátoru.

Aby mohla být vybrána celá oblast s registrační značkou, je nejprve potřeba zvýrazněné

části rozšířit. Nejprve je aplikován operátor Sobel v horizontálním směru (viz 4.5b), načež je aplikováno Gaussovské rozostření. Na takto upraveném snímku je opět provedeno morfologické uzavření se stejným strukturálním elementem. Tímto postupem je již zvýrazněna celá oblast registrační značky, jak lze vidět na obrázku 4.6a. Výsledný snímek je nakonec metodou Otsu prahován (obrázek 4.6b) a v takto vyprahovaném snímku jsou vyhledány kontury.



Obrázek 4.6: Obrázek (a) ukazuje zvýraznění celé registrační značky. Na obrázku (b) je výsledek prahování metodou Otsu.

#### 4.4.4 Zpracování výsledných kontur

Z obou výše navržených metod je získáno velké množství kontur. Nejprve jsou nalezeny obdélníky (obvykle rotované) s co nejmenší velikostí tak, aby ohraničovaly celou konturu. V případě druhé metody jsou ještě pospojovány takové obdélníky, které se nachází na přibližně stejné horizontální úrovni a mají přibližně stejné rozměry. Tím je vyřešen problém, kdy kvůli mezeře mezi znaky v registrační značce není vytvořen obdélník pokrývající celou oblast značky. Obdélníky jsou poté profiltrovány a zbudou pouze takové, které svou velikostí a poměrem stran odpovídají registrační značce a zároveň nepřesahují okraj snímku. U všech zbylých je provedena transformace perspektivy. Tím je z rotovaného obdélníku získán nerotovaný výřez z původního snímku, jak je znázorněno na obrázku 4.7a. Příklad výstupu této části programu je ukázán na obrázku 4.7b.



Obrázek 4.7: Obrázek (a) ukazuje zvýraznění celé poznávací značky. Na obrázku (b) jsou příklady výstupu lokalizace registrační značky.

#### 4.4.5 Klasifikace nalezených kandidátů

Pro rozhodnutí, zda se jedná či nejedná o registrační značku, byl zvolen lineární klasifikátor SVM (kapitola 3.4), jehož velkou výhodou je vysoká rychlost klasifikace. Pro účely klasifikace je každý výřez zmenšen na pevně určený rozměr 156 x 33 pixelů. Nakonec je výřez převeden do stupňů šedi. Těmito kroky je docíleno toho, že bude mít vstupní vektor SVM klasifikátoru vždy stejnou velikost. Výsledky úspěšnosti klasifikátoru jsou vypsány v kapitole 5.4.2.

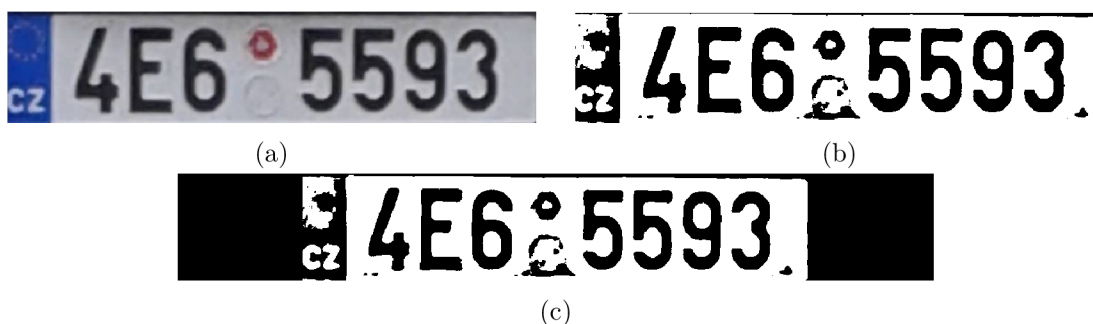
### 4.5 Rozpoznání registrační značky

Rozpoznávání registrační značky je rozděleno do dvou základních kroků. Prvním krokem je segmentace znaků v registrační značce. Druhým krokem je pak klasifikace znaků získaných v prvním kroku. Protože zde bude vždy existovat možnost mylné klasifikace SVM, je po segmentaci registrační značky dodatečně rozhodováno, zda se jedná o registrační značku na základě počtu znaků. Pokud počet znaků neodpovídá správnému rozsahu (kapitola 2.1), je chybný výřez zahozen. Tím je minimalizován počet falešně pozitivních vzorků.

#### 4.5.1 Předzpracování registrační značky

Výstup lokalizačních metod není vždy zcela přesný, proto je nejprve nutné vstup náležitě upravit a následně převést do požadované podoby. Následující navržené úpravy jsou často důležitým předpokladem pro úspěšnou segmentaci i následnou klasifikaci znaků.

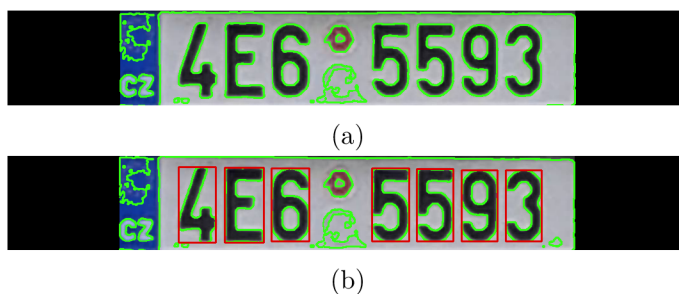
Ze všeho nejdříve je vstupní výřez převeden na šedotónový. Ten je poté binarizován použitím adaptivního prahování. Použitím dostatečně velkého okolí pixelu při prahování nevzniká ve výsledném binárním snímku téměř žádný šum (viz 4.8b). Velikost okolí je vypočtena na základě výšky výřezu a kvůli souměrnosti je vždy lichá. Binárnímu snímku je poté přidán černý okraj z obou stran o velikosti čtvrtiny šířky registrační značky (viz 4.8c). Díky tomu v některých případech splyne modrý pruh evropské unie nebo část rámečku registrační značky, které by jinak mohly být považovány za znak. Jako další je přidán bílý dolní a horní okraj široký 2 pixely, který oddělí znaky přilehlé k okraji výřezu od okraje.



Obrázek 4.8: Obrázek (a) ukazuje vstupní výřez registrační značky. Na obrázku (b) je zobrazen výsledek adaptivního prahování. Na obrázku (c) je binární snímek s rozšířeným okrajem.

### 4.5.2 Segmentace znaků

Navržená segmentace znaků vyhledává kontury v binarizovaném výřezu (viz 4.9a). Pro každou nalezenou konturu je získán její *bounding-box*, u kterého je ověřen správný poměr stran a správná velikost tak, aby mohl odpovídat znaku registrační značky. Pokud proporce *bounding-boxu* neodpovídají proporcím znaku, je kontura zahozena. Ostatní jsou řazeny dle své x-ové souřadnice zleva doprava. Pokud je nalezeno méně znaků než je požadované minimum znaků na registrační značce, je segmentace ukončena a výřez je zahozen. Na obrázku 4.9b jsou zobrazeny správně nalezené znaky.



Obrázek 4.9: Na obrázku (a) jsou zeleně vykresleny nalezené kontury. Na obrázku (b) jsou červeně zobrazeny výsledné *bounding-boxy*.

### 4.5.3 Klasifikace znaků

Navržený systém klasifikace znaků registrační značky využívá vícevrstvou umělou neuronovou síť s algoritmem zpětné propagace chyb. Pro aktivační přenosovou funkci byla zvolena logistická sigmoida (kapitola 3.5) a vytvořená neuronová síť obsahuje 100 skrytých vrstev. Po segmentaci registrační značky jsou znaky již binarizované a je potřeba je normalizovat a vytvořit jejich příznakový vektor. Pro normalizaci je nejprve výřez znaku transformován na čtverec tak, aby byl zachován poměr stran výřezu. Pro tento účel je pro výřez  $V$  sestavena transformační matice  $M$  dle vzorce 4.5.

$$m = \max(V_{\text{šířka}}, V_{\text{výška}}) \quad (4.4)$$

$$M = \begin{bmatrix} 1 & 0 & \frac{m}{2} - \frac{V_{\text{šířka}}}{2} \\ 0 & 1 & \frac{m}{2} - \frac{V_{\text{výška}}}{2} \end{bmatrix} \quad (4.5)$$

$$N(x, y) = V (M_{11}x + M_{12}y + M_{13}, M_{21}x + M_{22}y + M_{23}) \quad (4.6)$$

Pomocí takto sestavené transformační matice je poté výřez transformován na čtverec  $N$  dle vzorce 4.6. Velikost čtverce je následně změněna na pevně danou velikost 18 x 18 pixelů. Na obrázku 4.10 je zobrazen příklad normalizovaných znaků.



Obrázek 4.10: Normalizované znaky pro klasifikaci.

Vytvoření příznakového vektoru probíhá ve dvou základních krocích. V prvním je čtverec rozdělen na rovnoměrné čtvrtiny (viz 4.11). Pro každou jeho čtvrtinu je následně vytvořena

vertikální i horizontální projekce. Projekce je vytvořena tak, že je vypočten počet bílých pixelů v každém řádku (horizontální projekce) a v každém sloupci (vertikální projekce). V druhém kroku je do vektoru ještě přidán čtverec zmenšený na polovinu původní velikosti.



Obrázek 4.11: Znázorněné rozdělení čtverce na čtvrtiny, pro které je poté vytvořena vertikální a horizontální projekce.

## Kapitola 5

# Implementace a vyhodnocení dosažených výsledků

V této kapitole se zaměřím pouze krátce na implementaci, jelikož všechny podstatné informace byly zmíněny již v předchozích kapitolách. Hlavní zaměření této kapitoly bude především na vytvořené datové sady, principy trénování SVM a umělé neuronové sítě, a v neposlední řadě na dosažené výsledky. Všechny následující experimenty byly prováděny na operačním systému Windows 10, Intel(R) Core(TM) i7-3520M CPU @ 2.90 GHz, 8 GB RAM.

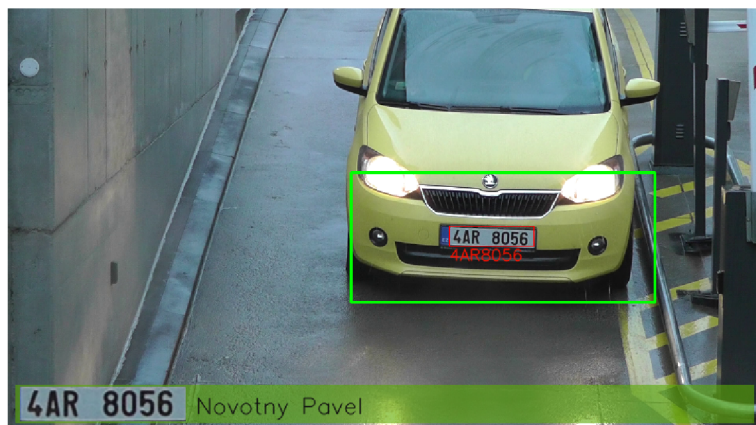
### 5.1 Vytvořený program

Pro účely této práce byl vytvořen program umožňující rozpoznávání registračních značek ze vstupního video souboru nebo ze vstupního obrázku. Program umožňuje grafickou reprezentaci výstupu se zobrazením výsledku lokalizace i rozpoznání registrační značky, včetně verdiktu o vpuštění na parkoviště. Zároveň byla vytvořena jednoduchá databáze, která sestává ze dvou souborů `whitelist` a `blacklist`. V prvním z nich jsou uvedeny záznamy s registračními značkami, které mají přístup na parkoviště. Ve druhém jsou uvedeny registrační značky, které vjezd na parkoviště mají zakázaný. Data jsou v těchto souborech reprezentována tak, že každý řádek reprezentuje jeden záznam. Každý záznam je složen z čísla registrační značky a jména příslušné osoby.

Pro vypracování softwarové aplikace byl zvolen programovací jazyk C++, a to především z důvodu jeho rychlosti. Jako vývojové prostředí bylo použito Microsoft Visual Studio od společnosti Microsoft Corporation. Aplikace byla cílena na platformy Windows a Linux. Zdrojové kódy aplikace byly vytvořeny dle standartu C++11.

Pro práci s multimediálními soubory, algoritmy počítačového vidění a strojovým učením byla použita knihovna OpenCV ve verzi 3.1.0. OpenCV (Open Source Computer Vision) je otevřená knihovna počítačového vidění spadající pod licenci BSD. Byla navržena pro výpočetní efektivitu a je optimalizována především pro jazyky C a C++.





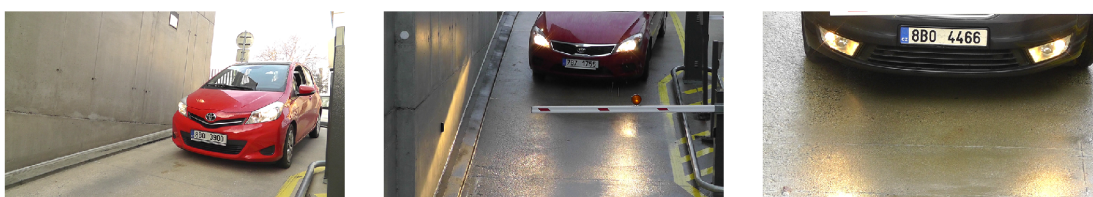
Obrázek 5.1: Ukázka grafického výstupu aplikace.

## 5.2 Datová sada

Pro účely trénování algoritmů strojového učení byla vytvořena datová sada čítající asi 1400 fotografií registračních značek. Kromě toho bylo ze třech různých úhlů natočeno celkem 24 automobilů projíždějících na parkovišti. Pro natáčení byla použita kamera Panasonic HDC-SD90 a byla pořízena videa s rozlišením 1920x1080 pixelů.



Obrázek 5.2: Ukázka pořízených fotografií do datové sady.



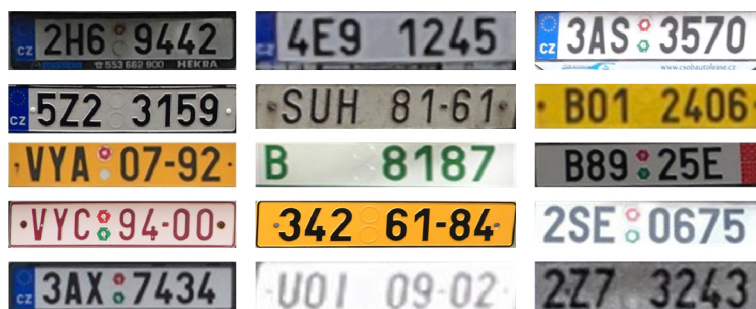
Obrázek 5.3: Ukázka z pořízených videozáznamů ze tří různých úhlů.

## 5.3 Trénování algoritmů strojového učení

Pro trénování klasifikátorů je použita část vytvořené datové sady fotografií. Oba klasifikátory potřebují ke svému učení učitele. Pro trénování klasifikátoru SVM jsou z datové



sady vytvořeny pozitivní a negativní vzorky. Pozitivní vzorky obsahují pouze výřez registrační značky. Naproti tomu negativní vzorky představují obrázky bez registračních značek. V tomto případě byly z fotografií v datové sadě extrahovány regiony s registrační značkou. Zatímco výřez registrační značky byl uložen do samostatného souboru (viz 5.4), v originálním snímku byla značka nahrazena černým obdélníkem (viz 5.6), čímž vznikl snímek bez registrační značky.



Obrázek 5.4: Ukázka pozitivních vzorků.



Obrázek 5.5: Ukázka vytvořených negativních vzorků.

Pro trénování umělé neuronové sítě pro klasifikaci znaků jsou pak z registračních značek extrahovány znaky, které jsou uloženy do příslušného adresáře - pro každý znak existuje oddělený adresář, do kterého se ukládají znaky stejného typu. Znaky jsou ukládány v originální podobě a normalizovány jsou až při trénování. Koeficient učení neuronové sítě byl nastaven na 0,025.



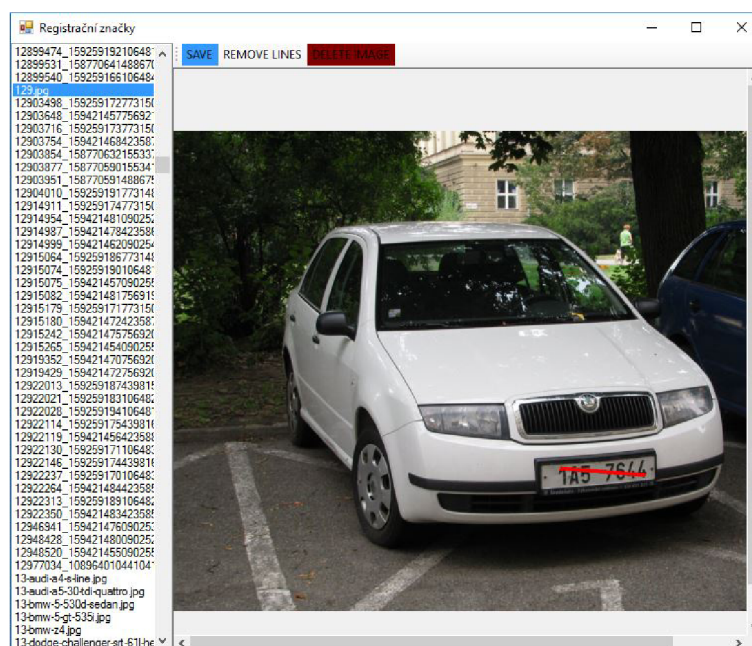
Obrázek 5.6: Ukázka vytvořených vzorků znaků registrační značky.

## 5.4 Vyhodnocení dosažených výsledků

V následujících kapitolách jsou ukázány výsledky jednotlivých experimentů provedených ve výsledném programu. Pro účely experimentů byla oddělena část datové sady, a to taková, která nebyla využita pro trénování algoritmů strojového učení.

### 5.4.1 Vyhodnocení úspěšnosti lokalizace registračních značek

Aby mohlo být automaticky ověřeno, zda byla registrační značka úspěšně nalezena, byla pro tento experiment vytvořena pomocná aplikace v prostředí WinForms. Pomocí této aplikace pak byly na testovacích snímcích vyznačeny dva body. Tyto body byly umístěny tak, aby nad nimi sestrojený obdélník pokrýval oblast se všemi znaky registrační značky (obrázek 5.7).



Obrázek 5.7: Ukázka aplikace pro vyznačení registračních značek.

Vyhodnocení úspěšnosti lokalizace probíhalo na datové sadě o 1054 snímcích s celkem 1062 vyznačenými značkami. Experiment byl proveden na různých kombinacích použitých metod lokalizace registrační značky. Dosažené výsledky jsou znázorněny v tabulce 5.1. Vý-

sledky se liší v závislosti na použitých metodách lokalizace. Lze tak snadno porovnat vliv a účinnost navržených metod. Průměrná velikost obrázků použitých pro testování byla 851x778 pixelů.

Použité metody	Časová náročnost	Úspěšnost
Samotný MSER	604,5 ms	87,48%
Samotný Top-Hat	9,9 ms	57,82%
Optimalizovaný MSER	77 ms	87,48%
Optimalizovaný MSER + Top-Hat	85 ms	92,47%

Tabulka 5.1: Tabulka s výsledky experimentů na metodách lokalizace registrační značky.

#### 5.4.2 Vyhodnocení úspěšnosti klasifikace kandidátních regionů

Vyhodnocení úspěšnosti klasifikátoru SVM probíhalo na podobných datech jako jeho trénování (kapitola 5.3). Z datové sady byla oddělena část fotografií, které nebyly použity při trénování. Z nich byly opět vytvořeny pozitivní a negativní snímky. Nad pozitivními snímky byla spouštěna klasifikace rovnou a byl očekáván vždy kladný výsledek. U negativních snímků byl nejprve spuštěn algoritmus lokalizace registračních značek, ze kterého byly získány kandidátní regiony. Nad všemi těmito regiony byla spuštěna klasifikace a byl očekáván vždy negativní výsledek. Výsledky experimentu jsou uvedeny v tabulce 5.2.

Typ	Vzorků	Časová náročnost	Úspěšnost
Pozitivní vzorky	702	0,009 ms	92,88%
Negativní vzorky	10011	0,28 ms	64,10%
Celkem	10713	0,27 ms	65,99%

Tabulka 5.2: Tabulka s výsledky experimentů s klasifikátorem SVM.

Jak lze vyčíst z výsledků v tabulce 5.2, klasifikátor dosahuje poměrně vysoké míry falešně pozitivních výsledků (přibližně 35,9%). Jelikož je však následně prováděna kontrola správného počtu znaků, je posléze úspěšně vyloučeno 99,87% falešně pozitivních regionů (tj. těch, které neobsahují registrační značku).

#### 5.4.3 Vyhodnocení úspěšnosti segmentace a klasifikace znaků

Pro vyhodnocení úspěšnosti segmentace znaků byla vytvořena zvlášť sada s 677 registračními značkami. Každý soubor v této sadě obsahuje pouze výřez s registrační značkou a jeho název odpovídá textu registrační značky. Na těchto souborech byla následně spuštěna segmentace znaků. Pokud počet znaků odpovídal délce názvu souboru, segmentace byla prohlášena za úspěšnou. Po úspěšné segmentaci následovalo vyhodnocení úspěšnosti klasifikátoru. Znak každého klasifikovaného segmentu byl porovnán s odpovídajícím znakem v názvu souboru. Na základě tohoto porovnání byla také vytvořena matice záměn, kterou lze nalézt v Příloha A. Ostatní výsledky experimentu lze nalézt v tabulce 5.3.

Typ	Vzorků	Časová náročnost	Úspěšnost
Segmentace	684	0,45 ms	91,67%
Klasifikace	4352	0,50 ms	90,03%

Tabulka 5.3: Tabulka s výsledky experimentů segmentace a rozpoznání registrační značky.

#### 5.4.4 Vyhodnocení spolehlivosti vpouštění na parkoviště

Pro tento experiment byla použita pořízená videa. Jak již bylo zmíněno v kapitole 5.2, byla pořízena videa ze tří pohledů - ze strany, shora a zepředu. Registrační značky všech 24 projíždějících automobilů byly přidány na seznam povolených registračních značek. Experiment byl považován za úspěšný právě tehdy, když byl příjezdějícímu automobilu povolen vjezd. Výsledky experimentů jsou uvedeny v tabulce 5.4.

Typ	Průjezdů	Vpuštěno	Úspěšnost
Pohled ze strany	15	14	93,75%
Pohled shora	4	4	100%
Pohled zepředu	5	5	100%
Celkem	24	23	95,84%

Tabulka 5.4: Tabulka s výsledky experimentu vpouštění automobilů na parkoviště.

## Kapitola 6

# Závěr

Cílem této práce bylo navrhnout a implementovat systém pro automatické vpouštění na parkoviště na základě rozpoznání registrační značky. Vytvořený systém je schopný detekovat příjezd vozidla, v aktivní části obrazu nalézt registrační značku, v takto nalezené značce segmentovat znaky a ty poté rozpoznat. Systém byl implementován v jazyce C++ za použití knihovny OpenCV. Při vývoji jsem dbal na schopnost systému zpracovat video v reálném čase s důrazem na spolehlivé nalezení registrační značky.

Před započítím práce jsem se seznámil se základy zpracování obrazu, a v průběhu práce jsem své znalosti neustále rozšiřoval. Pro účely práce jsem nastudoval již existující techniky detekce a rozpoznání registrační značky a podrobněji jsem se seznámil se souvisejícími algoritmy počítačového vidění a strojového učení. Na základě nastudovaných metod jsem vytvořil návrh systému, který jsem následně implementoval. V průběhu vývoje jsem aplikaci průběžně ladil na základě prováděných experimentů.

Na výsledné aplikaci bylo provedeno několik experimentů uvedených v kapitole 5. Z těch lze usoudit, že výsledný systém dosahuje uspokojivých výsledků. Například na pořízených videozáznamech simulujících reálné použití bylo dosaženo úspěšnosti 95.84%. Byla také navržena metoda pro zrychlení extrakce MSER, která tento algoritmus urychlila až sedmkrát. Pro prezentační účely bylo vytvořeno video prezentující fungování navrženého systému.

Tento systém má velký potenciál pro budoucí vývoj vzhledem k široké škále využití a možností. Do budoucna by systém mohl být rozšířen o některé statistické funkce, evidenci doby parkování a případné účtování za parkování. Další užitečnou funkcí by mohlo být předvídání vytížení parkoviště na základě dlouhodobého sběru dat. S tím by také mohla souviset schopnost spolupráce s navigačními systémy, které by mohly doporučit parkoviště na základě tohoto statistické předpokladu nebo přímo na základě aktuální obsazenosti parkoviště.



# Literatura

- [1] AL-GHAILI, A. M.; MASHOHOR, S.; RAMLI, A. R.: Vertical-Edge-Based Car-License-Plate Detection Method. *IEEE Transactions on Vehicular Technology*, ročník 62, č. 1, 2013: s. 26–38, ISSN 0018-9545.
- [2] BABU, C. N. K.; NALLAPERUMAL, K.: A license plate localization using morphology and recognition. In *2008 Annual IEEE India Conference*, ročník 1, 2008, ISSN 2325-940X, s. 34–39.
- [3] BARNICH, O.; DROOGENBROECK, M. V.: ViBe: A Universal Background Subtraction Algorithm for Video Sequences. *IEEE Transactions on Image Processing*, ročník 20, č. 6, 2011: s. 1709–1724, ISSN 1057-7149.
- [4] DUAN, T. D.; DUC, D. A.; DU, T. L. H.: Combining Hough transform and contour algorithm for detecting vehicles' license-plates. In *Intelligent Multimedia, Video and Speech Processing, 2004. Proceedings of 2004 International Symposium on*, 2004, s. 747–750.
- [5] Gleb V. TCHESLAVSKI: Morphological Image Processing: Grayscale morphology. 2010-04-22 [cit. 2010-04-22], [Online; navštíveno 20.4.2016].  
URL <http://www.ee.lamar.edu/gleb/dip/10-3%20-%20Morphological%20Image%20Processing.pdf>
- [6] KIM, H. S.; SOHN, S. Y.: Support vector machines for default prediction of SMEs based on technology credit. *European Journal of Operational Research*, ročník 201, č. 3, 2010: s. 838 – 846, ISSN 0377-2217.
- [7] LI, B.; TIAN, B.; YAO, Q.: A vehicle license plate recognition system based on analysis of maximally stable extremal regions. In *Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on*, IEEE, 2012, ISBN 978-1-4673-0388-0, s. 399–404.
- [8] MACEČEK, A.: Rychlost učení vícevrstvé sítě. 2011.
- [9] MATAS, J.; CHUM, O.; URBAN, M.: Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, ročník 22, č. 10, 2004: s. 761–767, ISSN 0262-8856, british Machine Vision Computing 2002.
- [10] MINISTERSTVO DOPRAVY, ČR: Předpis č. 343/2014 Sb. Vyhláška o registraci vozidel. [Online; navštíveno 19.3.2016].  
URL <https://www.zakonyprolidi.cz/cs/2014-343>



- [11] POON, J. C. H.; GHADIALI, M.; MAO, G. M. T.: A robust vision system for vehicle licence plate recognition using grey-scale morphology. In *Industrial Electronics, 1995. ISIE '95., Proceedings of the IEEE International Symposium on*, ročník 1, 1995, s. 394–399.
- [12] RUBIO, J.; ANGELOV, P.; PACHECO, J.: Uniformly Stable Backpropagation Algorithm to Train a Feedforward Neural Network. *IEEE Transactions on Neural Networks*, ročník 22, č. 3, 2011: s. 356–366, ISSN 1045-9227.
- [13] THOME, N.; VACAVANT, A.; ROBINAULT, L.: A cognitive and video-based approach for multinational License Plate Recognition. *Machine Vision*, ročník 22, č. 2, 2011: s. 389–407, ISSN 09328092.
- [14] VAPNIK, V. N.: *The nature of statistical learning theory*. New York : Springer, 1999, ISBN 978–1441931603.
- [15] YONGCHUN, L.; JING, Y.: Research of license plate character features extraction and recognition. In *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, 2012, s. 2154–2157.
- [16] ZAHEDI, M.; SALEHI, S. M.: License plate recognition system based on SIFT features. *Procedia Computer Science*, ročník 3, 2011: s. 998 – 1002, ISSN 1877–0509, world Conference on Information Technology.
- [17] ZAREMBA, W.; BLASCHKO, M. B.; MUDIGONDA, P. K.: Modeling the Variability of Eeg/meg Data through Statistical Machine Learning Supervisor's Statement Subject Classification. 2012.
- [18] ZHANG, X.; ABE, M.; KAWAMATA, M.: Motion detection in old film sequences using adaptive Gaussian mixture model. In *2011 18th IEEE International Conference on Image Processing*, 2011, ISSN 1522-4880, s. 2337–2340.

# Přílohy

## Seznam příloh

<b>A</b>	<b>Matice záměn klasifikace znaků</b>	<b>38</b>
<b>B</b>	<b>Manuál</b>	<b>39</b>
	B.1 Kompilace programu . . . . .	39
	B.2 Použití programu . . . . .	39

# Příloha A

## Matice záměn klasifikace znaků

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	H	I	J	K	L	M	N	P	R	S	T	U	V	X	Y	Z					
0	405	0	1	0	0	1	0	0	0	0	0	0	1	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0				
1	0	257	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
2	0	0	273	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0				
3	0	0	3	357	0	0	0	0	0	0	0	0	0	0	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	387	3	0	0	2	0	8	0	1	0	1	0	2	0	0	2	0	0	1	0	0	0	2	0	0	0	0	0	0	1			
5	0	3	0	0	1	309	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
6	9	0	6	1	2	5	310	0	4	1	0	6	1	0	1	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	1			
7	0	1	10	0	0	1	305	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1			
8	1	0	0	0	0	0	0	284	11	0	27	0	0	0	1	7	0	0	0	0	0	2	0	2	2	1	0	0	0	0	0	0	0	0			
9	0	0	0	0	1	0	0	0	0	227	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
A	8	0	1	2	1	5	1	0	14	1	108	11	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
B	1	0	0	0	1	0	0	1	0	0	0	126	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
C	0	4	0	0	0	0	0	0	1	0	0	0	31	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0		
D	1	0	2	0	0	0	0	0	1	0	0	5	0	9	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	
E	0	0	1	0	0	0	0	0	0	0	0	0	0	0	50	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
F	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
H	3	0	0	0	1	0	2	1	0	1	0	0	1	1	0	0	49	0	0	0	0	1	0	0	0	0	1	0	4	0	0	0	0	0	0		
I	0	31	0	0	0	0	0	0	1	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
J	0	0	1	5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	
K	1	0	8	0	0	0	0	1	0	2	0	7	0	3	0	0	0	0	0	15	0	1	0	1	4	0	0	0	2	3	0	0	0	0	0		
L	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
N	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	3	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	
P	1	0	0	0	0	1	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	
R	1	0	10	0	0	0	1	0	0	0	1	3	0	0	1	0	0	0	0	0	0	1	0	0	0	5	0	0	0	0	0	0	0	0	0	0	
S	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	
T	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	0	0	1	0	0		
U	2	0	1	0	0	1	1	0	2	0	0	0	0	0	0	0	1	1	0	0	0	2	0	1	0	0	0	43	0	0	0	0	1	0	0	1	
V	2	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	2	9	0	0	0	0	0	0	0	
X	0	2	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0		
Y	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	
Z	1	0	25	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0

# Příloha B

## Manuál

Jelikož byl aplikační software této práce vyvinut pro Windows i Linux, bude zde uveden návod k instalaci pro obě tyto platformy. Základní prerekvizitou pro zprovoznění programu na Linuxu je mít nainstalovanou knihovnu OpenCV ve verzi 3.1.0 a program CMake ve verzi alespoň 2.6.0. Pro Windows jsou přibaleny soubory této knihovny k projektu, je ovšem potřeba mít nainstalované Visual Studio.

### B.1 Kompilace programu

Pro kompilaci aplikace na platformě Windows stačí otevřít projekt Visual Studia pojmenovaný `CarParkANPR.sln`, který je na DVD umístěný v adresáři `Program`. V projektu je přednastavena konfigurace `Release` pro platformu `Win32`. Při spuštění či kompilaci projektu s touto konfigurací je v adresáři se souborem `CarParkANPR.sln` vytvořen adresář `Release`, ve kterém je vytvořen spustitelný soubor `CarParkANPR.exe` a jsou sem nakopírovány potřebné knihovny a soubory.

Pro kompilaci na platformě Linux je připraven soubor `CMakeLists.txt`, který je na DVD umístěný v adresáři `Program`. Nejprve je potřeba přejít do adresáře se souborem `CMakeLists.txt` a následně vygenerovat `Makefile` následujícím příkazem:

```
$ cmake .
```

Poté, co je `Makefile` vygenerován, stačí vyvolat příkaz `make`. Po vyvolání tohoto příkazu je projekt zkompileován a spustitelný soubor `CarParkANPR` spolu s potřebnými soubory je umístěn do adresáře `build`, který se vytvoří ve stejném adresáři jako se nachází soubor `CMakeLists.txt`.

### B.2 Použití programu

Pro ovládání programu je možné program spustit s následujícími argumenty:

**[-d | --debug]**

– zapne ladící režim, ve kterém jsou vypisovány některé informace o běhu programu

**[-h | --help]**

– zobrazí nápovědu programu a nelze ho kombinovat s jinými parametry

**[-z | --display-size]=CISLO**

- nastaví maximální velikost okna s výstupem. Výchozí je 600 pixelů.

**[-v | --video]=CESTA\_K\_SOUBORU**

- pro otevření videosouboru
- klávesou ESCAPE lze program ukončit
- klávesou MEZERNÍK lze video pozastavit. Libovolnou klávesou ho pak lze opět spustit

**[-r | --directory]=CESTA\_ADRESARE**

- otevře adresář s obrázky
- obrázky lze procházet klávesami R doleva a L doprava
- klávesou P je spuštěno zpracování snímku

**[-i | --image]=CESTA\_K\_SOUBORU**

- pro otevření jednoho obrázku, stejné ovládání jako předchozí parametr

**--add-whitelist=CISLO\_RZ**

- pro přidání registrační značky na seznam povolených značek
- lze kombinovat s parametrem **--name=JMENO**, což umožní přiřadit k záznamu jméno

**--add-blacklist=CISLO\_RZ**

- pro přidání registrační značky na seznam zakázaných značek
- lze kombinovat s parametrem **--name=JMENO**, což umožní přiřadit k záznamu jméno

**--remove-rz=CISLO\_RZ**

- odebere RZ ze seznamu zakázaných i povolených RZ