

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

RE-IDENTIFIKACE VOZIDLA POMOCÍ ROZPOZNÁNÍ JEHO REGISTRAČNÍ ZNAČKY

DIPLOMOVÁ PRÁCE

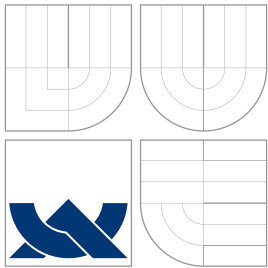
MASTER'S THESIS

AUTOR PRÁCE

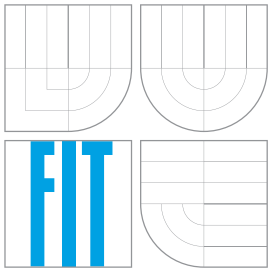
AUTHOR

Bc. JAKUB ŠPAŇHEL

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

RE-IDENTIFIKACE VOZIDLA POMOCÍ ROZPOZNÁNÍ JEHO REGISTRAČNÍ ZNAČKY

RE-IDENTIFICATION OF VEHICLES BY LICENSE PLATE RECOGNITION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAKUB ŠPAŇHEL

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. ADAM HEROUT, Ph.D.

BRNO 2015

Abstrakt

Tato práce se zabývá návrhem algoritmů detekce a rozpoznání registrační značky vozidla, které by byly použitelné pro re-identifikaci vozidla v obraze, a vytvořením jednoduchého systému analýzy dopravy. Byly navrženy a otestovány jednotlivé části toho systému. Konkrétně se jedná o detekci vozidel, detekci a rozpoznání registrační značky vozidla a systém samotný. Detekce vozidel je prováděna pomocí metody odečítání pozadí a vytváření ohraničení těchto oblastí s úspěšností $\approx 92\%$. Registrační značka je detekována pomocí kaskády klasifikátoru a dosahuje úspěšnosti 81.72% a přesnosti 94.42%. Rozpoznání poznávací značky pomocí metody *Template matching* dosahuje úspěšnosti 60.55%. Proto byl představen zcela nový princip rozpoznávání registrační značky vozidla pomocí jejího skenování užitím principu *sliding window* a rozpoznáváním neuronovou sítí. Neuronová síť dosahuje úspěšnosti 64.47% pro testovací datovou sadu při použití pěti příznaků. Malou úspěšnost neuronové sítě však ovlivňuje nedostatek vzorků některých znaků registrační značky.

Abstract

This thesis aims at proposing vehicle license plate detection and recognition algorithms, suitable for vehicle re-identification. Simple urban traffic analysis system is also proposed. Multiple stages of this system was developed and tested. Specifically – vehicle detection, license plate detection and recognition. Vehicle detection is based on background subtraction method, which results in an average hit rate of $\approx 92\%$. License plate detection is done by cascade classifiers and achieves an average hit rate of 81.92% and precision rate of 94.42%. License plate recognition based on Template matching results in an average precision rate of 60.55%. Therefore the new license plate recognition method based on license plate scanning using the sliding window principle and neural network recognition was introduced. Neural network achieves a precision rate of 64.47% for five input features. Low precision rate of neural network is caused by small amount of training sample for some specific license plate characters.

Klíčová slova

Počítačové vidění, doprava, analýza dopravy, detekce vozidel, detekce registrační značky, rozpoznání registrační značky, re-identifikace vozidel, odečítání pozadí, kaskáda klasifikátorů, template matching, sliding window, skenování registrační značky, neuronová síť

Keywords

Computer vision, Urban traffic, Urban traffic analysis, Vehicle detection, License plate detection, License plate recognition, Re-identification of vehicles, Background subtraction, Cascade classifier, Template matching, Sliding window, License plate scanning, Neural network

Citace

Jakub Špaňhel: Re-identifikace vozidla pomocí rozpoznání jeho registrační značky, diplomová práce, Brno, FIT VUT v Brně, 2015

Re-identifikace vozidla pomocí rozpoznání jeho registrační značky

Prohlášení

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

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jakub Špaňhel

27. května 2015

Poděkování

Chtěl bych poděkovat panu doc. Ing. Adamu Heroutovi, Ph.D. za vedení této práce, poskytnutí materiálů, rozšíření datové sady, jeho cenné rady, konstruktivní nápady a připomínky.

Dále pak Ing. Markétě Dubské, Ph.D. za vytvoření některých z poskytnutých materiálů.

Zvláštní poděkování potom patří moji rodině.

© Jakub Špaňhel, 2015.

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

Obsah

1 Úvod	2
2 Počítačové vidění v dopravě	3
2.1 Analýza dopravy z videa	3
2.2 Detekce vozidla v obraze	5
2.3 Detekce registrační značky vozidla	6
2.4 Rozpoznání registrační značky vozidla	7
3 Re-identifikace vozidla pomocí rozpoznání registrační značky	8
3.1 Systém analýzy dopravy	10
3.2 Detekce vozidla v obraze	14
3.3 Detekce registrační značky vozidla	17
3.4 Metody rozpoznání RZ	23
4 Implementace a vyhodnocení	32
4.1 Pořízená datová sada	32
4.2 Pomocné aplikace	33
4.3 Detekce vozidla ze statické kamery	36
4.4 Detekce registrační značky vozidla	37
4.5 Příprava a trénování neuronové sítě	40
4.6 Rozpoznání detekované registrační značky	43
4.7 Jednoduchý systém analýzy dopravy	45
5 Závěr	47
A Obsah CD	53
B Kontingenční tabulky experimentů testování neuronové sítě	54

Kapitola 1

Úvod

Doprava, slovo bez jehož významu se v dnešním světě neobejdeme. Silnice jsou plné vozidel, protože jich tisíce každodenně vyjíždí do ulic měst. Dopravní situace je v řadě těchto měst neúnosná, a to zejména v dopravní špičce, kdy řidiči stojí mnohdy i desítky minut v kolonách, aby se úspěšně dostali do svého cíle. Dopravní nehody, opravy silnic, uzavírky... To všechno jsou faktory ovlivňující dopravní situaci v těchto městech. Elektrické světelné signalizační zařízení, neboli lidově semafor, je nasazován již více než sto let, aby bránil totálnímu kolapsu dopravy ve městech. S rostoucím počtem vozidel, ale tento systém ve své aktuální podobě pomalu přestává stačit. Nastává doba pro inteligentní dopravní systém, který by dokázal automaticky vyhodnocovat aktuální hustotu dopravy a ovládat podle ní semaforey, měnit povolenou rychlost, a docílit tak lepší průjezdnosti v těchto kritických místech.

Tato práce se zabývá návrhem algoritmů detekce a rozpoznání registrační značky vozidla, použitelnými pro jeho re-identifikaci v obraze a vytvořením jednoduchého systému analýzy dopravy. Re-identifikace vozidel v obraze je důležitá, aby bylo umožněno například modelovat aktuální dopravní situaci, či vyhodnocovat trajektorie vozidel na silnicích, a tak ještě lépe porozumět dopravní situaci při analýze dopravy.

Hlavním tématem této práce je právě návrh a testování algoritmů detekce a rozpoznání registrační značky vozidla, které jsou součástí jednotlivých částí systému analýzy dopravy. O principech a metodách využívaných při počítačovém vidění v dopravě pojednává kapitola 2. Kapitola 3 obsahuje návrh systému re-identifikace vozidla pomocí rozpoznání jeho registrační značky. Popis implementace a dosažené výsledky potom nalezneme v kapitole 4.

Kapitola 2

Počítačové vidění v dopravě

V dnešní době se počítačové vidění v dopravě používá převážně na systémy analýzy dopravy, které nám poskytují nejen statistické údaje, ale jsou schopné poskytnout i informace o spáchání dopravního přestupku či detekovat dopravní nehody. Další využití nalézá počítačové vidění například u systému automatického otevírání závor na základě jejich registrační značky.

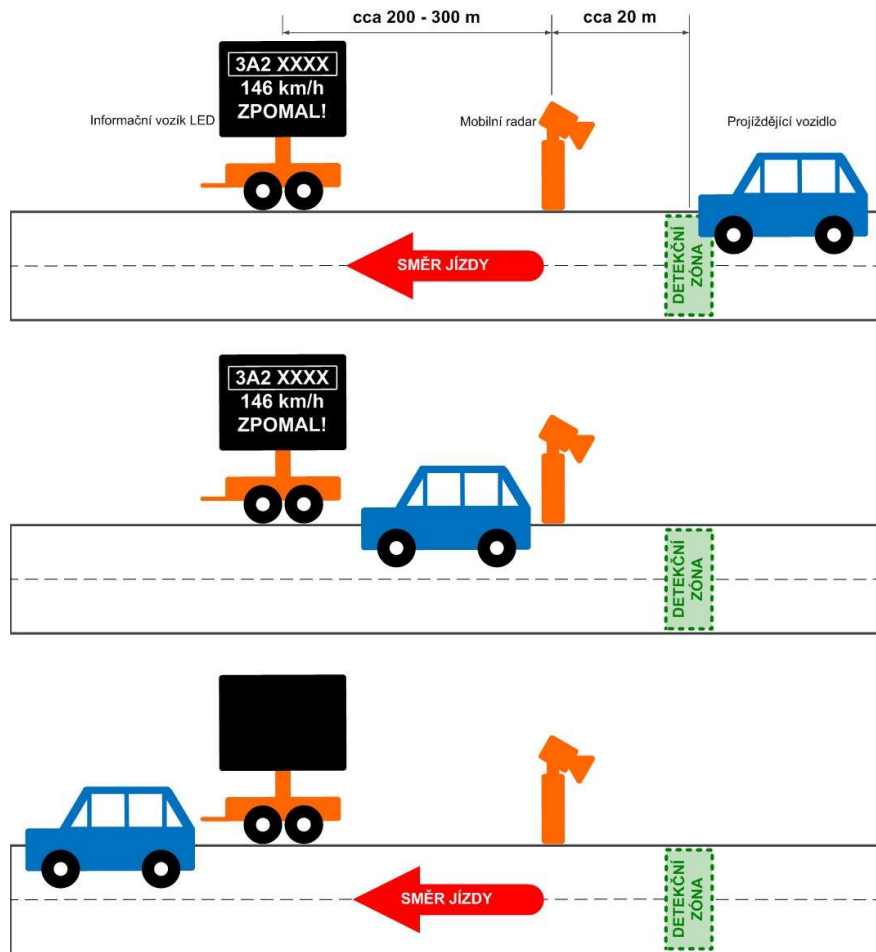
Hlavní téma této práce se zabývá analýzou dopravy. Proto se zaměříme hlavně na analýzu dopravy a procesy s ní související. Při analýze dopravy se obvykle systémy skládají z několika komponent, které na sebe navazují a vzájemně spolupracují. Typicky se jedná o komponenty zabezpečující detekci a případně i sledování pohybu vozidel, dále pak komponenty zajišťující detekci a eventuálně i rozpoznání jejich registračních značek.

Možnostmi analýzy dopravy se podrobněji zabývá sekce 2.1. Porovnáním metod detekce vozidel se zabývá sekce 2.2, detekcí registračních značek vozidla sekce 2.3 a samotným rozpoznáním registrační značky vozidla potom sekce 2.4.

2.1 Analýza dopravy z videa

Dnešní systémy analýzy dopravy jsou schopny více či méně přesně odhadnout rozměry vozidla, jeho rychlost, spočítat počet projíždějících vozidel, ale i detekovat dopravní nehody či přestupky spojené se dopravní signalizací. Ve většině případů se tedy jedná o statistické údaje, ale k dispozici jsou i údaje informativního charakteru.

To, že jde vývoj těchto systémů kupředu, potvrzuje i vytvoření a nasazení radarového systému MOTES, zahrnující detekci a rozpoznání registrační značky vozidla, který si nechalo firmou Eltodo vytvořit Ředitelství silnic a dálnic České republiky [26]. Schéma toho systému je k dispozici na obrázku 2.1.



Obrázek 2.1: Schéma znázorňující funkčnost radarového systému MOTES vyvinutého společností Eltodo. Schéma bylo převzato ze stránek Ředitelství silnic a dálnic ČR [26].

Moderní způsob analýzy dopravy

Moderní způsob analýzy dopravy z videa můžeme například najít ve článku pocházejícím z naší fakulty [5]. Tento článek se zabývá automatickou kalibrací kamery pro analýzu dopravy, a je zde uveden způsob, jakým lze kalibrovat kameru pomocí zjištění směru pohybu vozidel na silnici. Tato metoda využívá počítání optického toku pro kalibraci kamery a určení obalového kvádrů vozidla. Díky přesnému určení obalového kvádrů vozidla jsou autoři schopni odhadnout rozměry vozidla či jeho aktuální rychlost, nebo například určit teplotní mapu pohybu vozidel na silnici, a to všechno s minimální odchylkou oproti reálným hodnotám. Základní principy užití v tomto systému jsou detailněji popsány v sekci 3.1.

2.2 Detekce vozidla v obraze

V současné době se využívá několik principů v oblasti detekce vozidel. Tyto principy můžeme rozdělit do několika kategorií. Sekce 2.2.1 popisuje metody založené na znalosti tvaru vozidla. Metody založené na pravděpodobnosti výskytu vozidla jsou poté popsány v sekci 2.2.2. Z našeho pohledu nejzajímavější jsou metody využívající statickou kameru, zmíněné v sekci 2.2.3. Tomuto tématu jsem se již věnoval ve své bakalářské práci [25], ve které jsem popisoval i tyto metody.

2.2.1 Metody založené na tvaru vozidla

Metody detekce automobilů, založené na tvaru automobilu, obvykle trpí nižší mírou úspěšnosti kvůli rozmanitosti tvaru vozidel a pozadí. Jako zástupce této kategorie lze uvést metody ze článků [2] a [14]. Například metoda uvedená ve článku [2], zabývající se detekcí vozidel v reálném čase, je založena na hypotéze přítomnosti vozidel v obdélníkových podoblastech a použití robustní klasifikace vektorů klíčových bodů. Jeden vektor je vyextrahován z každé oblasti, která by mohla obsahovat vozidla. Je připravena vzorová množina vektorů tříd vozidel a objektů neodpovídajících vozidlům. Hypotéza je následně ověřena výpočtem Mahalanobisových vzdáleností mezi vektory.

2.2.2 Metody založené na výskytu a vzhledu vozidla

Metody uvedené v této sekci využívají při svém vytváření pravděpodobnosti. Tyto pravděpodobnosti se týkají zejména výskytu a vzhledu vozidel, případně rozložení jejich intenzit [13]. Například metoda představená ve článku [27], je schopná detekovat lidské obličejy a osobní automobily. Tato metoda detekce 3D objektů používá histogramy vzhledu objektů, které chce detekovat, i jejich protikladů. Každý histogram reprezentuje průměrnou statistiku podmnožiny vlnových koeficientů a její pozice na objektu. Metoda je založena na použití velkého množství histogramů reprezentujících široké spektrum vizuálních atributů, a je schopná detekovat osobní automobily z různých úhlů pohledu.

2.2.3 Metody využívající statickou kameru

Další skupina metod detekce automobilů potom využívá pohled ze statické kamery. Například metoda představená ve článku [34] se zabývá přesnou segmentací jedoucích vozidel. Segmentace je založena na Gaussovském modelu pohybu pro jedoucí vozidla v dynamických scénách. Zkoumáním rozdílu mezi vektory pohybu dynamického pozadí a vektory pohybu jedoucích vozidel lze zjistit, že pohybové vektory jedoucích vozidel se shlukují do menší oblasti, zatímco pohybové vektory dynamického pozadí se rozptylují.

Jinou alternativu k segmentaci jedoucích vozidel představuje článek [15]. Vzhledem k tomu, že stíny pohybujících se objektů často narušují sledování, tato metoda navrhuje

segmentaci pomocí skrytého Markovova modelu (HMM¹), který v reálném čase klasifikuje každý pixel nebo region do tří kategorií: stíny, popředí a pozadí. Model provádí odhad stavu v reálném čase, čehož se využívá při segmentaci. Tato metoda byla testována na dopravních kamerách sledujících provoz, tudíž z pevného bodu.

Další možností rychlé detekce automobilů se věnuje článek [36]. Tato technika, která je tentokrát založena na přítomnosti více náznaků, využívá nehybného obrazu. Detekce se skládá ze dvou úrovní, kdy první je založená na získání náznaků hran a dalších bodů zájmu, a druhá se zabývá modelováním a trénováním Gaborových momentů.

2.3 Detekce registrační značky vozidla

Tato sekce obsahuje srovnání a seznámení se s principy a metodami využívanými při detekci registrační značky vozidla, dále jen RZ vozidla. Obecně můžeme metody detekce RZ rozdělit na dvě kategorie, metody založené na statické analýze snímku, tedy získávání informací o barvě, pozici, hranách, konturách atd., a dále metody založené na strojovém učení. Některé metody navrhují kromě detekce také způsob rozpoznání znaků RZ, jako například ve článku [30]. První kategorií metod detekce RZ se zabývá sekce 2.3.1, druhou kategorií se potom zabývá sekce 2.3.2.

2.3.1 Metody detekce založené na statické analýze obrazu

Metoda představená ve článku [31] detekuje RZ vozidla na základě barvy a textury. V prvním kroku detekce je využito matematické morfologie a znalostí o barvě RZ pro odhadnutí přibližné lokace. Přesná lokace je potom určena na základě příznaků z okolní textury.

Článek [30] naopak navrhuje robustní metodu detekce a rozpoznání registrační značky založenou na analýze kontur získaných ze snímku. Systém je složen z fáze detekce RZ a fáze jejího rozpoznání. Detekce je prováděna získáním kandidátních oblastí výskytu RZ. Přesná poloha je potom určena pomocí techniky detekce textu založené na hranách.

2.3.2 Metody detekce RZ založené na strojovém učení

Metoda detekce RZ v reálném čase, představená ve článku [29], je založena na dvoustupňové adaptivní metodě AdaBoost. Metoda AdaBoost použitá v tomto algoritmu pracuje s Haarovými příznaky pro výpočet a výběr klíčových bodů ze snímku RZ. AdaBoost algoritmus je použit pro klasifikaci částí obrazu v rámci prohledávacího okna, které je výsledkem klasifikace silného klasifikátoru. Toto okno je poté klasifikováno jako registrační značka či okolí. Experimentální výsledky této metody vykazují úspěšnost detekce 98.38% při průměrné době zpracování ≈ 49 ms.

Na použití algoritmu AdaBoost je také založena metoda představená ve článku [32]. Oproti předchozí metodě ovšem nevyužívá Haarových příznaků, ale gradientního obrazu

¹Hidden Markov Model

a získá tak hrubý odhad polohy RZ. Verifikace RZ potom probíhá pomocí heuristiky rozhodovací strategie založené na hlasování.

2.4 Rozpoznání registrační značky vozidla

V této části se budeme věnovat rozpoznání registračních značek vozidla, dále jen RZ vozidla. Seznámíme se s aktuálními principy a metodami využívanými v této oblasti. V dnešní době se používá mnoho různých přístupů v oblasti rozpoznání registrační značky. Mezi ty základní můžeme řadit klasické OCR (*Optical Character Recognition*), používané na rozpoznání tištěného textu. Zde se budeme zabývat pokročilejšími metodami rozpoznání RZ vozidla, které jsou založené na klasifikaci.

V sekci 2.4.1 jsou představené metody, které využívají principu neuronových sítí či jejich kombinace s dalšími technikami, v sekci 2.4.2 jsou potom metody založené na principech klasifikace.

2.4.1 Metody rozpoznání založené na neuronových sítích

Metoda automatického rozpoznání registrační značky vozidla představená ve článku [17] využívá 2-D Haarovu diskretní vlnkovou transformaci (DWT) a umělých neuronových sítí. 2-D Haarova DWT je využita hned dvakrát pro získání pouze nízkofrekvenční části obrazu. Rozpoznání potom probíhá pomocí skenovacího okna přes detekovanou RZ. Experimentální výsledky této metody vykazují úspěšnost rozpoznání 95.33%. Hybridní technika rozpoznání RZ pomocí výběru klíčových bodů, opět pomocí vlnkové transformace, a následně klasifikace pomocí umělé neuronové sítě byla potom představena ve článku [28].

Článek [33] představuje využití PCNN (*Pulse Coupled Neural Network*) a techniky *Template Matching* pro rozpoznání RZ vozidla. První PCNN získává lokaci registrační značky vozidla v šedotónovém obrazu, druhá PCNN je poté použita k extrakci jednotlivých znaků nalezené RZ. Po extrakci jednotlivých znaků probíhá rozpoznání pomocí techniky *Template Matching* a RZ je rozpoznána na základě podobnosti. Uvedená metoda má oproti ostatním metodám lepší výsledky při nízkém osvětlení scény či horší kvalitě obrazu.

2.4.2 Rozpoznání RZ pomocí klasifikace

Metoda rozpoznání RZ založená na hierarchické klasifikaci byla představena ve článku [10]. První fáze rozpoznání této metody spočívá v nalezení optimálních vlnkových oblastí při jejich dekompozici a K-L transformaci. Následně dojde k rozpoznání RZ pomocí extrakce klíčových bodů a hierarchické klasifikace.

Článek [35] představuje algoritmus rozpoznání RZ vozidla založený na použití *Support Vector Machine*. Systém rozpoznání představený v této práci je složen ze tří částí – předzpracování RZ a její lokalizace, segmentování jednotlivých znaků RZ a nakonec jejich rozpoznání.

Kapitola 3

Re-identifikace vozidla pomocí rozpoznání registrační značky

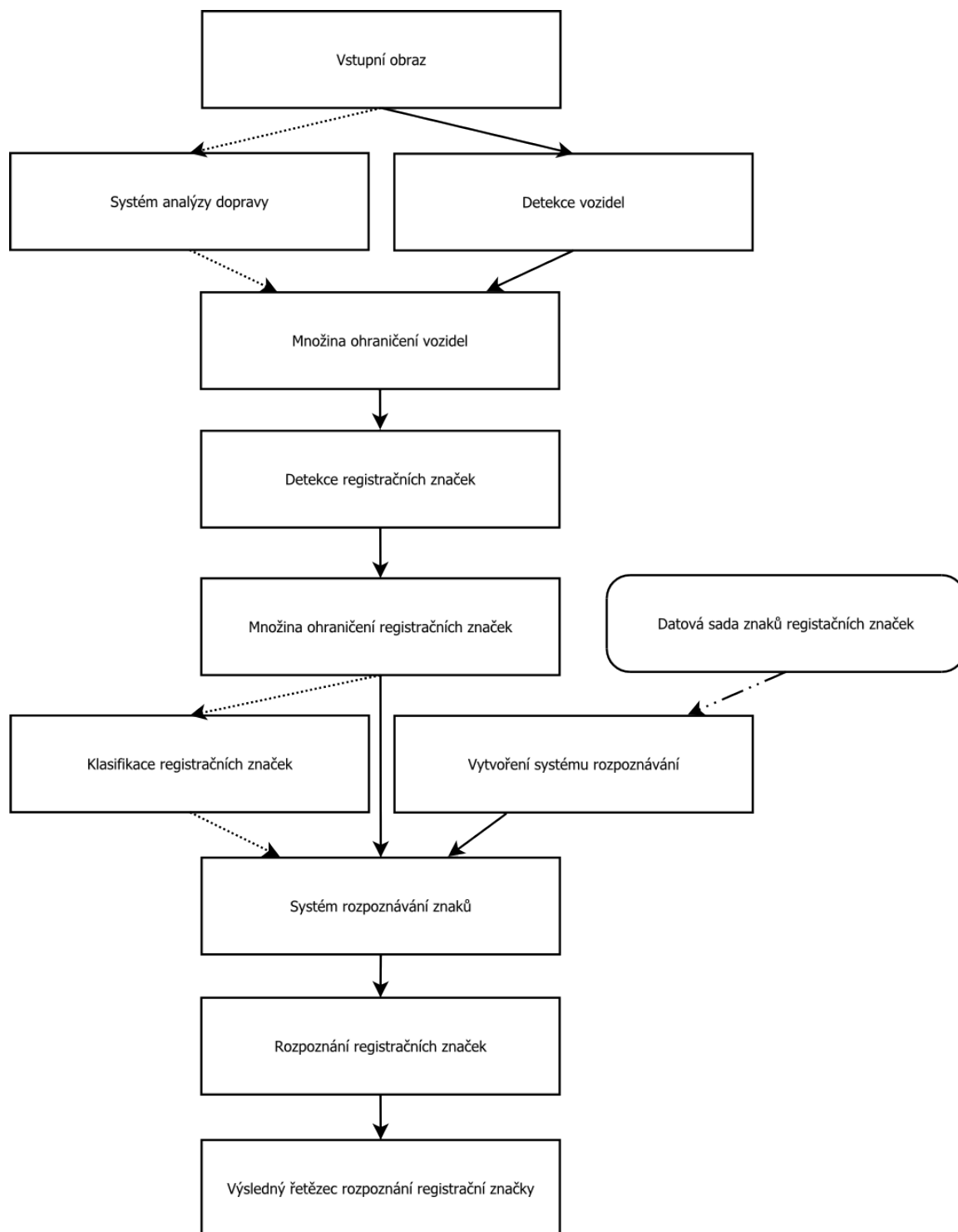
Navrhl jsem jednotlivé fáze zpracování vstupních dat pro systém re-identifikace vozidla. Pro každou fázi systému jsem implementoval více použitelných metod a prováděl s nimi experimenty. Jednotlivé části systému jsou demonstrovány diagramem na obrázku [3.1](#)

Jak vyplývá z diagramu [3.1](#), první fází je detekce vozidel, která je popsána v sekci [3.2](#). Alternativně lze použít systém analýzy dopravy pro získání ohraničení vozidel. Detekce vozidel je v aplikaci důležitá z důvodu real-time zpracování obrazu, aby nedocházelo k prodávám a zbytečnému prohledávání celého vstupního obrazu během detekce registrační značky vozidla. Výstupem této fáze je množina ohraničení vozidel.

Druhou fází, kterou popisuje sekce [3.3](#), je detekce registrační značky vozidla. Algoritmus detekce je navržen takovým způsobem, aby byl schopný detekovat registrační značky v celém obraze. Pro lepší výsledky, časové i funkční, je ovšem vhodnější detekci provádět až na detekovaných vozidlech. Po dokončení detekce registračních značek by mělo dojít ke klasifikaci, zda se jedná o registrační značku či nikoliv. Vhodným klasifikátorem se zde jeví Support Vector Machine (SVM), který je nutno nejprve řádně natrénovat dostatečnou datovou sadou a jeho účinnost ověřit principem „cross-validace“. Jak lze využít SVM pro klasifikaci registračních značek popisuje článek [\[35\]](#).

Při opětovném pohledu do diagramu na obrázku [3.1](#) je patrné, že další logickou fází je rozpoznání detekovaných registračních značek vozidla popsané v sekci [3.4](#). K této fázi se také váže vytvoření systému rozpoznávání registrační značky vozidla. Bližší popis vytváření datové sady znaků a vlastní trénování systému rozpoznávání je uveden v kapitole [4](#), konkrétně v sekci [4.5](#).

Tento systém by měl navazovat na systém analýzy dopravy představený ve článku [\[5\]](#), který je blíže popsán v sekci [3.1](#). Se znalostí obecného čtyřúhelníku ohraničujícího přední, resp. zadní část vozidla, lze rovnou přejít k detekci registrační značky. Různými možnostmi detekce registrační značky vozidla se zabývá sekce [3.3](#). Metodami vhodnými pro rozpoznání registrační značky vozidla se potom zabývá sekce [3.4](#).



Obrázek 3.1: Diagram návrhu aplikace

3.1 Systém analýzy dopravy

Autoři článku [5] představují systém automatické kalibrace kamery pro analýzu dopravy. Tento systém dokáže automaticky zkalibrovat kameru bez potřeby uživatelského vstupu a zvládá detekovat vozidla téměř z jakéhokoliv úhlu. Výhodou tohoto systému je možnost určení přesného ohraničení vozidla pomocí obalového kvádru, který vzniká během kalibrace kamery na danou scénu. Klíčovým bodem tohoto systému je detekce úběžníků, které pomáhají určovat orientaci kamery s ohledem na tok dopravy. [5].

3.1.1 Automatická kalibrace kamery

Kalibrační metoda použitá v tomto systému byla představena ve článku [4]. Metoda detekuje dva hlavní ortogonální směry – první ve směru dopravy a druhý, který je kolmý ke směru prvnímu a zároveň rovnoběžný s cestou. S předpokladem, že hlavní bod kamery je ve středu projektivní roviny, je vypočítán třetí ortogonální směr a ohnisková vzdálenost kamery. Detekce používá Houghovu transformaci založenou na paralelních souřadnicích, které mapují celou 2D projektivní rovinu do konečného prostoru, označovaného jako *diamond space*, pomocí po částech lineárního mapování přímek. [5]

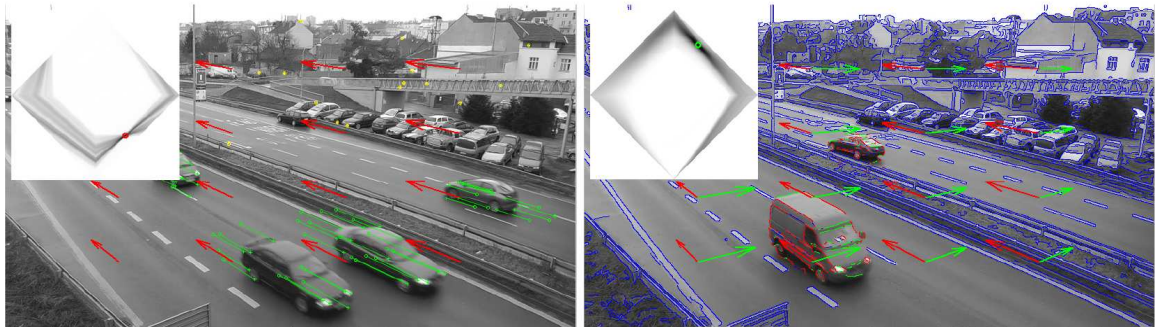
Pro detekci prvního úběžníku se následně detekují klíčové body a jejich pohyb v dalších snímcích je sledován pomocí KLT trackeru. Body, které mají mezi snímky znatelný pohyb, jsou považovány jako lineární fragmenty trajektorie vozidel a dostávají jistou váhu (hlasy) v akumulátoru prostoru *diamond space*. Bod s největší vahou (nejvíce hlasy), je potom považován jako první úběžník. [5]

Druhý úběžník odpovídá směru souběžnému k vozovce a je kolmý na první úběžník. Pro jeho detekci se opět využívá *diamond space* prostor. Mnoho hran na vozidle se může shodovat s druhým úběžníkem, a mohou tedy být také zařazeny do akumulátoru. Uvažují se ovšem jenom hrany na pohybujících se objektech – předpokládaných vozidlech. Vyloučeny z akumulátoru jsou hrany, které odpovídají více prvnímu úběžníku, a také hrany s přibližně vertikální orientací. Druhou podmínku lze porušit, je-li první úběžník detekován jako blížící se nekonečnu. Na obrázku 3.2 vpravo lze vidět hranový model pozadí, vyloučené a akumulované hrany společně s *diamond space* prostorem. [5]

3.1.2 Konstrukce obalového kvádru vozidla

Dalším krokem je konstrukce obalového kvádru pozorovaných vozidel. Autoři článku předpokládají, že siluety vozidel mohou být extrahovány pomocí vytvoření modelu pozadí a následně detekcí popředí. Tato část odpovídá detekci vozidla metodou odečítání pozadí, která je popsána podrobněji v sekci 3.2.1. Detekce částí popředí může být udělána spolehlivě, včetně odstranění stínů vozidel. Dále autoři předpokládají, že se vozidla pohybují ve směru k/od prvního úběžníku. [5]

Přístup autorů je založen na pozorování, že v oblasti vozidel se vyskytují stabilně hrany, které lze považovat za spolehlivé. Jak je vidět na obrázku 3.3, detekovaná oblast vozidla je



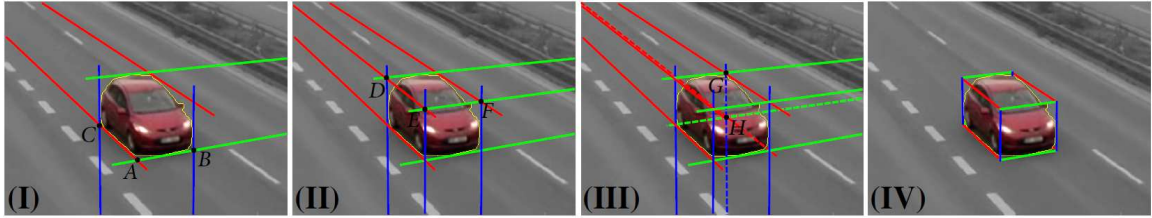
Obrázek 3.2: Detekce prvního a druhého úběžníku. (Vlevo) Sledované body jsou použity pro určení prvního úběžníku. Body označené zeleně znázorňují výrazný pohyb a jsou akumulovány. Body označené žlutě jsou stabilní a nemají žádnou váhu. Akumulovaný prostor *diamond space* je zobrazen v levém horním rohu. (Vpravo) Akumulace druhého úběžníku. Modré hrany náleží pozadí. Červené hrany jsou vyloučeny z akumulování kvůli jejich vertikálnímu směru nebo směru k prvnímu úběžníku. Zelené hrany jsou akumulovány do prostoru *diamond space*, který je zobrazen v levém horním rohu. Jeho maximum je znázorněno pomocí zelené kružnice. Převzato ze článku [5].

pro ilustraci ponechána barevně, zatímco zbytek obrázku je de-saturován. V tomto případě prochází červené přímky prvním úběžníkem a jsou tečnami oblasti výskytu vozidla. Zelené přímky jsou poté tečny oblasti výskytu vozidla, které prochází druhým úběžníkem. Modré tečny potom prochází třetím úběžníkem. [5]

Jelikož nejsou oblasti výskytu vozidel úplně přesné a vozidlo není přesný kvádr, je vytváření obalového kvádru vozidla neurčitě, resp. záleží na pořadí vybírání tečen a jejich průsečíků. Autoři navrhují následující pořadí, které se jeví jako nejvíce stabilní: Prvně je vytvořen bod A , který je průsečíkem spodní červené a zelené tečny. Následně body B a C jsou definovány jako průsečíky spodní zelené tečny s pravou modrou a spodní červené s levou modrou, v tomto pořadí. Tento proces je znázorněn v první části obrázku 3.3. Sestrojené úsečky AB a AC definují kratší a delší stranu základny kvádrů. Bod D leží na průsečíku horní zelené tečny a levé modré tečny. Společně s přímkou procházející bodem A a třetím úběžníkem, jednoznačně definuje bod E , jak je ukázáno na druhé části obrázku 3.3. Bod E může být také určen pomocí bodu F – což vede k alternativní pozici bodu E . Jako kandidátní je vybrán bod E , který má větší vzdálenost $|AE|$, což zaručuje, že celá oblast výskytu vozidla bude uzavřena v obalovém kvádru. Se znalostí bodů F a D , je bod G určen jako průsečík přímky procházející bodem D a druhým úběžníkem, a přímky procházející bodem F a prvním úběžníkem, jak ukazuje třetí část obrázku 3.3. [5]

Pokud je konfigurace úběžníků, s přihlédnutím k centru oblasti výskytu vozidla, jiná než bylo popsáno v předchozích odstavcích, může se nastavení a pořadí použitých tečen a bodů nepatrně změnit. Jelikož bývají střecha a boky vozidla nějakým způsobem prohnuty, může být detekovaný obalový kvádr vozidla trochu menší, než by byl ve skutečnosti. Experimenty autorů ukázali, že výsledná přesnost není ovlivněna nepatrně zmenšeným obalovým kvádrem vozidla. Pro přesný odhad rozměrů vozidla a stanovení přesného obalového kvádrů,

musí být odstraněny stíny z detekovaných oblastí popředí – tedy oblastí výskytu vozidel. Autoři předpokládají pouze výskyt měkkých stínů a k jejich odstranění používají metodu prezentovanou ve článku [12] autorů Horprasert et al. [5]



Obrázek 3.3: Konstrukce obalového kvádra vozidla. (I) Tečny a jejich relevantní průsečíky A, B, C . (II) Odvozené přímky a jejich průsečíky E, D, F . (III) Odvozené přímky a jejich průsečíky G, H . (IV) Hotový obalový kvádr vozidla. Převzato ze článku [5].

3.1.3 Detekce jízdních pruhů

Znalost obalového kvádra vozidla může být také nápomocna k přesné segmentaci dopravních pruhů, dokonce i ze záběrů, kde se vozidla jedoucí v jednotlivých jízdních pruzích vzájemně překrývají. Pro každou trajektorii vozidla se akumulují vyplněné pásy určené čtyřmi vrcholy $A_i, B_i, A_{i+1}, B_{i+1}$, kde i označuje body v i -tém snímku videa. Po nashromáždění pásů jsou nalezena minima na přímkce kolmé ke směru vozovky, tj. na přímkce procházející druhým úběžníkem, a tato minima jsou určena jako ohraničení jízdních pruhů. Přesná segmentace jízdních pruhů je zobrazena na obrázku 3.4. [5]



Obrázek 3.4: Ukázka přesné segmentace metody detekce jízdních pruhů z automatického systému kalibrace kamery pro analýzu dopravy. Obrázek byl převzat ze článku [5]

3.1.4 Transformace vstupního obrazu

Kamera snímající vozovku nemusí být vždy umístěná přímo nad jízdními pruhy, tak jak je tomu například na českých dálnicích. Jelikož není budování těchto bran levnou záležitostí, předpokládejme, že bude kamera snímající vozovku umístěna například na sloupu veřejného

osvětlení, tedy pod značným úhlem oproti kolmému pohledu na vozovku. Z tohoto úhlu bude přední, resp. zadní, část obalového kvádrů vozidla reprezentovat obecný čtyřúhelník, který je perspektivně zkreslený, namísto pravoúhelníku, kolmému k pohledu kamery.

Znalost přesného obalového kvádrů vozidla nám potom umožňuje provádět transformace vstupního obrazu a transformovat tak perspektivní zkreslení vozidla. Abychom mohli provést transformaci obrazu, je potřeba znát čtyři vrcholy A, B, C, D obalového kvádrů, reprezentující přední, resp. zadní, část vozidla. Pro zachování správného poměru stran transformovaného snímku je dále potřeba znalost výšky a šířky vozidla. V následujícím kroku je nutné vypočítat transformaci z bodů $[A; B; C; D]$ do pravoúhelníku o souřadnicích $[[0, 0]; [W, 0]; [W, H]; [0, H]]$, kde W je šířka vozidla a H je jeho výška. Po provedení této transformace dojde ke srovnání registrační značky vozidla, která bude lépe detekovatelná.

Transformace a extrakce přední části vozidla

Srovnání vstupního obrazu a extrakci přední části vozidla lze provést například s použitím nástroje *Matlab* v několika krocích.

```
% Extrakce pravoúhelníků předních částí vozidel
```

```
% Vrcholy reprezentující přední část vozidla
```

```
A = Tracks(i).box(1, :);
```

```
B = Tracks(i).box(2, :);
```

```
C = Tracks(i).box(3, :);
```

```
D = Tracks(i).box(4, :);
```

```
% Vypočítání výšky a šířky vozidla
```

```
W = Tracks(i).size(1); % šířka
```

```
H = Tracks(i).size(2); % výška
```

```
% Výpočet transformační matice z bodů [A, B, C, D]
```

```
% do pravoúhelníku [0 0; W 0; W H; 0 H]
```

```
transform_image = fitgeotrans([A; B; C; D],  
[0 0; W 0; W H; 0 H], 'projective');
```

```
% Scale factor
```

```
scale = min(100./[W,H]);
```

```
% Převod I do I_t použitím transformace transform_image
```

```
% Extrakce přední části automobilu - [0, W], [0, H]
```

```
I_t = imwarp(I, transform_image, 'bicubic', 'OutputView',  
imref2d(round([H,W].*scale), [0,W], [0,H]));
```

3.2 Detekce vozidla v obraze

V této části práce si blíže představíme a přiblížíme některé metody, které jsou vhodné pro detekci vozidla ze statické kamery. Sekce 3.2.1 se zabývá metodou odečítání pozadí, která je využita pro získání oblastí výskytu vozidel v systému analýzy dopravy popisovaném v sekci 3.1. Fáze detekce vozidel je využívána pouze k urychlení dalších procesů, z toho důvodu nemusí být určení ohraničení vozidla stoprocentně přesné. Detekce samotná však musí splňovat kritérium nulového počtu *false negative* i za cenu více *false positive*, jelikož všechna vozidla v obraze musí být detekována.

Detekce vozidla je v návrhu systému uvedena jenom pro úplnost. Při nasazení systému do reálného provozu by měla být tato fáze přeskočena, jelikož vstupem systému mají být části obalového kvádrů vozidla směřující ke kameře, které jsou součástí výstupu systému analýzy dopravy popsaném v článku [5], jehož základní principy byly přiblíženy v sekci 3.1.

3.2.1 Metoda odečítání pozadí

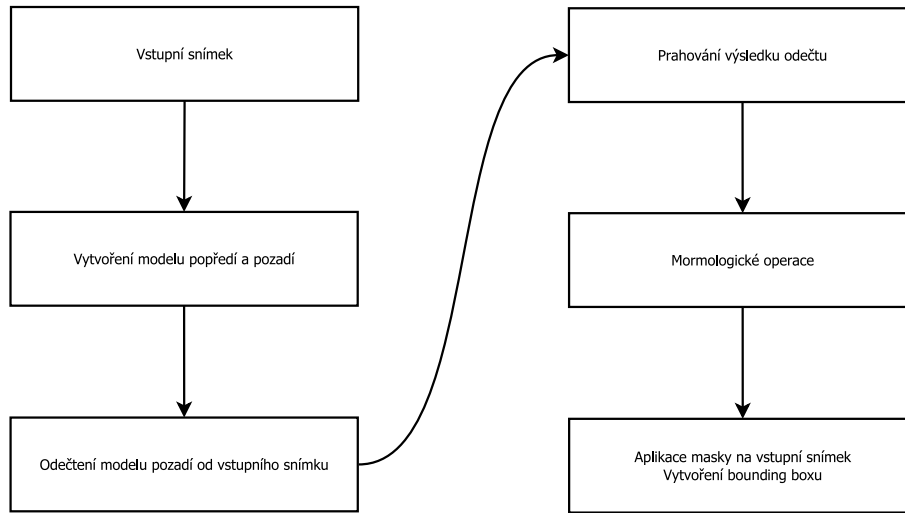
Jak již název napovídá, jednu z metod detekce vozidel lze vytvořit pomocí metody odečítání pozadí. Modely pozadí i popředí pro segmentaci vstupního obrazu podle barvy jsou založeny na směsici Gaussovských funkcí. Jednorozměrná vystředěná Gaussova funkce je uvedena v rovnici (3.1), kde x je proměnná, μ je střední hodnota a σ je směrodatná odchylka. Pro vytváření modelu pozadí si ale nevystačíme s jednorozměrnou Gaussovou funkcí, ale potřebujeme funkci vícerozměrnou. Vícerozměrná Gaussova funkce je uvedena v rovnici (3.2), kde d je počet proměnných vícerozměrného vstupního vektoru \vec{x} , $\vec{\mu}$ je vektor středních hodnot vstupních proměnných a Σ je kovarianční matice velikosti $d \times d$. Podrobněji se tímto tématem zabývá kniha [18].

V praxi to znamená, že je ze sekvence snímků vytvořen model pozadí a model popředí. Model pozadí se aktualizuje vždy z několika po sobě jdoucích snímků a v každém kroku je následně odečten od vstupního snímku a vyprahován, čímž vznikne maska popředí. Pro zkvalitnění výsledků segmentace jsou na masku popředí použity morfologické operace (eroze, dilatace, morfologické otevření/uzavření).

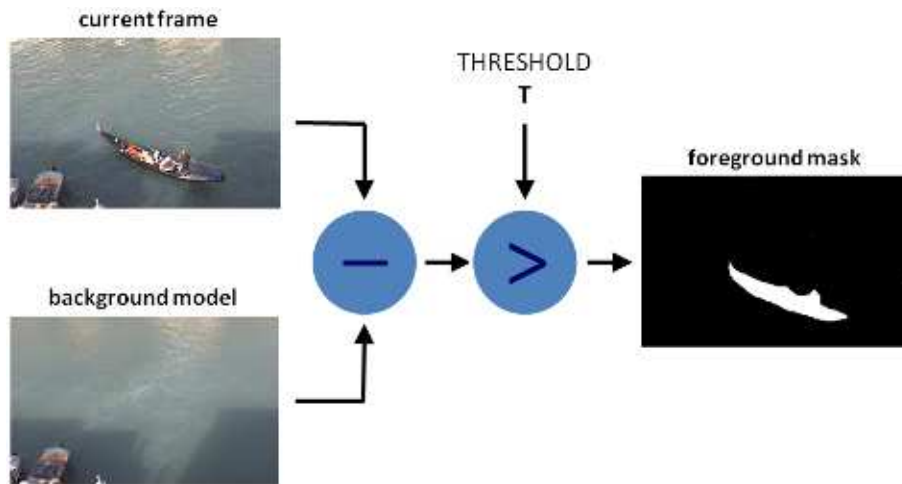
Nakonec je pro jednotlivé nalezené oblasti vypočítáno ohraničení, neboli bounding box. Výstupní oblast výskytu vozidla nemusí být celistvá ani po opakovaném aplikování morfologických operací, jak je zobrazeno na obrázku 3.7, a může tedy dojít k vytvoření většího počtu ohraničení pro jednu detekovanou oblast. Z toho důvodu je vhodné provést post-processing získaných ohraničení a redukovat tak počet ohraničení na nezbytné minimum. Ukázka možného výskytu ohraničení pro jednu detekovanou oblast výskytu vozidla a redukce počtu těchto ohraničení je zobrazena na obrázku 3.8. Celý proces zobrazuje diagram na obrázku 3.5. Princip odečítání pozadí je poté ukázán na obrázku 3.6.

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(x - \mu)^2}{2\sigma^2} \quad (3.1)$$

$$G(\vec{x}, \vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \frac{-(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})}{2\sigma^2} \quad (3.2)$$



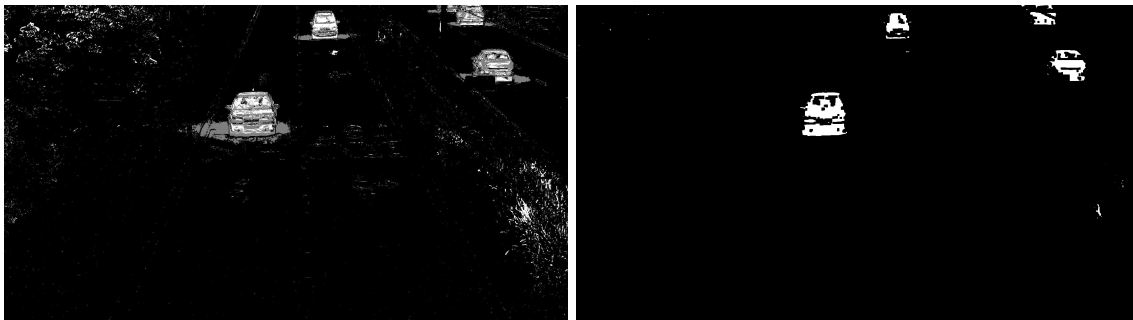
Obrázek 3.5: Diagram principu detekce vozidla.



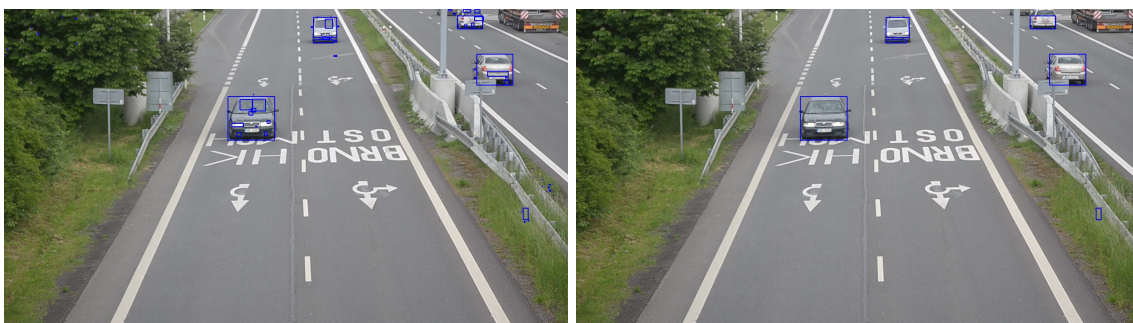
Obrázek 3.6: Ukázka principu metody odečítání pozadí. Obrázek byl převzat z [22]

3.2.2 Metoda rozdílů snímků

Dalším možným principem detekce vozidla ze statické kamery je použití metody rozdílů snímků. Tato metoda je velice podobná metodě odečítání pozadí, nepotřebuje však vytvářet



Obrázek 3.7: Ukázka výsledku odečtení pozadí bez morfologických operací (vlevo) a s nimi (vpravo).



Obrázek 3.8: Vizualizace výpočtu ohraničení detekovaných oblastí. (Vlevo) Před redukcí přebytečných ohraničení. (Vpravo) Ohraničení detekovaných oblastí po aplikaci redukce.

modely popředí ani pozadí pomocí Gaussovských funkcí, je proto rychlejší, i když může být méně přesná. Princip této metody spočívá v provedení rozdílu dvou snímků. Mějme snímek A , který bude reprezentovat aktuální snímek z videosekvence. Dále mějme snímek B , což je statický snímek čistého pozadí. Výsledný snímek C je pak dán rozdílem předchozích dvou snímků, přesněji $C = A - B$. Snímek C reprezentuje pohybující se objekt za předpokladu jeho výskytu v obraze. Celý tento proces je ukázán na obrázku 3.9.



Obrázek 3.9: Ukázka metody rozdílů snímků. (Vlevo) Zpracovávaný snímek videa. (Uprostřed) Statický snímek pozadí. (Vpravo) Výsledný snímek po odečtení. Upraveno pro lepší viditelnost segmentovaných objektů.

3.3 Detekce registrační značky vozidla

Detekce registrační značky vozidla, dále jen RZ vozidla, je stěžejní částí toho systému. Během návrhu fáze detekce RZ vozidel jsem vyzkoušel různé techniky detekce RZ.

Standardní registrační značka České republiky a většiny států Evropy má šířku 520 mm a výšku 110 mm, existují však i další atypické rozměry registračních značek. Ilustrace registračních značek České republiky je vyobrazena na obrázku 3.10.

Máme-li k dispozici vnější rozměry i vnitřní dispozice registrační značky, můžeme snadno vypočítat poměr stran RZ i poměr výšky a šířky jejích znaků. Zmíněné rozměry a poměry stran jsou zobrazeny na obrázku 3.11. Znalosti vnějších rozměrů a můžeme využít nejen při trénování a detekci registrační značky vozidla pomocí kaskády klasifikátorů, ale i u dalších technik detekování registrační značky pomocí geometrických tvarů.

vzor	Párové značky /v cm, číselné označení/					Nepárové značky /v cm, číselné označení/					
	Osobní automobily		Nákladní	Americký formát		Připojná vozidla	-	Mota	Mopedy		
	2x 52x11	2x 28x20	52x11, 28x20	52x11, 34x20	2x 32x16	52x11, 32x16	52x11	34x20	28x20	20x16	8x11
	101	105	106	103	115	116	102	104	-	118	119**
2004											
2001											
Osobní 94											
Nákladní 94											
Cizinci 94											
Půjčovny 94											

Obrázek 3.10: Ukázka formátu a vzorů registračních značek vyskytujících se v České republice. Obrázek byl převzat z [11].

Při návrhu a implementaci této části systému jsem se nejdříve zaměřil na detekci RZ založenou na výskytu klíčových bodů v obraze. Následně jsem implementoval detekci RZ, založenou na kaskádě klasifikátorů, schopnou detekovat RZ vozidla bez znalosti přesného ohraničení vozidla a dodatečných transformací obrazu. V poslední fázi jsem vytvořil návrh

detekce RZ vozidla s přihlédnutím ke znalosti obecného čtyřúhelníku ohraničujícího přední, resp. zadní část vozidla.

Pro urychlení fáze detekce registrační značky vozidla a kvůli neznalosti pozice obecného čtyřúhelníku ohraničující přední část vozidla, navazuje tato část na samostatnou detekci vozidla pomocí metody odečítání pozadí, která byla popsána v sekci 3.3. Předpokladem níže popsaných technik je tedy pravoúhelníkem ohraničená oblast předpokládaného výskytu vozidla ve vstupním obraze.



Obrázek 3.11: Vizualizace rozměrů a poměrů stran standardní registrační značky České republiky. Obrázek byl převzat z [6] a dále upraven.

3.3.1 Detekce RZ založená na výskytu klíčových bodů

Jedním ze základních principů je použití základních klíčových pro lokalizaci registrační značky vozidla, mezi které patří například detekce rohů či hran. K lokalizaci registrační značky potom můžeme použít základní principy zpracování obrazu jako je prahování, zjištění kontur obrazu, aplikace morfologických operací či segmentace obrazu pomocí watershed algoritmu. Za předpokladu, že máme k dispozici odhad polohy vozidla, je detekce RZ pomocí klíčových bodů přesnější než při prohledávání celého obrazu.

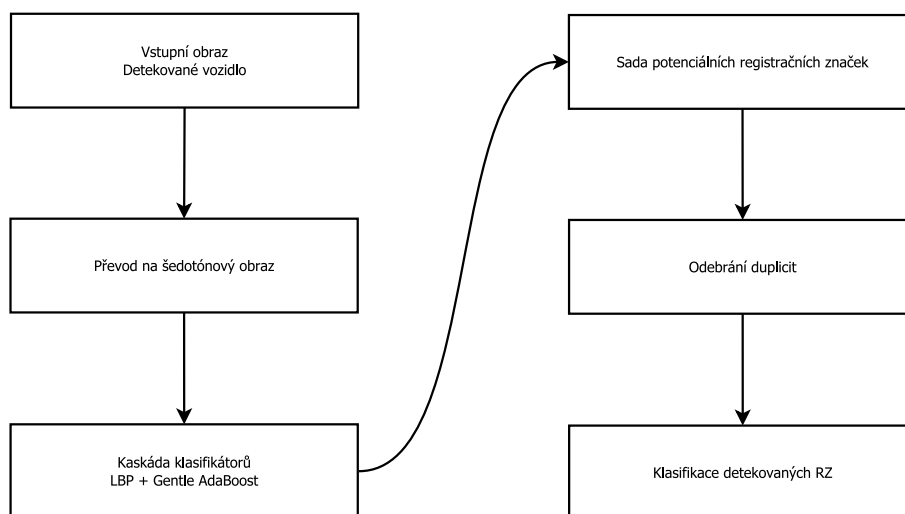
Algoritmus detekce RZ pomocí klíčových bodů je založen na následujícím principu. V první fázi je obraz předzpracován a jsou detekovány klíčové body, které budou použity pro další zpracování. V dalším kroku je provedeno prahování obrazu, na jehož výsledek jsou aplikovány morfologické operace, pro následující výpočet ohraničení obrazu pomocí Watershed algoritmu. Následujícím krokem je nalezení oblasti pro počáteční bod semínkového vyplňování pomocí zjištění nejvyšší koncentrace klíčových bodů v obraze a získání vlastní masky semínkového vyplňování. Posledním krokem je potom výpočet minimálního ohraničení získané masky, které je použito pro redukci oblasti zájmu na samotnou registrační značku vozidla.

Tento algoritmus má dobrou úspěšnost detekce registrační značek ve statických snímcích, která pro některé testovací sady přesahuje 85%. Průměrná úspěšnost potom přesahuje

76%. Při simulaci záběrů dopravní kamery však algoritmus neposkytoval potřebnou úspěšnost detekce registračních značek, jelikož nebyl například schopen detekovat a správně separovat celou registrační značku vozidla. Z toho důvodu bylo od tohoto algoritmu upuštěno, a bylo přistoupeno k algoritmům detekce využívajících strojového učení.

3.3.2 Detekce RZ pomocí kaskády klasifikátorů

Nejvhodnějším detekčním algoritmem využívajícím k detekci strojové učení byla zvolena kaskáda klasifikátorů. Systém detekce RZ pomocí kaskády klasifikátorů je navržen obecně, aby byl schopen detekovat RZ vozidel bez znalosti jejich polohy, tj. bez dodaných ohraničení vozidel, ale jejich dodání razantně urychlí čas potřebný k vyhledávání značky v obraze. Jelikož ohraničení vozidel je ve většině případů reprezentováno obecným čtyřúhelníkem, je pro některé algoritmy vhodné v případě jejich dodání obraz transformovat. Navržený systém se však do jisté míry zvládá vypořádat s drobným natočením či perspektivou registrační značky.



Obrázek 3.12: Diagram principu detekce registrační značky vozidla.

Jak vyplývá z diagramu 3.12, systém detekce RZ pomocí kaskády klasifikátorů se skládá z několika fází. V první fázi se jedná o předzpracování vstupního obrazu pro jeho následnou detekci pomocí kaskády klasifikátorů. Kaskáda klasifikátorů použitá v této práci je založena na algoritmu *Gentle AdaBoost*, který využívá příznaky *Local Binary Pattern (LBP)*.

Local Binary Pattern

Metoda LBP – Local Binary Pattern (česky – lokální binární vzor), se využívá k tvorbě příznaků, které charakterizují vlastnosti obrazu. V případě metody LBP je příznak reprezentován histogramem. Základní verze LBP pro oblast 3×3 pixely je pro získání výsledné

hodnoty středového pixelu v podstatě odvozena jako porovnání středového pixelu se svým okolím. Pro body P a \mathbf{P}_x proces závisí na prahování, které je zde funkcí

$$s(x) = \begin{cases} 1 & \text{pokud } \mathbf{P}_x > P \\ 0 & \text{jinak} \end{cases} \quad (3.3)$$

Výsledná hodnota prostředního pixelu je potom stanovena pomocí binárního váhování aplikovaného na výsledek prahování. Tedy LBP kód pro bod P s osmi sousedními body x je vypočítán jako

$$\text{LBP} = \sum_{x \in 1..8} s(x) \times 2^{x-1} \quad (3.4)$$

Princip výpočtu LBP je ukázán na obrázku 3.13, kdy bod pro výpočet LBP je středový bod a osm hodnot na pozicích x okolo něj je jeho osm nejbližších sousedních bodů. Více informací ohledně metody Local Binary Pattern je uvedeno v knize [19].

118	190	6	1	1	0	1	128	64	1	128	0
69	106	110	0		1	2		32	0	161	32
42	31	106	0	0	0	4	8	16	0	0	0

Obrázek 3.13: První matice zobrazuje oblast snímku 3×3 pixely. Druhá matice obsahuje výsledek prahování (kód 10100001_2). Třetí matice zobrazuje váhy jednotlivých pixelů. Poslední matice potom zobrazuje výslednou hodnotu a její přispěvatele.

Gentle AdaBoost a kaskáda klasifikátorů

AdaBoost je algoritmus konstrukce silnějšího klasifikátoru z jednodušších (slabších) klasifikátorů $h_t(x) : \chi \rightarrow \{-1, +1\}$, pomocí jejich lineární kombinace, dle vzorce

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x), \quad (3.5)$$

kde $h_t(x)$ je základní (slabší) klasifikátor, hypotéza nebo příznak; α_t je jeho váha, a rovnice

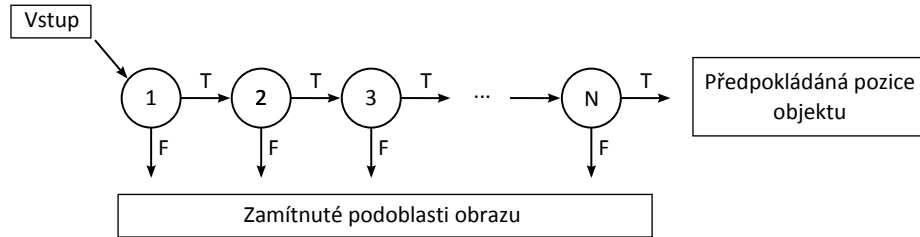
$$H(x) = \text{sign}(f(x)) \quad (3.6)$$

je výsledný (silnější) klasifikátor/hypotéza. Snahou je v každé iteraci minimalizovat chybu jednotlivých slabších klasifikátorů a vytvořit tak klasifikátor silnější. V každém kroku trénování klasifikátoru pomocí metody AdaBoost jsou eliminovány slabé klasifikátory a vybere se vždy ten nejsilnější klasifikátor pro daný krok. Více informací o AdaBoostu je k dispozici ve článku [8].

Gentle AdaBoost je potom jedna z variant základního algoritmu AdaBoost, který se snaží vybírat $f_t(x)$ uvážlivě pomocí pevného kroku, namísto minimalizace chyby klasifikátoru. Výběr $f_t(x)$ je prováděn tak, aby došlo k minimalizaci $\sum_i w_{t,i} (y_i - f_t(x_i))^2$. Více

informací o této i jiných variantách metody AdaBoost je k dispozici ve článku [9].

Kaskáda klasifikátorů je potom implementací některého algoritmu konstrukce silného klasifikátoru z jednodušších klasifikátorů, které jsou postupně aplikovány na oblasti zájmů v obraze, jež jsou buď zamítnuty, nebo předány dalšímu stupni klasifikace. Pokud jsou daná oblast zájmu projde až na konec kaskády, je velmi pravděpodobné, že obsahuje hledaný objekt. Ukázka principu fungování kaskády klasifikátorů je zobrazena na obrázku 3.14. Kaskáda klasifikátorů použitá v této práci je vytvořena pomocí algoritmu Gentle AdaBoost.



Obrázek 3.14: Ukázka struktury kaskády klasifikátorů.

3.3.3 Detekce RZ vyhledáváním geometrických tvaru

Dalším možným přístupem k detekci registrační značky v obraze může být vyhledávání geometrické tvaru v obraze pomocí Houghovy transformace pro vyhledávání přímek či generalizované Houghovy transformace pro vyhledávání téměř libovolných tvarů.

Použití Houghovy transformace je vhodné zkombinovat se znalostí přesné pozice přední, resp. zadní části vozidla. Díky této znalosti bychom měli být schopni vyjádřit tyto úsečky jako přímky v Houghově prostoru a zredukovat tak počet přímek nalezených pomocí Houghovy transformace.

V dalším kroku je vhodné provést redukci počtu detekovaných přímek odpovídajících stejné hraně. V tomto případě lze využít principu duality v duálním prostoru, kdy lze zaměnit vlastnosti přímek a bodů. Z toho důvodu můžeme říct, že přímky prochází jedním bodem a body leží na jedné přímce, a pravdivost libovolného tvrzení se nezmění. Průsečík těchto detekovaných přímek tedy můžeme reprezentovat přímkou a vyhledat nejbližší body (jiná prezentace přímek) k této přímce. Tímto způsobem lze provést shlukování detekovaných přímek s podobnou orientací.

Poslední krok nám potom zbývá na nalezení průsečíků přímek, které odpovídají hranám registrační značky vozidla. V tomto kroku by bylo vhodné použít znalosti o velikosti a poměru stran registrační značky vozidla pro vylepšení algoritmu vyhledávání průsečíků. Výsledkem by pak měli být 4 body reprezentující jednotlivé rohy registrační značky vozidla.

Houghova transformace

Houghova transformace je technika lokalizace geometrických tvarů v obraze. Nejčastěji bývá používána na detekci přímk, kružnic a elips. V případě přímk je matematická definice ekvivalentní k Radonově transformaci. Houghova transformace definuje mapování bodů obrazu do prostoru akumulátoru (Houghova prostoru).

V Kartézském souřadném systému je přímka v homogenních souřadnicích vyjádřena jako

$$Ay + Bx + 1 = 0, \quad (3.7)$$

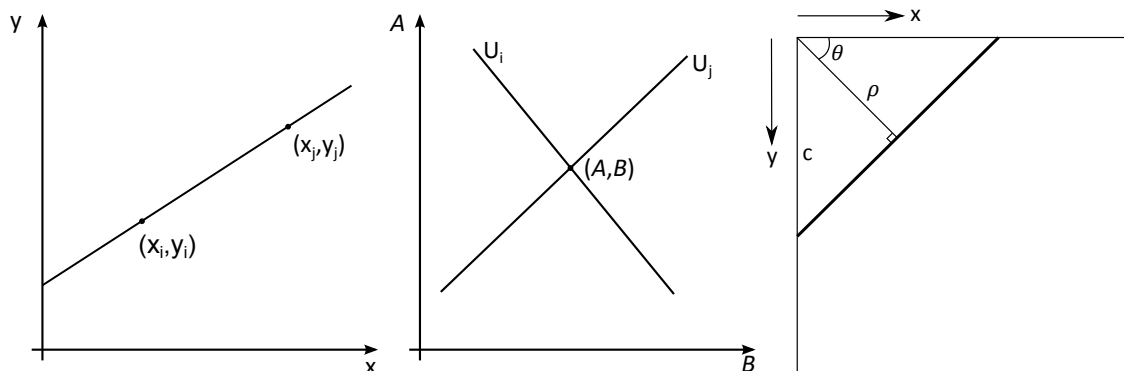
kde dvojice hodnot (A, B) definuje přímku. Rovnice (3.7) však obsahuje jistou symetrii, jelikož také dvojice souřadnic (x, y) definuje přímku. Houghova transformace shromažďuje důkaz o bodu (A, B) s předpokladem, že všechny body (x, y) definují stejnou přímku v prostoru (A, B) . To znamená, že v případě, že množina kolineárních bodů $\{(x_i, y_i)\}$ definuje přímku (A, B) , poté

$$Ay_i + Bx_i + 1 = 0 \quad (3.8)$$

Přímka v Houghově prostoru je reprezentována rovnicí

$$\rho = x \cdot \cos \theta + y \cdot \sin \theta, \quad (3.9)$$

kde ρ je nejkratší vzdálenost od počátku souřadného snímku obrazu (horní levý roh) k bodu ležícímu přímce a θ je úhel mezi osou x a tímto bodem. Rovnice (3.9) představuje vyjádření přímky v polárních souřadnicích. Obrázek 3.15 zobrazuje vizualizaci parametrů detekované přímky. Více informací o Houghově transformaci je uvedeno v knize [20].



Obrázek 3.15: (Vpravo) Obraz obsahující přímku v Kartézském souřadném systému. (Uprostřed) Přímky v duálním prostoru (A, B) . (Vlevo) Repräsentace přímky v polárních souřadnicích a vizualizace parametrů přímky získaných Houghovou transformací.

Generalizovaná Houghova transformace

Mnoho tvarů je daleko komplexnějších než přímky, kružnice a elipsy. Často sice můžeme rozdělit komplexní tvar na několik základních geometrických primitiv, ale může to vést

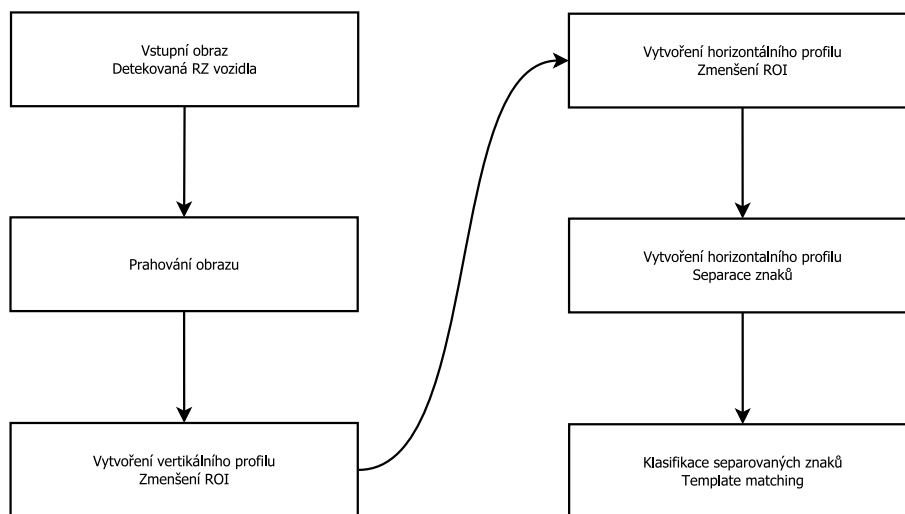
k velmi komplexním strukturám, což může být nevýhodné, a proto je v podstatě vhodnější extrahovat celý tvar najednou. Takovéto mapování celého tvaru najednou se nazývá Generalizovaná Houghova transformace, kterou lze využít na lokalizaci libovolného tvaru s neznámou pozicí, velikostí i orientací. Generalizovaná Houghova transformace může být formálně definovaná pomocí duality křivky. Více informací o Generalizované Houghově transformaci, stejně jako Houghově transformaci obecně, je uvedeno v knize [20].

3.4 Metody rozpoznání RZ

Rozpoznání registrační značky vozidla, dále jen RZ, je pro tento systém další klíčovou záležitostí. Bez tohoto rozpoznání by nebylo možné vozidlo re-identifikovat a případně i určit, o kolikátý výskyt vozidla se jedná. Při návrhu a vývoji této části systému jsem se zaměřil na několik různých způsobů rozpoznání registrační značky vozidla.

První návrh zvažoval metodu *template matching* popsanou v sekci 3.4.1, která však potřebuje přesnou segmentaci jednotlivých znaků registrační značky. Pro rozpoznání registrační značky bez nutnosti segmentace jednotlivých znaků byl navržen zcela nový princip rozpoznání registrační značky popsaný v sekci 3.4.2.

3.4.1 Template matching pro rozpoznání znaků RZ



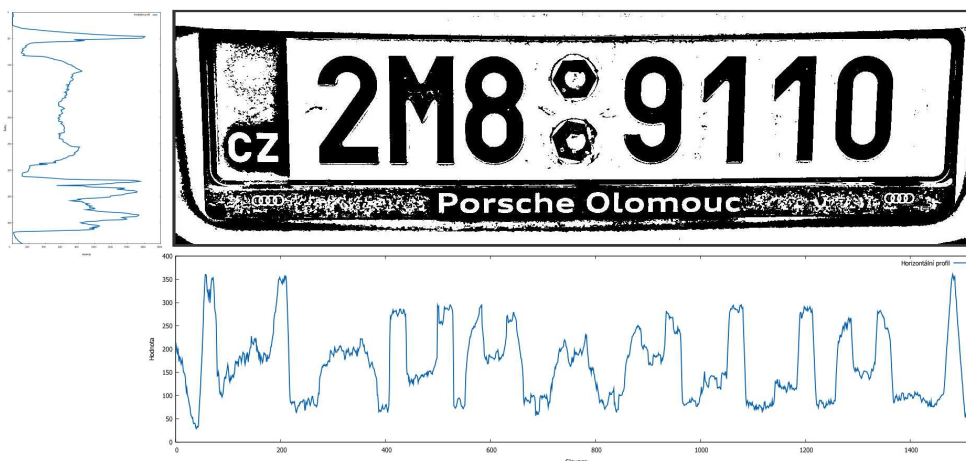
Obrázek 3.16: Diagram rozpoznání znaků registrační značky.

První metodou rozpoznání RZ vozidla, kterou se budeme zabývat, je metoda označovaná jako *Template Matching*. Tato metoda slouží k porovnání shody dvou obrazů, tedy šablony a dané vstupní oblasti. Hlavní výhodou použití metody *Template matching* jako nástroje pro rozpoznání znaků registrační značky vozidla je ten, že si vystačí s minimální datovou sadou. Je potřeba pouze jeden vzorek, šablona, od každého znaku, který chceme rozpoznávat. Na druhou stranu je potřeba separovat jednotlivé znaky registrační značky, abychom

dostali očekávané výsledky. Princip rozpoznání RZ pomocí metody *Template Matching* je reprezentován diagramem na obrázku 3.16. Jednou z důležitých součástí tohoto principu je kromě datové sady šablon, také vytvoření kvalitního separátoru jednotlivých znaků RZ. Bližší informace a různé principy využití metody *Template matching* jsou k dispozici v knize [21].

Vytvořil jsem systém, který z detekované RZ vozidla a převedením jejího obrazu do binárních hodnot postupně vyseparoval znaky pomocí horizontálních a vertikálních profilů intenzit. Pod pojmem profil si zde můžeme představit jistou formu histogramu, kde jednotlivé biny histogramu reprezentují index řádku/sloupce vstupního obrazu a jejich hodnotu je určena součtem pixelů s nulovou intenzitou v řádku/sloupci.

V profilech jsou poté od středu ke krajům vyhledávány tzv. *peaky*, tj. oblasti s nejvyšší koncentrací pixelů s nulovou intenzitou v binu profilu. Od těchto peaků jsou následně vyhledávána minima směrem zpět ke středu obrazu. Pozice těchto minim v profilu se použijí jako indexy pro zmenšení oblasti zájmu. Podobný systém byl navržen ve článku [1]. Avšak tento systém využívá horizontální a vertikální profil hran, namísto profilů intenzit.



Obrázek 3.17: Ukázka ideálního vstupu s výsledkem profilace. Vertikální (vlevo) a horizontální profil (dole) intenzity registrační značky vozidla společně se vstupní obrazem registrační značky. V horizontální profilu si lze všimnout precizního oddělení jednotlivých znaků.

Nízké rozlišení detekovaných RZ ze záběrů simulujících dopravní kamery však způsobilo, že znaky RZ nejsou dostatečně odděleny a bylo tedy téměř nemožné je v některých případech korektně separovat. Pokud by nebyly znaky správně separovány, nebyla by tato metoda použitelná. Z toho důvod byl navržen zcela nový způsob rozpoznávání znaků RZ, popsáný v sekci 3.4.2.

Princip separace znaků

Nejprve je vypočítán vertikální profil řádků obrazu dle rovnice 3.11, podle kterého se určí horní a spodní hrana znaků RZ vozidla. Následně je vypočítán horizontální profil sloupců

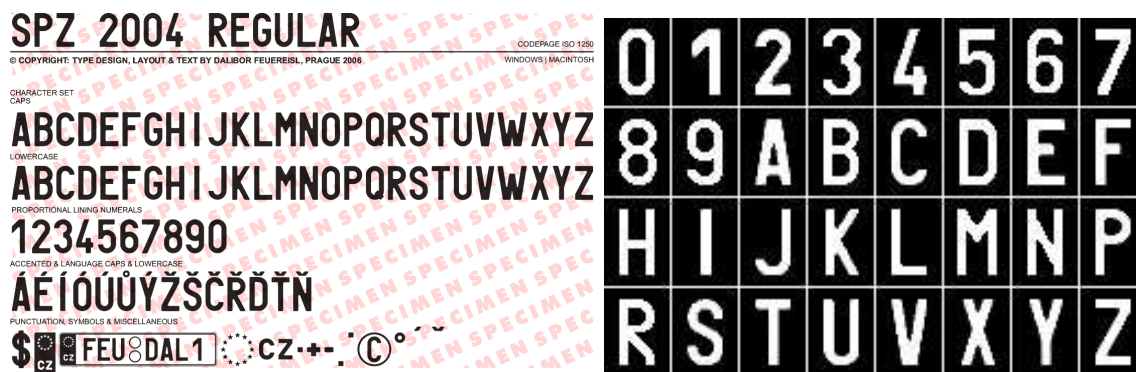
obrazu dle rovnice 3.12, podle které se určí levé a pravé ohraničení znaků RZ vozidla. Obě tyto rovnice využívají funkci $countZeros(I)$ uvedenou v rovnici 3.10, která vrací počet pixelů s nulovou intenzitou, tedy pixely černé barvy, v zadaném řádku/sloupci matice vstupního obrazu A_{ij} .

$$countZeros(I) = \sum_{I:src(I)=0} 1 \quad (3.10)$$

$$VP[i] = countZeros(A_i) \quad (3.11)$$

$$HP[i] = countZeros(A_j) \quad (3.12)$$

Separace RZ na jednotlivé znaky je opět prováděna pomocí horizontální profilů, kdy jsou separovány jednotlivé peaky pomocí okolních minim. Separované znaky je následně nutné zpracovat a předchystat tak pro jejich rozpoznání. V této fázi je využíváno prahování a morfologické operace pro získání čisté podoby separovaného znaku.



Obrázek 3.18: (Vlevo) Ukázka fontu SPZ Regular používaného pro výrobu registračních značek v České republice. Obrázek byl převzat z [7]. (Vpravo) Ukázka šablon pro metodu *template matching* vytvořených kombinací sesbíraných vzorků dat. V datové sadě chybí znaky G, O, Q a W, kvůli zákazu používání těchto znaků na registračních značkách.

Rozpoznání znaků

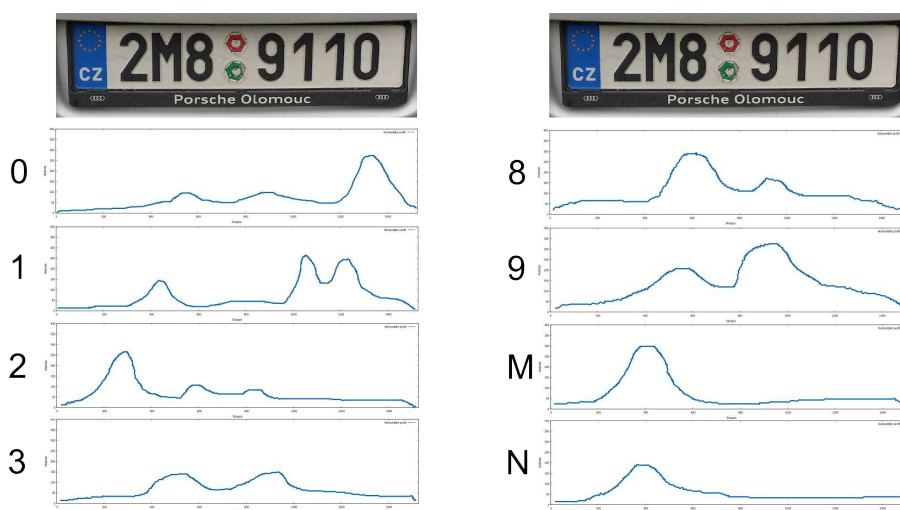
Samotné rozpoznání potom funguje na principu tzv. *template matching*, kdy je vytvořena sada šablon, anglicky *templates*, které jsou následně porovnávány se separovanými znaky. Ukázka fontu použitého pro tvorbu registračních značek České republiky a vytvořená datová sada šablon pro jednotlivé akceptovatelné znaky je zobrazena na obrázku 3.18. Porovnání pracuje na principu překrývání, kdy jsou znaky nejprve srovnány na stejnou velikost a následně je vyhodnocena procentuální úspěšnost překrytí. Znak s nejvyšším stupněm překrytí je následně vyhodnocen jako výsledek porovnání.

3.4.2 Rozpoznání RZ pomocí neuronové sítě a sliding window skenování

Navrhl jsem nový systém rozpoznání znaků, který již nepočítá s rozdělením RZ na jednotlivé znaky, ale bere ji jako celek. Princip tohoto systému spočívá ve vytvoření neuronové sítě natrénované pro jednotlivé znaky, které se vyskytují na registračních značkách, a skenování detekované registrační značky pomocí *sliding window*.

Na každou pozici, kterou bude projíždět *sliding window* během skenování detekované RZ, bude spuštěna klasifikace pomocí neuronové sítě, která vrátí pro každý znak ohodnocení dané pozice pomocí pravděpodobnosti, respektive věrohodnosti (likelihood). Výsledkem klasifikace celé RZ by potom měl být jakýsi průběh věrohodností pro každý znak a každou pozici *sliding window*. Pro lepší představu je idea výstupu klasifikace zobrazena na obrázku 3.19.

Na základě průběhu ohodnocení všech znaků pro daná vstupní data jsme schopni zvolit nejlépe ohodnocený znak pro každou pozici registrační značky. Tímto způsobem vznikne řetězec, který nejvíce odpovídá registrační značce dodané na vstupu. Posledním krokem rozpoznání registrační značky vozidla je potom samotná finalizace a validace získaného řetězce, která je navržena v sekci 3.4.4. Tímto krokem bychom měli být schopni vypustit klasifikaci registrační značky pomocí *Support Vector Machine*, která je uvedena v diagramu 3.1, jelikož při předložení snímku neobsahujícího registrační značky, by krok finalizace a validace RZ neměl vrátit žádný výsledek.



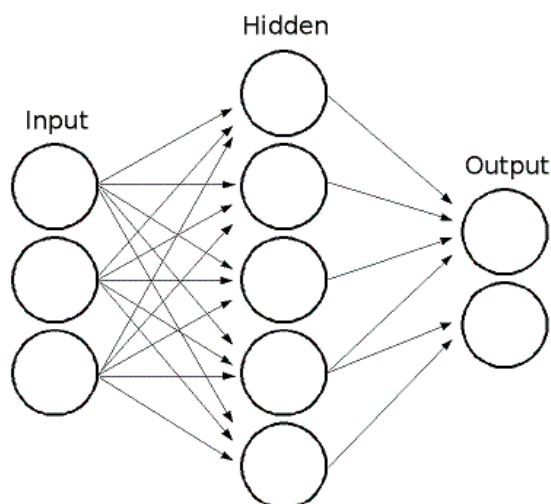
Obrázek 3.19: Ukázka předpokládaného výstupu klasifikace. Na obrázku je zobrazena detekovaná registrační značka vozidla a ohodnocení vybraných znaků pro jednotlivé pozice ve snímku.

Za předpokladu, že máme k dispozici obalový kvádr vozidla, jehož výpočet byl popsán v sekci 3.1.2, a jsme tím pádem schopni provést transformaci obrazu, která je popsána v sekci 3.1.4, bychom měli být schopni vypočítat také transformaci opačnou, tj. z pravoúhelníku do obecného čtyřúhelníku. Pokud touto inverzní transformační maticí transformujeme datovou sadu neuronové sítě pro každý jízdní pruh, a tedy provedeme její trénování pro každý

jízdní pruh samostatně, bude možné rozpoznávat registrační značky vozidla bez nutnosti transformace obrazu.

Neuronová síť

Neuronová síť je matematický výpočetní model, používaný v umělé inteligenci, se vzorem chování odpovídající skutečné neuronové síti. Umělá neuronová síť se skládá z umělých neuronů a je určena pro zpracování dat. Každý umělý neuron má libovolný počet vstupů, ale vždy jen jeden výstup. Aby bylo umožněno si předávat informace, jsou jednotlivé neurony mezi sebou vzájemně propojeny a transformují se pomocí přenosové funkce. Důležitou součástí je učení neuronové sítě. Existují dva přístupy k učení (trénování) neuronové sítě – učení s učitelem (supervizor) a učení bez učitele. První uvedený princip upravuje váhy jednotlivých neuronů tak, aby na daný vstup odpověděla síť požadovaným výstupem. U druhého principu výstup nevyhodnocujeme a síť si sama třídí vstupní vzorky. Hlavní využití v oblasti zpracování obrazu nachází neuronové sítě při rozpoznávání (klasifikaci) objektů v obrazu.



Obrázek 3.20: Struktura 3-vrstvé perceptronové sítě se třemi vstupy, dvěma výstupy a pěti neurony v prostřední vrstvě. Neurony jednotlivých vrstev jsou vzájemně propojeny. Převezato z [24].

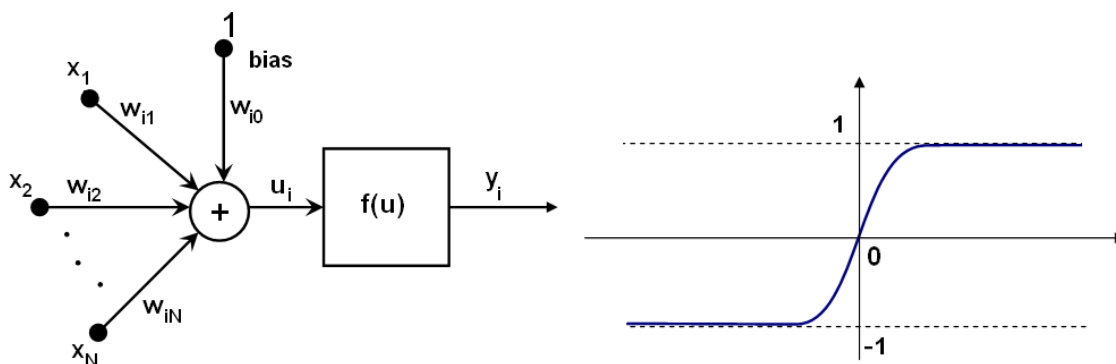
Neuronová síť použitá v této práci je vícevrstvá síť perceptronů, která se skládá z jedné vrstvy vstupní a výstupní, mezi kterými se nachází jedna či více vrstev skrytých. Každá vrstva obsahuje minimálně jeden neuron přímo propojený s neurony předchozí i následující vrstvy. Vizualizace struktury neuronové sítě a jejího propojení je k dispozici na obrázku 3.20. Všechny neurony použité v síti jsou stejné. Jejich struktura je zobrazena na obrázku 3.21. Jednotlivé neurony mají několik vstupů, na kterých přebírají výstupy neuronů z předchozí vrstvy, a jeden výstup, pomocí kterého posílají hodnoty pro zpracování další vrstvou. Hodnoty získané z předchozí vrstvy jsou sečteny společně s určitými váhami, individuál-

ními pro každý neuron, a „biasem“. Tato suma je dále transformována pomocí přenosové (aktivační) funkce. Vzorec

$$u_i = \sum_j (w_{i,j}^{n+1} \times x_j) + w_{i,bias}^{n+1} \quad (3.13)$$

$$y_i = f(u_i) \quad (3.14)$$

vyjadřuje výpočet výstupu y_i pro vrstvu $n + 1$ s použitím výstupů x_j vrstvy n , vizualizovaného obrázkem 3.21. [24]



Obrázek 3.21: (Vlevo) Struktura neuronu. (Vpravo) Přenosová funkce typu symetrické sigmoidy. Převzato z [24].

3.4.3 Analýza příznaků datové sady pro neuronové sítě

Základním typem příznaků pro trénování neuronové sítě je informace o intenzitě jednotlivých pixelů v obraze. Získání tohoto typu příznaku je možné provést pomocí de-saturace, neboli převodu vstupního barevného snímku do snímku šedotónového. Informace o intenzitě není bohužel ve většině případů dostatečná, jelikož pouhá změna úhlu pohledu na detekovanou registrační značku může vést k selhání rozpoznání kvůli změně rozložení intenzity pixelů. Z toho důvodu je vhodné zvážit i další příznaky, které lépe reprezentují vstupní informace a dění v obraze. Vhodnými příznaky mohou být například informace o umístění a směru hran v obraze nebo míra změn v obraze. Tyto informace můžeme získat pomocí aplikace operátoru první a druhé derivace na vstupní data. Možným rozšířením příznaků pro trénování neuronové sítě je tedy například detekce hran či detekce frekvencí. Vytvářením různých typů příznaků pro klasifikaci obrazu se zabývá článek [3].

Jedním z možných způsobů detekce hran je aplikace Sobelova operátoru na vstupní obraz. Sobelův operátor pracuje vždy s určitým jádrem ($3 \times 3, 5 \times 5, 7 \times 7, \dots$), které je aplikováno na vstupní obraz pomocí principu kros-korelace. Tímto způsobem lze získat vektory gradientů obrazu v jednotlivých směrech, podle typu aplikovaného jádra. Pro detekci hran užitím Sobelova operátoru se v počítačovém vidění po získání vektoru gradientu počítá jeho norma. Bližší popis Sobelova operátoru je uveden níže.

Způsobem, kterým lze zajistit detekci frekvencí v různých směrech, je aplikace Gaborových filtrů na vstupní obraz. Základem detekce frekvencí pomocí Gaborových filtrů je vytvoření konvolučního jádra, které je následně aplikováno principem kros-korelace na vstupní data. Výpočet konvolučního jádra je velmi dobře parametrizovatelný, a proto je možné provádět detekci frekvencí v různých směrech, v závislosti na vstupních parametrech funkce. Detekované frekvence nám podávají informace o hranách v obraze. Bližší informace o Gaborových filtrech jsou popsány níže.

Sobelův operátor

Sobelův operátor je klasický lineární filtr pro detekci hran, jehož základní forma je založena na jednoduchém jádru 3×3 , jehož struktura je zobrazena na obrázku 3.22.[16]

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

Obrázek 3.22: První matice reprezentuje jádro pro detekci hran ve vertikálním směru. Druhá pak reprezentuje jádro Sobelova operátoru pro detekci hran ve směru horizontálním.

Pokud se díváme na obraz jako na dvourozměrnou funkci, můžeme Sobelův operátor vnímat jako míru změn v obraze ve vertikálním a horizontálním směru. Z matematického pohledu se tato míra nazývá gradient a je definována jako dvourozměrný vektor vytvořený první derivací funkce obrazu ve dvou ortogonálních směrech

$$grad(I) = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]^T \quad (3.15)$$

Sobelův operátor tedy udává aproximaci gradientu obrazu diferenciací pixelů ve vertikálním a horizontálním směru. Sobelův operátor operuje s malým oknem o velikosti daného jádra kolem určeného pixelu za účelem omezení vlivu šumu v obraze. Jestliže je gradient dvourozměrný vektor, tak má svou normu a směr. Norma vektoru gradientu vypovídá o amplitudě změn v obraze a obvykle se počítá jako Euklidovská norma (také známé jako $L2$ norma)

$$|grad(I)| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (3.16)$$

Tato norma udává vzdálenost od počátku souřadného systému. Vektor gradientu vždy směřuje do směru největší změny. Pro obraz to znamená, že směr gradientu bude ortogonální ke hraně směřující od nižší k vyšší intenzitě. Rovnicí

$$\angle grad(I) = a \tan \left(-\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x} \right) \quad (3.17)$$

je potom dán směr úhlu gradientu. [16]

Gaborovy filtry

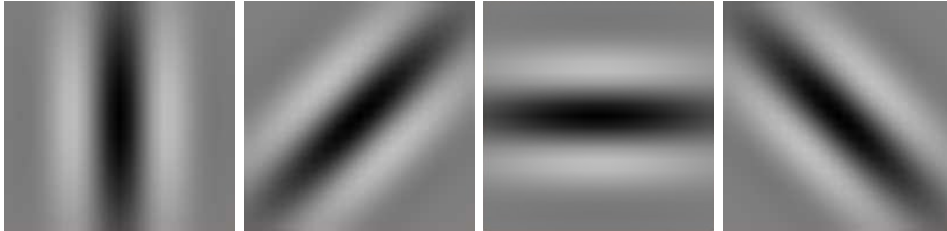
Gaborův filtr je reprezentantem operátoru druhé derivace a je používán k detekci frekvencí v různých směrech. Jedná se o lineární filtr, který je definován jako součin Gaussovy a harmonické funkce, kde Gaussova funkce ohraničuje funkci harmonickou, charakteristickou svým směrem a frekvencí. Gaborův filtr je dán komplexní funkcí

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{x'}{\lambda} + \psi\right)\right) \quad (3.18)$$

$$x' = x \cos \theta + y \sin \theta \quad (3.19)$$

$$y' = -x \sin \theta + y \cos \theta, \quad (3.20)$$

kde x a y jsou vstupní proměnné, λ je vlnová délka sinusovského faktoru, θ je orientace normály k paralelním prvkům Gaborovy funkce ($0 - 180^\circ$), ψ je posunutí fáze, σ je směrodatná odchylka Gaussovské obálky a γ je poměr stran v prostoru. Obrázek 3.23 zobrazuje vizualizaci vypočítaných jader Gaborových filtrů pro různý úhel natočení θ .



Obrázek 3.23: Vizualizace jader Gaborova filtru. Zleva $\theta = 0^\circ$, $\theta = 45^\circ$, $\theta = 90^\circ$, $\theta = 135^\circ$. Pro všechny filtry je velikost jádra zvolena na hodnotu 21, $\sigma = 4.0/21.0$, $\lambda = 10.0$, $\gamma = 0.5$, $\psi = 180^\circ$.

3.4.4 Finalizace a validace rozpoznané RZ

Posledním krokem rozpoznání registrační značky vozidla metodou navrženou v kapitole 3.4.2, je finalizace a validace obsahu RZ z rozpoznaného řetězce registrační značky. V tomto kroku je potřeba vybrat nejlépe ohodnocené znaky ze skupin odpovídajících pozici jednotlivých znaků registrační značky.

Předpokládejme, že pro registrační značku 6T9 7316 bude výsledný nejlépe ohodnocený řetězec získaný klasifikátorem vypadat následujícím způsobem:

```
8666666_111TTTTT5_8099999_6_____3____2773777____3333336__111111_66666666
```

Z tohoto řetězce je člověk schopen sestavit nejpravděpodobnější registrační značku s pouhou znalostí počtu požadovaných znaků. Přechody mezi jednotlivými znaky jsou přehledně odděleny pomocí znaku _, a lze tak zvolit nejpravděpodobnější znak pro každou skupinu. Z pohledu stroje to ovšem tak jasné není.

Je potřeba vytvořit systém, který ze vstupního řetězce bude schopen separovat jednotlivé skupiny znaků, jež budou dále zpracovány pro výběr nejvěrohodnějšího reprezentanta.

Pro vytvoření tohoto systému by bylo pravděpodobně možné použít nějakou derivaci Hidden Markovova Modelu, který se používá u systému zpracování řeči, a zpracovávat vstupní řetězec s přihlédnutím k jeho celkové délce i délkám skupin znaků, a použitím prediktoru. Výstupem tohoto modelu by měly být skupiny znaků odpovídající počtu znaků registrační značky. Na každou získanou skupinu by měla proběhnout opět aplikace Hidden Markovova Modelu pro získání nejvěrohodnějšího znaku.

Kapitola 4

Implementace a vyhodnocení

Tato kapitola popisuje dosavadní průběh implementace jednotlivých částí systému. Všechny vytvořené aplikace jsou implementovány v jazyce C++ s využitím open source knihovny OpenCV. Při implementaci částí systému re-identifikace vozidel byly provedeny různé experimenty s jednotlivými moduly tohoto systému. Tyto experimenty představují testování jednotlivých metod popsaných v rámci návrhu systému, tak i jejich nastavení. Jsou zde uvedeny veškeré informace od velikosti datových sad počínaje, přes implementační detaily, po vyhodnocení jednotlivých metod konče.

4.1 Pořízená datová sada

Pro účely testování bylo pořízeno více jak 150 minut video záznamu ve FullHD kvalitě pořízené digitálním fotoaparátem, obsahující scénu zachycující pohled na dopravní komunikaci z různých úhlů pohledu a s různým nastavením. Dále bylo pořízen týdenní záznam z bezpečnostní kamery směřující na dopravní komunikaci, v rozlišení 2800×1600 pixelů, který bude postupně zpracováván a s jeho pomocí bude rozšiřována datová sada registračních značek a jejich znaků.

Pomocí „on-board“ kamery umístěné na předním skle automobilu byla rozšiřována sada snímků registračních značek, která nyní čítá přes 4650 vzorků. Tyto registrační značky budou dále zpracovávány a použity pro rozšíření datové sady pro trénování neuronové sítě.

Byla vytvořena také sada separovaných znaků registrační značky pro vytvoření šablon používaných metodou *template matching*. Pro testování a vyhodnocení detekčních metod byla dále vytvořena testovací množina statických snímků, která čítá celkem 488 testovacích snímků. Bližší informace o této datové sadě jsou k dispozici v sekci 4.4.3 a v tabulce 4.1.

4.2 Pomocné aplikace

Během návrhu a implementace systému re-identifikace vozidel v obraze bylo potřeba vytvořit pomocné aplikace, které by nám usnadnily zpracovávání sesbíraných dat a dále pak například vizualizovat dopad různého nastavení Gaborových filtrů pro získání příznaků datové sady pro neuronové sítě.

4.2.1 Zpracování videozáznamu na sekvenci snímků

Rozsáhlou datovou sadu pořízenou „on-board“ kamerou, umístěnou na čelním skle automobilu, bylo potřeba zpracovat pro jednodušší získávání vzorků různorodých registračních značek vozidel České republiky.

Z toho důvodu vznikla aplikace `video2frames`, která má za úkol zredukovat velikost datové sady co se objemu dat týče. Aplikace převádí vstupní video soubor na sekvenci snímků. Pro zredukování objemnosti dat je ukládán pouze každý 10 snímek videa. Primárně je aplikace určena pro zpracovávání záznamu „on-board“ kamery, ale je použitelná na jakýkoli jiný video záznam.

4.2.2 Získávání registračních značek ze sekvence snímků

Následujícím krokem zpracování sekvence snímků, která vznikla ze záznamu „on-board“ kamery, je získávání registračních značek.

K tomuto účelu vznikla aplikace `frames2licensePlate`, která s použitím kaskády klasifikátorů, popsané v sekci 3.3.2, rozšiřuje datovou sadu pro trénování neuronové sítě. Aplikace tedy vyhledává registrační značky ve vstupní sekvenci snímků, čímž rozšiřuje již zmíněnou datovou sadu pro trénování a testování neuronové sítě.

Princip vyhledávání registračních značek je velmi podobný tomu, který je popisován v sekci 3.3.2, nicméně v tomto případě se prohledává celý vstupní obraz. Resp. prohledává se ta část obrazu, která neobsahuje časové údaje „on-board“ kamery.

Ukázka datové sady získané z „on-board“ kamery pomocí aplikace `frames2licensePlate` je k vidění na obrázku 4.1. Registrační značky získané touto metodou se vyznačují různými rozměry, rozlišením, natočením, ale i světelnými podmínkami v závislosti na vzdálenosti detekované registrační značky a denní době i okolních podmínkách při pořízení záznamu.

Celkem bylo ze záběrů „on-board“ kamery získáno přes 4650 snímků registračních značek. Sběr dat probíhal nejčastěji na trase Olomouc – Brno – Olomouc, proto jsou v datové sadě nejvíce zastoupeny právě registrační značky z těchto krajů. Velikost této datové sady ale není zdaleka konečná a je nutné ji ještě zvětšit z důvodu pokrytí většího množství možných znaků registrační značky.



Obrázek 4.1: Ukázka datové sady registračních značek získaných pomocí aplikace `frames2licensePlate` ze záběrů „on-board“ kamery, umístěné na čelním skle vozidla. Registrační značky získané touto metodou se vyznačují různými rozměry, světlými podmínkami i rozlišením, v závislosti na vzdálenosti detekované registrační značky.



Obrázek 4.2: Ukázka registrační značky 2M5 9261 s vyznačením principu segmentace na jednotlivé řezy v průběhu skenování registrační značky pomocí *sliding window*.



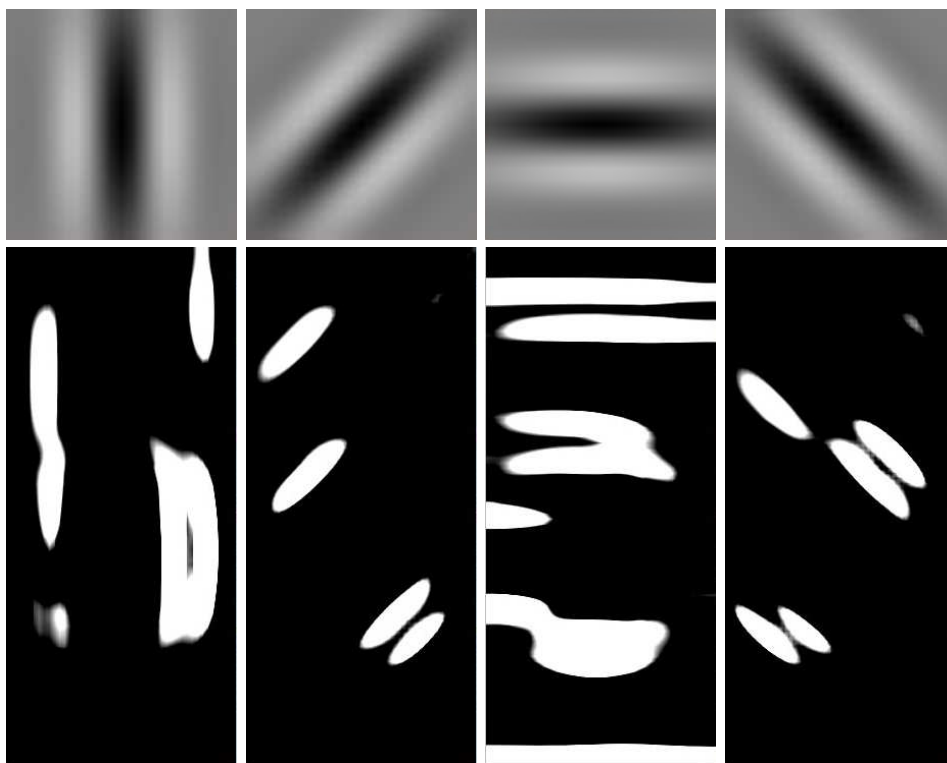
Obrázek 4.3: Ukázka RZ 2M5 9261, rozdělená pomocí skenování *sliding window* na jednotlivé segmenty. Každý z těchto segmentů je poté použit na trénování neuronové sítě.

4.2.3 Příprava datové sady pro trénování neuronové sítě

Posledním krokem pro přípravu datové sady, pro trénování a testování neuronové sítě, je rozdělení registračních značek na jednotlivé segmenty. Princip segmentace registrační značky na jednotlivé řezy (segmenty) byl představen v sekci 3.4.2. Pro tyto účely byla vytvořena aplikace `licensePlate2segments`, která má za úkol převést získané registrační značky na sady segmentů. Vstupní snímek je zobrazen na obrázku 4.2, zpracovaný výstup potom na obrázku 4.3.

4.2.4 Vizualizace Gaborových filtrů

Aplikace `gaborFeatures` byla implementována z důvodu jednoduššího a přehlednějšího testování aplikace Gaborových filtrů na datovou sadu klasifikátoru znaků. Aplikace nabízí vizualizaci vypočítaného jádra Gaborova filtru, vstupních dat i finálního výsledku po aplikování filtru. Nabízí možnost měnit jednotlivé parametry Gaborovy funkce, popsané v rámci sekce 3.4.3, a vidět tak okamžitou odezvu filtru. Aplikace implementuje výpočet jádra Gaborova filtru pomocí funkce `getGaborKernel()` knihovny *OpenCV* i výpočet vlastní. Ukázka vizualizace jader Gaborova filtru a jeho aplikace na vstupní snímek je zobrazena na obrázku 4.4.



Obrázek 4.4: Ukázka různých druhů Gaborových filtrů a výsledku jejich aplikace na vstupní snímek (číslice 5). Zleva $\theta = 0^\circ$, $\theta = 45^\circ$, $\theta = 90^\circ$, $\theta = 135^\circ$. Pro všechny filtry je velikost jádra zvolena na hodnotu 21, $\sigma = 4.0/21.0$, $\lambda = 10.0$, $\gamma = 0.5$, $\psi = 180^\circ$.

4.3 Detekce vozidla ze statické kamery

Detekce vozidla ze statické kamery je založena na metodě odečítání pozadí, představené v sekci 3.2.1. Pro vytvoření modelu popředí a pozadí byla použita implementace třídy `BackgroundSubtractorMOG2` z části video analýzy knihovny *OpenCV*, která implementuje algoritmus segmentace pozadí/popředí založený na směsici Gaussovských funkcí.

Byly použity dvě instance této třídy s obdobným nastavením, které napomáhají určení polohy výskytu vozidla v obraze. Obě instance jsou tvořeny třemi Gaussovými funkcemi a používají defaultní hodnotu prahu Mahalanobisovi vzdálenosti. Rozdíl nastává při nastavování prahu modelu pozadí. První instance má tuto hodnotu nastavenou na 0.5 a slouží k segmentaci pohybujících se objektů od statického pozadí. Druhá instance má tuto hodnotu nastavenou na 0.9 a má zvolenou délku historie na 2 snímky. Tato instance slouží k redukci stínů detekovaných objektů.

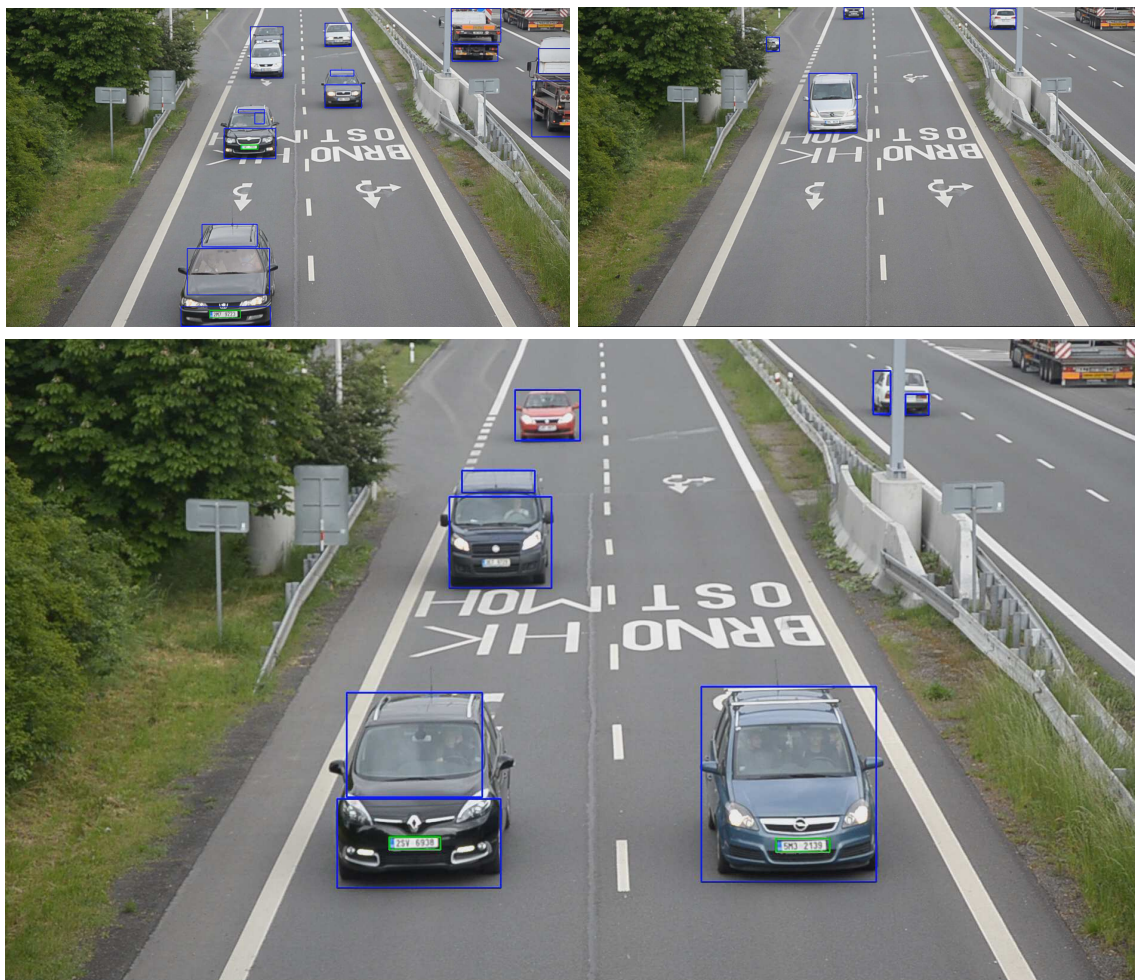
Pro získání výsledné polohy vozidla jsou od sebe odečteny získané masky popředí obou instancí a jsou použity morfologické operace pro uzavření daných oblastí a odstranění okolního šumu. Dále je na získanou masku popředí aplikována funkce pro vytvoření konvexních obálek oblastí popředí, které jsou dále zpracovávány. Tyto konvexní obálky určují ohraničení jednotlivých částí obrazu klasifikovaných jako popředí. Jelikož nemusí být získané oblasti zcela uzavřeny, může pro jedno vozidlo vzniknout větší počet ohraničení. Z toho důvodu je provedena redukce počtu ohraničení, které se vzájemně překrývají a je ponecháno jenom největší ohraničení oblasti.

Vyhodnocení detekce vozidel

Detekci vozidel není lehké vyhodnotit, jelikož navržený princip detekce pomocí metody odečítání pozadí detekuje veškerý pohyb v obraze. Jediným možným kritériem je tedy vyhodnocení přesnosti výpočtu ohraničení jednotlivých objektů tak, aby při detekci nedošlo k vynechání žádného vozidla a bylo na něm možné alespoň v jednom případě detekovat registrační značku (za předpokladu dostatečného rozlišení detekovaných vozidel, a tedy i registrační značky). Do vyhodnocení tedy započítáváme jenom vozidla jedoucí směrem ke kameře, na kterých je možnost detekovat registrační značku pro následné rozpoznání.

Během testování detektoru vozidel na testovací datové sadě byla detekována všechna vozidla v obraze, kdy pro každé vozidlo byla detekována registrační značka minimálně v jednom snímku. Úspěšnost detekce vozidel je tedy 100%, při necelých 10% *false alarm*, i přes jistou nestabilitu výpočtu ohraničení vozidel v některých případech. Vyšší počet *false alarm* nám ale v této fázi nevádí, jelikož se jedná pouze o redukci oblastí zájmů pro detekci registračních značek.

Provedeme-li vyhodnocení detekce vozidel, resp. výpočtu jejich ohraničení, po jednotlivých snímcích, s přihlédnutím potřeby detekce oblasti s lokalizací registrační značky, dostáváme úspěšnost detekce $\approx 92\%$.



Obrázek 4.5: Ukázka detekce vozidel. Modré pravoúhelníky označují vypočítané ohraničení pohybujících se objektu. Zelené obdélíky potom detekovanou pozici registrační značky pomocí kaskády klasifikátorů.

4.4 Detekce registrační značky vozidla

Během vývoje vznikly dvě verze detekce registrační značky vozidla. První z nich detekuje registrační značku pomocí vyhledávání klíčových bodů v obraze, která je vyhodnocována na sadě statických snímků zobrazených na obrázku 4.6. Bližší popis této metody je uveden v sekci 4.4.1.

Druhá verze je založena na detekci registrační značky s použitím kaskády klasifikátorů. Tato verze byla použita i do jednoduchého systému analýzy dopravy, který je uveden v sekci 4.7 a byla implementována pro detekci registračních značek ze záběrů simulující záběry dopravní kamery, na kterých byla také vyhodnocena.

4.4.1 Detekce RZ pomocí klíčových bodů

Detekce registrační značky pomocí vyhledávání klíčových bodů v obraze, včetně použití algoritmu, byla popsána v sekci 3.3.1. Pro implementaci byly použity funkce z částí knihovny *OpenCV* `feature2d` a `imgproc`. Pro pre-processing snímků je využito funkcí `resize()`, `gaussianBlur()` či `equalizeHist()`. Pro získání klíčových bodů je využito funkcí `cornerHarris()` a `canny()`. Lokalizace oblasti registrační značky je potom provedena pomocí funkcí `threshold()`, morfologických operací `erode()` a `dilate()`, následované funkcemi `watershed()` a `floodFill()`. Posledním krokem je získání minimálního ohraničení pomocí pixelů masky s hodnotou 255, které reprezentují vyhledaný objekt.

4.4.2 Detekce RZ s použitím kaskády klasifikátorů

Detekce registrační značky s použitím kaskády klasifikátorů, popsaná v sekci 3.3.2, má na svém vstupu množinu ohraničení oblastí výskytu vozidel. Pomocí těchto oblastí je redukována oblast zájmu (*Region Of Interest*), na kterou je následně aplikován pre-processing (de-saturace, blur, ekvalizace histogramu) a spuštěn algoritmus detekce registrační značky. Vlastní detekce je implementována pomocí instance třídy `CascadeClassifier` a její metody `detectMultiScale`. Této metodě je nastaven minimální počet sousedů každé kandidátní oblasti na 3, minimální detekované oblasti na 10×2 pixely a maximální velikost je potom nastavena na velikost vstupního snímku. Výstupem detektoru registračních značek je množina ohraničení všech možných výskytů registračních značek v obraze, která je vstupem modulu rozpoznání registrační značky vozidla.

Před použitím kaskády je potřeba ji natrénovat na vyhledávání požadovaných objektů. To se provádí pomocí aplikací `opencv_createsamples` a `opencv_traincascade`, které jsou součástí knihovny *OpenCV*. První jmenovaná aplikace generuje trénovací i testovací množinu dat ve formátu specifickém pro aplikaci `opencv_traincascade`. Popis jednotlivých parametrů a použití aplikací je k dispozici zde [23].

4.4.3 Vyhodnocení detekce RZ

Detekce registračních značek byla vyhodnocena více způsoby. První vyhodnocení probíhalo na množině dat statických snímků a byla porovnávána úspěšnost detekce obou metod. Detekce registrační značky s použitím kaskády klasifikátorů byla poté vyhodnocena i na videosnímčích reprezentujících záběry dopravní kamery.

Statické snímky

Porovnání detekce registrační značky na statických snímčích probíhalo na několika testovacích datových sadách. Rozložení vzorků v jednotlivých datových sadách je zobrazeno v tabulce 4.1. Testovací data byla pořízena mnou, data označená jako *reálná* jsou pořízena lidmi bez znalosti tohoto systému detekce. Ukázka snímků z datové sady kombinující testo-

vací a reálnou datovou sadu je k dispozici na obrázku 4.6. Výsledky úspěšnosti na datových sadách je potom zobrazeno v tabulce 4.2.



Obrázek 4.6: Ukázka vstupního datasetu pro detekci registrační značky vozidla pomocí vyhledávání klíčových bodů i kaskády klasifikátorů.

Tabulka 4.1: Popis rozložení testovací datové sady snímků. Testovací data byla pořízena během implementace metod detekce RZ. Reálná data byla pořízena lidmi bez znalosti omezení těchto detekčních systémů.

Vstupní množina	Počet snímků	Původ dat
Sada 1	104	Testovací
Sada 2	52	Testovací
Sada 3	106	Testovací
Sada 4	148	Reálná
Sada 5	78	Reálná

Tabulka 4.2: Porovnání metod detekce registrační značky vozidla.

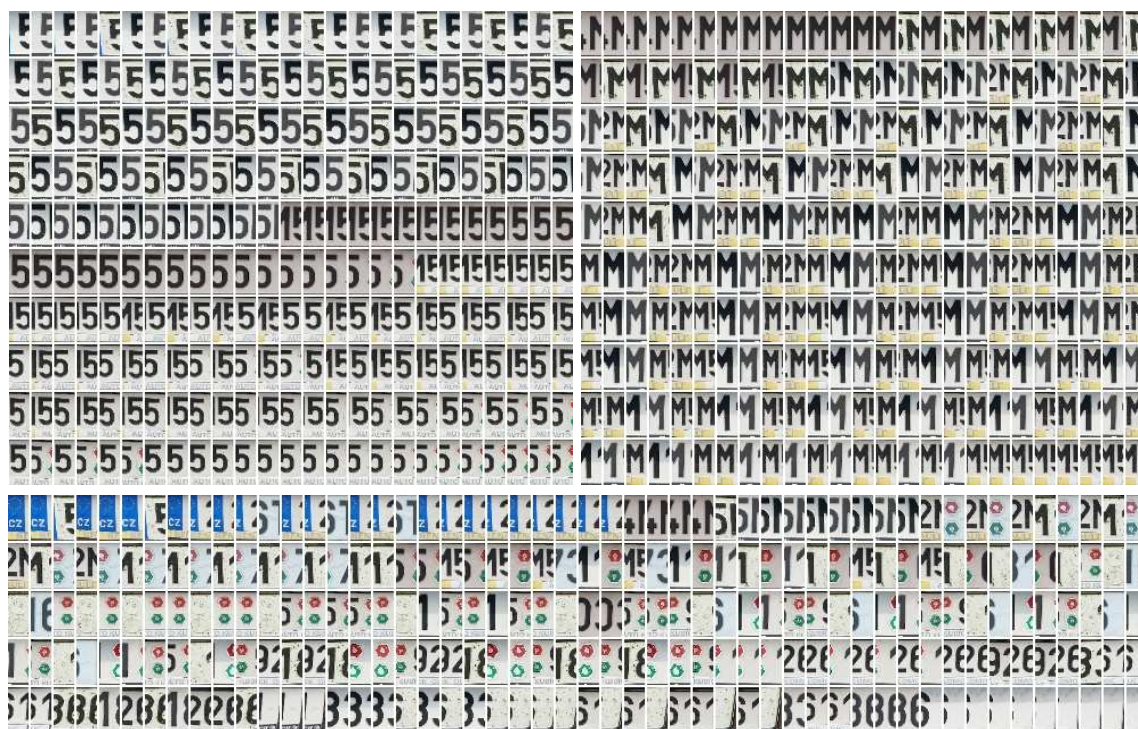
Metody detekce RZ vozidla		
Vstupní množina	Klíčové body	Kaskáda klasifikátorů
Sada 1	81.26%	90.55%
Sada 2	86.67%	94.17%
Sada 3	92.45%	96.34%
Sada 4	56.32%	78.97%
Sada 5	64.51%	84.38%
Průměr	76.24%	88.94%

4.4.4 Simulace dopravní kamery

Druhou fází vyhodnocení bylo testování detekce registrační značky pomocí kaskády klasifikátorů na videosnímecích s pohledem na pozemní komunikace. Provedeme-li na této datové sadě vyhodnocení pro všechny detekované oblasti předpokládaného výskytu vozidla, dosáhl algoritmus detekce registračních značek úspěšnosti (hit rate) 81.72% a přesnosti (precision) 94.42%. Neúspěšná detekce byla v 18.82% případů. Alespoň jednou byla registrační značka detekována na všech projíždějících vozidlech ve 100% případů. Požadavkem systému sice je alespoň jednou detekovaná registrační značka na každém detekovaném vozidle, ale přesto požadujeme od fáze detekce registrační značky co nejvyšší úspěšnost.

4.5 Příprava a trénování neuronové sítě

Nedílnou součástí rozpoznávání registrační značky vozidla pomocí neuronové sítě jsou pomocné aplikace pro přípravu datové sady pro trénování neuronové sítě a její vlastní trénování. Jedná se o aplikace `prepare_dataset` a `train_dataset`. Jak už název aplikací napovídá, první z nich se stará o přípravu dat pro trénování a testování neuronové sítě, druhá pak o vlastní trénování a testování neuronové sítě. Ukázka datové sady použité pro trénování neuronové sítě je zobrazena na obrázku 4.7.



Obrázek 4.7: Ukázka datové sady pro trénování neuronové sítě. (Vlevo) Ukázka trénovací množiny pro znak 5. (Vpravo) Ukázka trénovací množiny znaku M. (Dole) Ukázka datové množiny, která není klasifikována jako znak registrační značky.

4.5.1 Příprava dat pro trénování a testování

Aplikace `prepare_dataset` převádí množinu vstupních dat na sekvenci čísel, která je následně uložena do textového souboru a slouží pro trénování/testování neuronové sítě. Pro získání hodnot v podobném rozsahu pro různé vstupy je třeba provést normalizaci vstupního obrazu. Na výběr bylo více možností jak obraz normalizovat. Nabízela se možnost provést normalizaci například pomocí vzorce

$$I_{norm} = I_{orig}/255, \quad (4.1)$$

ale tato normalizace provede pouze úpravu rozsahu hodnot do rozsahu 0 – 1. Jako nejvhodnější byla nakonec zvolena normalizace pomocí vzorce

$$I_{norm} = (I_{orig} - \mu)/\sigma, \quad (4.2)$$

kde μ je střední hodnota a σ je směrodatná odchylka. Pomocí této normalizace dostaneme rovnoměrné rozložení hodnot okolo počátku osy souřadného systému.

Dále bylo potřeba vybrat vhodné dodatečné příznaky pro trénování neuronové sítě. Byly provedeny mnohé experimenty s vytvářením dodatečných příznaků vstupních dat, ze kterých byla nakonec zvolena verze používající čtyři Gaborovy filtry s natočením normály $0^\circ, 45^\circ, 90^\circ$ a 135° . Celkem je tedy použito 5 rozdílných příznaků pro jeden vstupní snímek, kdy ke čtyřem Gaborovým filtrům náleží ještě příznak intenzity vstupního snímku. Před přípravou datové sady je potřeba všechny vytvořené příznaky normalizovat dle rovnice (4.2).

4.5.2 Vytváření neuronové sítě

Aplikace `train_dataset` poté vytváří vlastní neuronovou síť a trénuje ji. Neuronová síť použitá v rámci této práce je standardní síť dostupná v knihovně *OpenCV*, v části *Machine Learning*. Jedná se o plně dopřednou síť ANN (*Artificial Neural Network*), která pro své trénování používá error back-propagation algoritmus a aktivační funkci typu symetrické sygmoidy.

Výsledná neuronová síť se skládá ze tří vrstev. První vstupní vrstva obsahuje 900 vstupních neuronů. Jedná se o pět vstupních příznaků velikosti 9×20 pixelů. Poslední vrstva je vrstvou výstupní a obsahuje 37 neuronů – jeden neuron pro každý výstupní znak. Výstupní množina se sestává ze znaků $\{0-9, A-Z, -\}$, kde $\{-\}$ vyjadřuje, že se nejedná o znak. Velikost prostřední skryté vrstvy je potom spočítána jako průměr součtu počtu vstupních a výstupních neuronů. Parametry trénování byly stanoveny na maximum 1000 iterací při rozlišení odlišnosti znaků 0.0000001 epsilon.

4.5.3 Experimenty

Podstatnou částí při vytváření neuronové sítě bylo testování různých kombinací jejího nastavení. Bylo potřeba najít vhodné rozměry neuronové sítě a parametry pro její trénování. Velikost vstupního obrazu byla stanovena na 9×20 pixelů, s přihlédnutím na vypočítaný poměr stran znaků na registrační značce. Počet vstupních neuronů je tedy ovlivněn počtem použitých příznaků. Konečná velikost bude odpovídat počtu použitých příznaků násobených velikostí vstupního obrazu. Počet neuronů výstupní vrstvy je dán počtem predikovaných tříd. V našem případě se jedná o 37 neuronů.

Jediným zbývajícím možným nastavením je velikost a případně počet skrytých vrstev. Experimentálně byla stanovena nejvhodnější velikost skryté vrstvy jako průměr součtu vstupních a výstupních neuronů. Také bylo experimentálně ověřeno, že jedna skrytá vrstva je dostačující. Při navýšení počtu skrytých vrstev na dvě nedošlo k výraznému rozdílu výsledků rozpoznání oproti vrstvě jedné. Větší počet vrstev měl za následek přetrénování neuronové sítě, kdy úspěšnost rozpoznání znaků byla mizivá, přičemž čas potřebný k trénování sítě neúměrně naroste. Například pro 5 skrytých vrstev nedosahovala úspěšnost rozpoznání ani 3%, při době trénování ≈ 6 hodin.

Testování vlivu změny parametrů neuronové sítě

Po stanovení parametrů velikosti neuronové sítě bylo potřeba ověřit vliv změny parametrů trénování na její výsledky. Tabulka 4.3 vizualizuje vliv změny parametrů trénování neuronové sítě na výsledek rozpoznání. Sloupec *Příznaky* reprezentuje počet použitých příznaků pro trénování. Hodnota 1 odpovídá použití příznaku intenzity. Hodnota 5 potom odpovídá použití příznaku intenzity a výsledku aplikace čtyř Gaborových filtrů s různou orientací jádra. Sloupec *Epsilon* udává nastavení míry odlišnosti pro rozpoznání znaků. Sloupec *Max. iterací* informuje o hodnotě limitu počtu iterací. Následující tři sloupce se již vztahují k výsledkům trénování neuronové sítě. Sloupec *Počet iterací* udává hodnotu potřebného počtu iterací k natrénování neuronové sítě. Sloupec *Úspěšnost* udává procentuální úspěšnost rozpoznání na testovací datové sadě. Poslední sloupec potom udává čas potřebný k trénování a otestování neuronové sítě.

Výsledná úspěšnost rozpoznání znaků registrační značky pomocí neuronové sítě je ovlivněna také počtem výskytu znaků v trénovací datové sadě. Trénovací datová sada obsahuje 2884 snímků znaků, některé znaky jsou však v trénovací datové sadě zastoupeny jenom minimálně, jelikož je bylo obtížně získat během pořizování datové sady. Při nasazení systému bude potřeba datovou sadu rozšířit, aby se dostatečným počtem vzorků pokryly všechny potřebné znaky. Velice zajímavým zjištěním je, že přidání příznaků o hranové informaci vede ke zhoršení úspěšnosti rozpoznání.

Tabulka 4.3: Ukázka testování vlivu změny parametrů trénování neuronové sítě na výsledek rozpoznání.

Příznaky	Epsilon	Max. iterací	Počet iterací	Úspěšnost %	Čas
1	1×10^{-2}	1000	13	68.9474	0m2.752s
1	1×10^{-3}	1000	19	69.0789	0m3.930s
1	1×10^{-4}	1000	62	70.3947	0m12.016s
1	1×10^{-5}	1000	258	72.6316	0m48.075s
1	1×10^{-6}	1000	428	73.0263	1m20.310s
1	1×10^{-7}	1000	428	73.0263	1m20.030s
1	1×10^{-8}	1000	778	72.8947	2m28.418s
1	1×10^{-9}	1000	1000	71.9737	3m7.205s
1	1×10^{-9}	10000	2291	70.7895	7m1.916s
5	1×10^{-2}	1000	15	55.6579	1m15.604s
5	1×10^{-3}	1000	29	58.0263	2m23.614s
5	1×10^{-4}	1000	54	57.2368	4m10.332s
5	1×10^{-5}	1000	312	55.1316	24m40.185s
5	1×10^{-6}	1000	1000	55.9211	78m26.191s
5	1×10^{-6}	10000	2293	63.4211	179m19.494s
5	1×10^{-7}	10000	2780	64.4737	229m24.992s

4.5.4 Vyhodnocení úspěšnosti rozpoznání znaků neuronovou sítí

Vyhodnocení úspěšnosti rozpoznání znaků neuronovou sítí je součástí trénování neuronové sítě. Vyhodnocení probíhá na testovací datové sadě, která byla vytvářena společně s trénovací datovou sadou, avšak výběrem jiných vstupních snímků. Testovací datová sada obsahuje celkem 760 testovacích snímků. Výsledky vyhodnocení jsou vizualizovány pomocí kontingenční tabulky (anglicky také známé jako *confusion matrix*). Kontingenční tabulka sumarizuje výsledky testování algoritmů pro pozdější použití. Řádky tabulky reprezentují skutečné třídy testovacích dat, zatímco sloupce tabulky nám poskytují třídy predikované. Snahou je dosáhnout nejvyššího možného počtu hodnot na hlavní diagonále. Ukázky kontingenčních tabulek a kompletní výsledky experimentů neuronových sítí jsou k dispozici v příloze. Kontingenční tabulky pro nejúspěšnějších experimentů výše pro jeden a pět příznaků jsou k nahlédnutí v příloze. Kompletní kontingenční tabulky všech experimentů jsou součástí elektronické přílohy.

4.6 Rozpoznání detekované registrační značky

Pro modul rozpoznání znaků detekované registrační značky vozidla byly implementovány a testovány metody template matching a nově navržená metoda rozpoznání registrační značky pomocí neuronové sítě a *sliding window* skenování obrazu. Principy těchto metod jsou vysvětleny v sekci 3.4. Implementační detaily a vyhodnocení jsou uvedeny níže.

4.6.1 Template matching

Aplikace demonstrující metodu rozpoznávání znaků pomocí *template matching* zároveň prezentuje metodu detekce registrační značky pomocí vyhledávání klíčových bodů. Algoritmus zde pracuje v několika fázích. První fází je vlastní detekce registrační značky, která byla představena v sekci 3.3.1, a její specifikací a vyhodnocením se zabývala sekce 4.4.1. V dalším kroku dochází k separaci jednotlivých znaků registrační značky a samotnému rozpoznávání. Oba tyto kroky byly přiblíženy v sekci 3.4.1. Princip této metody je prezentován na obrázku 4.8, ukázky výstupů jsou pak zobrazeny na obrázku 4.9.

Samotné rozpoznání jednoho znaku registrační značky probíhá v několika etapách. Vstupem rozpoznávače je separovaný znak registrační značky, který chceme rozpoznat. Na vstupní snímek je spuštěn pre-processing rozpoznání, kdy je vstupní znak očištěn od případného šumu a jeho okolí je redukováno na minimální ohraničení znaku. Takto připravený a očištěný znak je následně rozšířen o stejnou oblast, o jakou byla rozšířena množina šablon jednotlivých znaků. Tímto principem je dosaženo zarovnání znaků a je možné přistoupit k samotnému rozpoznání. Rozpoznání potom probíhá použitím funkce `bitwise_xor()` mezi znakem určeným k rozpoznání a množinou vstupních šablon. Jako nejvhodnější se potom vybírá znak, který má minimální chybu překryvu.



Obrázek 4.8: Ukázka principu rozpoznání registrační značky vozidla pomocí metody *template matching*. (Vlevo) Vstupní snímek pro detekci a rozpoznání. (Uprostřed) Detekované znaky uvnitř detekované registrační značky. (Vpravo) Výsledek rozpoznání registrační značky společně s detekovanou oblastí registrační značky.



Obrázek 4.9: Ukázka výstupu detekce registrační značky pomocí vyhledávání klíčových bodů a její rozpoznání s využitím metody *template matching*. Na obrázcích 2 a 9 je vidět selhání detektoru registračních značek. Obrázek 3 potom zobrazuje chybné rozpoznání písmena Z, kdy byla jako více odpovídající zvolena číslice 2.

4.6.2 Rozpoznání RZ pomocí neuronové sítě a sliding window skenování

Vstupem modulu rozpoznání registrační značky pomocí neuronové sítě je snímek registrační značky vozidla, který je získán pomocí separace oblasti detekované registrační značky ze vstupního obrazu. Nad tímto obrazem je provedena změna rozlišení na výšku 20 pixelů se zachováním poměru stran obrazu. Takto upravená registrační značka je skenována pomocí *sliding window* a pro každou pozici je vytvořen řez registrační značky stejně jak je popsáno v sekci 4.2.3.

Nad těmito řezy je následně provedeno vytvoření příznaků obrazu a příprava dat pro rozpoznání. Samotné rozpoznání potom probíhá pomocí instance třídy CvANN_MLP a její metody `predict()`, která pro každou třídu vypočte ohodnocení dodaných dat. Výsledný znak je vybrán jako nejlépe ohodnocená třída pro daná vstupní data. Tímto způsobem je pro každou registrační značku vygenerován řetězec, který nejvíce odpovídá dodaným vstupním datům. Kromě vygenerovaného řetězce jsou k dispozici ještě ohodnocení získaných znaků pro každou pozici registrační značky. Tyto informace budou důležité pro fázi finalizace a validace rozpoznané registrační značky, která prozatím nebyla implementována.

4.6.3 Vyhodnocení rozpoznávání registračních značek vozidel

Rozpoznávání registračních značek vozidla pomocí metody *Template matching* bylo vyhodnoceno na datové sadě statických snímků popsané v sekci 4.4.3 v tabulce 4.1. Dosažená procentuální úspěšnost pro rozpoznání celé registrační značky, tj. pro korektní rozpoznání všech 7 znaků registrační značky, je uvedena v tabulce 4.4.






Tabulka 4.4: Porovnání úspěšnosti rozpoznání registrační značky vozidla na statických snímcích.

Metoda <i>Template matching</i>	
Vstupní množina	Úspěšnost %
Sada 1	71.39%
Sada 2	86.46%
Sada 3	79.59%
Sada 4	22.50%
Sada 5	42.81%
Průměr	60.55%

Na stejné datové sadě probíhalo také testování nově navržené metody rozpoznání registrační značky postupným skenováním užitím principu *sliding window* a klasifikací pomocí neuronové sítě. Ukázka výstupu je zobrazena na obrázku 4.10.

4.7 Jednoduchý systém analýzy dopravy

Posledním krokem implementace bylo vytvoření jednoduchého systému analýzy dopravy, který by implementoval některé výše popsané metody a demonstroval by jejich použitelnost

-  44444444_MMMMMMMMM_55555555_____00000000_333333333_666666666_11111111____
-  22222222_MMMMMMMMM_55555555_____999999999_222222_666666_11111111____
-  306665553336MMMMMM5116_61111111_____31111111_868666688_3333333_866888886_1
-  22222222_MMMMMMMMM_55555555_____999999999_222222_666666_11111111____
-  0866655553_36MMMMMMMM1M1M33333333306666_____3333333336_666666666663333333336_2_44444662_____4

Obrázek 4.10: Ukázka řetězců získaných rozpoznáním registrační značky pomocí neuronové sítě.

na záběrech pozemních komunikací simulujících pohled dopravní kamery. Pro implementaci tohoto systému bylo využito detekce vozidel pomocí metody odečítání pozadí, detekce registračních značek pomocí kaskády klasifikátorů a rozpoznání registrační značky použitím neuronové sítě během skenování obrazu pomocí *sliding window*. Posledním chybějícím krokem k dotvoření systému analýzy dopravy je fáze finalizace a validace registrační značky. Po doplnění posledního kroku chybí už jen krůček k samotné re-identifikaci vozidla v obraze.

4.7.1 Propojení jednotlivých fází systému

Prvním krokem je detekce vozidla v obraze, jejímž výsledkem je množina ohraničení pohybujících se objektů v obraze. Na takto redukovanou sadu vstupních dat navazuje detekce registrační značky vozidla, která vyhledá výskyt registrační značky v detekované oblasti možného výskytu vozidla. Na této získané oblasti je potom aplikováno vlastní rozpoznání registrační značky, které nám vygeneruje řetězec odpovídající vstupním datům s patřičným ohodnocením.

Kapitola 5

Závěr

Cílem této práce bylo navrhnout a otestovat algoritmy detekce a rozpoznání registrační značky vozidla vhodné pro systém re-identifikace vozidla. Součástí bylo také vytvořit modul detekce vozidla, a také vytvořit jednoduchý systém analýzy dopravy prezentující použitelnost algoritmů pro zpracování dat dopravní kamery. Navržený systém je rozdělen do několika jednotlivých fází zpracování vstupních dat. Konkrétně se jedná o fázi detekce vozidla, fázi detekce registrační značky vozidla a fázi rozpoznání registrační značky vozidla. Pro jednotlivé fáze systému byly navrženy, implementovány a testovány různé metody a přístupy zpracování vstupních dat. Jednotlivé fáze, metody a přístupy byly následně samostatně vyhodnoceny, byly zjištěny jejich klady a zápory a vhodnost jejich použití pro danou úlohu. Pro testování a vyhodnocení těchto algoritmů bylo posbíráno a pořízeno přes 10200 minut záznamu, 488 statických snímků vozidel a 4650 vzorků registračních značek vozidel.

Ve fázi detekce vozidel je použita metoda odečítání pozadí, která dosahuje úspěšnosti $\approx 92\%$ při vyhodnocení po snímcích. Fáze detekce vozidel je v této práci uvedena pouze pro úplnost. Při reálném nasazení systému by měla být nahrazena výstupem systému analýzy dopravy představeném ve článku [5].

V oblasti detekce registrační značky vozidla byly na statických snímcích testovány metody detekce pomocí vyhledávání klíčových bodů a kaskády klasifikátorů. První uvedená metoda dosáhla průměrné úspěšnosti detekce 76.24%, zatímco druhá metoda dosáhla úspěšnosti 88.94%. Metoda detekce registrační značky vozidla pomocí kaskády klasifikátorů byla také testována na záběrech dopravních komunikací simulujících pohled dopravní kamery. Na této datové sadě dosáhla metoda úspěšnosti (hit rate) 81.72% a přesnosti detekce (precision) 94.42%.

Rozpoznání registrační značky vozidla pro datovou sadu statických snímků bylo provedeno pro metodu *Template matching*. Na této datové sadě bylo dosaženo průměrné úspěšnosti rozpoznání celé registrační značky, tj. všech sedmi znaků korektně, ve výši 60.55%.

Pro fázi rozpoznání registrační značky ze záběrů simulujících pohled dopravní kamery byla navržena zcela nová metoda rozpoznání registrační značky vozidla pomocí neuronové sítě a jejího skenování pomocí *sliding window*. Tato metoda produkuje řetězce specifické pro

registrační značky. Pro finální rozpoznání registrační značky je potřeba najít vhodný algoritmus pro výběr kandidátních znaků s největší pravděpodobností výskytu. Tato finalizace rozpoznání registrační značky vozidla je zde uvedena pouze v návrhu a nebyla implementována. Z toho důvodu nebylo možné provést vyhodnocení rozpoznání registrační značky vozidla jako celku, ale je zde uvedeno vyhodnocení rozpoznání jednotlivých znaků registrační značky na testovací množině dat. Na této testovací množině dosahuje natrénovaná neuronová síť úspěšnosti rozpoznání 64.47% při použití pěti příznaků pro jeden vstupní snímek. Tato úspěšnost je ale ovlivněna nedostatečným množstvím vzorků pro některé znaky registrační značky.

Dalším postupem v této práci bude vytvoření poslední kroku rozpoznání registrační značky vozidla, kterým je finalizace a validace řetězců vytvořených skenováním registrační značky vozidla a jejím rozpoznáváním pomocí neuronové sítě. Tato fáze je klíčová pro dokončení celého systému. Samotná re-identifikace vozidla již potom nebude velký problém.

Dále je nutné pokračovat v rozšiřování datových sad a vylepšování jednotlivých fází systému. Velice důležitou roli bude hrát zejména rozšíření datové sady znaků pro trénování neuronové sítě, aby systém rozpoznání dosahoval kvalitních výsledky a byl stabilní. Stabilitou je zde myšleno stabilní výsledek rozpoznání stejné registrační značky při pohybu vozidla v různých snímcích.

Po dokončení zmíněných fází bude nutné systém otestovat v kombinaci s výstupy systému analýzy dopravy ze článku [5]. Při získání znalostí o obecném čtyřúhelníku ohraničujícím vozidlo bude možné provést úpravu systému pro lepší výsledky rozpoznávání registrační značky vozidla bez nutnosti transformací obrazu.

Literatura

- [1] Aggarwal, A. K.; Aggarwal, A. K.: Vehicle Registration Plate Recognition System Based on Edge Transition by Row and Column Profile on Still Image. In *International Journal of Engineering Research and Technology (IJERT)*, ročník 2, 2013.
- [2] Alonso, D.; Salgado, L.; Nieto, M.: Robust vehicle detection through multidimensional classification for on board video based systems. In *Proc. IEEE ICIP*, ročník 4, září 2007, s. 321–324.
- [3] Dollár, P.; Tu, Z.; Perona, P.; aj.: Integral Channel Features. In *BMVC*, 2009.
- [4] Dubská, M.; Herout, A.; Juránek, R.; aj.: Fully Automatic Roadside Camera Calibration for Traffic Surveillance. *IEEE Transactions on Intelligent Transportation Systems*, ročník 2014, č. 1, 2014: s. 1–10, ISSN 1524-9050.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=10645
- [5] Dubská, M.; Sochor, J.; Herout, A.: Automatic Camera Calibration for Traffic Understanding. In *Proceedings of BMVC 2014*, The British Machine Vision Association and Society for Pattern Recognition, 2014, ISBN 1-901725-49-9, s. 1–10.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=10682
- [6] Euroskop.cz: Řízení vozidla v EU [online]. 2005 [cit. 2015-01-10].
URL <https://www.euroskop.cz/597/sekce/rizeni-vozidla-v-eu/>
- [7] Feureil, D.: Autorské písmo SPZ 2004 Regular [online]. 2006 [cit. 2015-05-22].
URL <http://www.feudal.cz/spz/files/nahled01.gif>
- [8] Freund, Y.; Schapire, R.: A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, ročník 14, č. 771-780, 1999: str. 1612.
- [9] Friedman, J.; Hastie, T.; Tibshirani, R.; aj.: Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, ročník 28, č. 2, 2000: s. 337–407.
- [10] Gao, H.; Sun, W.; Liu, X.; aj.: Research and Implementation of a License Plate Recognition Algorithm Based on Hierarchical Classification. *Journal of Computer and Communications*, ročník 2, 2014: str. 25.

- [11] Hájek, M.: České registrační značky [online]. 2012 [cit. 2015-05-22].
URL http://www.3260.cz/typy_poznavacich_znacek.htm
- [12] Horprasert, T.; Harwood, D.; Davis, L. S.: A statistical approach for real-time robust background subtraction and shadow detection. In *Proc. IEEE ICCV*, ročník 99, 1999, s. 1–19.
- [13] Jazayeri, A.; Cai, H.; Zheng, J. Y.; aj.: Vehicle Detection and Tracking in Car Video Based on Motion Model. In *IEEE Transactions on intelligent transportation systems*, ročník 12, červen 2011, s. 583–595.
- [14] Jiangwei, C.; Lisheng, J.; Lie, G.; aj.: Study on method of detecting preceding vehicle based on monocular camera. In *Proc. IEEE Intelligent Vehicles*, 2004, s. 750–755.
- [15] Kato, J.; Watanabe, T.; Joga, S.; aj.: An HMM-based segmentation method for traffic monitoring movies. In *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 24, září 2002, s. 1291–1296.
- [16] Laganière, R.: *OpenCV 2*. Packt Publishing Ltd., 2011, iISBN 978-1-849513-24-1.
- [17] Lee, R.-C.; Hung, K.-C.: An Automatic Vehicle License Plate Recognition Based on Prewitt Edge Detection and Artificial Neural Network. *International Journal Of Scientific Research And Education*, ročník 1, č. 05, 2014.
- [18] Nixon, M. S.; Aguado, A. S.: *Feature Extraction & Image Processing for Computer Vision*.
- [19] Nixon, M. S.; Aguado, A. S.: *Feature Extraction & Image Processing for Computer Vision*, kapitola 8.3. Elsevier Ltd., třetí vydání, 2012, ISBN 978-0-12-396549-3, s. 411–417.
- [20] Nixon, M. S.; Aguado, A. S.: *Feature Extraction & Image Processing for Computer Vision*, kapitola 5.5. Elsevier Ltd., třetí vydání, 2012, ISBN 978-0-12-396549-3, s. 243–291.
- [21] Nixon, M. S.; Aguado, A. S.: *Feature Extraction & Image Processing for Computer Vision*, kapitola 5.3. Elsevier Ltd., třetí vydání, 2012, ISBN 978-0-12-396549-3, s. 222–234.
- [22] OpenCV: How to Use Background Subtraction Methods [online]. 2014-12-30 [cit. 2015-01-10].
URL http://docs.opencv.org/trunk/doc/tutorials/video/background_subtraction/background_subtraction.html
- [23] OpenCV: Cascade Classifier Training [online]. 2014-12-30 [cit. 2015-05-22].
URL http://docs.opencv.org/doc/user_guide/ug_traincascade.html

- [24] OpenCV: Neural Networks [online]. 2014-12-30 [cit. 2015-05-22].
URL http://docs.opencv.org/doc/user_guide/ug_traincascade.html
- [25] Špaňhel, J.: Detekce automobilu v obraze, bakalářská práce, Brno, FIT VUT v Brně, 2013.
- [26] Ředitelství silnic a dálnic ČR: Mobilní radar zachytí i registrační značku [online]. 2014-09-16 [cit. 2015-01-10].
URL <http://www.rsd.cz/doc/Informacni-servis/mobilni-radar-zachyti-i-registracni-znacku>
- [27] Schneiderman, H.; Kanade, T.: A statistical method for 3D object detection applied to faces and cars. In *Proc. IEEE CVPR*, 2000, s. 746–751.
- [28] Sharma, J.; Mishra, A.; Saxena, K.; aj.: A hybrid technique for License Plate Recognition based on feature selection of wavelet transform and artificial neural network. In *Optimization, Reliability, and Information Technology (ICROIT), 2014 International Conference on*, IEEE, 2014, s. 347–352.
- [29] Song, M. K.; Sarker, M. M. K.: Modeling and Implementing Two-Stage AdaBoost for Real-Time Vehicle License Plate Detection. *Journal of Applied Mathematics*, ročník 2014, 2014.
URL <http://dx.doi.org/10.1155/2014/697658>
- [30] Soumya, K. R.; Babu, A.; Therattil, L.: License Plate Detection and Character Recognition Using Contour Analysis. *International Journal*, ročník 3, č. 1, 2014, ISSN 2278-3091.
- [31] Tan, C.; Cao, J.: An Algorithm for License Plate Location Based on Color and Texture. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on*, ročník 2, 2013, s. 356–359,
doi:10.1109/IHMSC.2013.232.
- [32] Wang, R.; Sang, N.; Huang, R.; aj.: License plate detection using gradient information and cascade detectors. *Optik-International Journal for Light and Electron Optics*, ročník 125, č. 1, 2014: s. 186–190.
- [33] Wang, X.-h.; Yu, J.-j.; Miao, Z.-h.; aj.: License plate recognition based on pulse coupled neural networks and template matching. In *Control Conference (CCC), 2014 33rd Chinese*, IEEE, 2014, s. 5086–5090.
- [34] Zhang, W.; Fang, X. Z.; Yang, X.: Moving vehicles segmentation based on Bayesian framework for Gaussian motion model. In *Pattern Recognition Letters*, ročník 27, 2006, s. 956–967.

- [35] ZhengHao, D.; FengXin: Research on License Plate Recognition Algorithm based on Support Vector Machine. In *Journal of multimedia*, ročník 9, Academy Publisher, 2014, s. 253–260.
URL <http://ojs.academypublisher.com/index.php/jmm/article/viewFile/jmm0902253260/8451>
- [36] Zhu, Z.; Lu, H.; Hu, J.; aj.: Car detection based on multi-cues integration. In *Proc. 17th ICPR*, ročník 2, 2004, s. 699–702.

Příloha A

Obsah CD

- Aplikace
 - video2frames
 - frames2licensePlate
 - licensePlate2segments
 - OCR_NN
 - TemplateMatching_Example
 - NeuralNetwork_Example
 - vehicleReidentification
- Data
 - Vzorky RZ
 - Testovací videosekvence
 - Testovací statické snímky
- Technická zpráva
- Výstupy
 - Zpracované videosekvence
 - Kontingenční tabulky experimentů
- Prezentační video

Příloha B

Kontingenční tabulky experimentů testování neuronové sítě

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z				
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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	59	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	0	0	0	0	0	6	
2	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	0	0	0	0	0	0	0	1	0
3	0	8	31	36	0	1	1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	22	
4	1	3	0	0	0	0	1	2	0	3	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	
5	0	1	0	0	0	53	3	1	0	0	0	0	0	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	1	
6	0	8	2	0	0	1	47	0	0	2	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	3	
7	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	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	17	0	18	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	1	
9	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	0	0	0	0	0	0	0	1	
A	0	0	0	0	0	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	0	0	0	
B	0	0	0	0	0	0	0	0	0	0	6	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	
C	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	0	0	0	0	0	0	0	0	0	0	0	0	0	
D	0	0	0	0	0	0	0	0	0	0	0	0	10	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	
E	0	0	0	0	0	0	0	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	0	0	0	0	0	
F	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	0	0	0	0	0	0	0	0	0	0	
G	0	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	0	0	0	0	0	0	0	0	0	
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
I	0	0	0	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	0	0	0	0	0	0	0	
J	0	0	0	0	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	0	0	0	0	0	0	
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M	1	6	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	
N	0	0	0	0	0	0	0	0	0	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	0	
O	0	0	0	0	0	0	0	0	0	0	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	
P	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
Q	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	1	0	0	0	0	0	0	0	0	0	0	0	0		
R	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	5	0	0	0	0	0	0	0	0	0	0	0	0	
S	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	0	0	0	0	0	0	0	0	
T	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	11	0	0	0	0	0	0	0		
U	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	0	0	0	0			
V	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	3	0	0	0	0	0	0			
W	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	1	0	0	0	0	0			
X	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	0	0	0	1	0	0	0		
Y	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	0	0	0	0	1	0	0		
Z	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	0	0	0	0	0	0	2	0	
Σ	0	13	1	0	0	3	5	0	2	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	215		

Obrázek B.1: Kontingenční tabulka. Jeden vstupní příznak, $\epsilon = 1 \times 10^{-7}$, počet iterací 428, úspěšnost rozpoznání 73.0263%, čas trénování 1m20.030s.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		
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	1	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	53	2	0	0	0	0	0	0	0	1	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	8	
2	0	0	0	0	0	0	0	0	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	
3	3	15	11	35	0	2	2	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0	0	0	25		
4	1	4	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	1	0	0	0	0	0	0	18		
5	1	4	0	1	0	32	5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	4	0	0	0	0	1	0	0	0	0	0	0	0	0	11		
6	0	8	1	2	0	0	35	0	6	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	10		
7	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	0	0	0	0	0		
8	0	0	1	0	0	1	14	0	17	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	2		
9	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	1	0	0	0	0	0	0	0	0		
A	0	0	0	0	0	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		
B	0	0	0	0	0	0	0	0	0	0	6	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	
C	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	0	0	0	0	0	0	0	0	0	0	0	
D	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
E	0	0	0	0	0	0	0	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	0	0	0	
F	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	0	0	0	0	0	0	0	0	
G	0	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	0	0	0	0	0	0	0	
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
I	0	0	0	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	0	0	0	0	0	
J	0	0	0	0	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	0	0	0	0	
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M	0	19	0	0	0	7	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	3	0	0	0	0	0	0	13		
N	0	0	0	0	0	0	0	0	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	
O	0	0	0	0	0	0	0	0	0	0	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	
P	0	0	0	0	0	0	0	0	0	0	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	
Q	0	0	0	0	0	0	0	0	0	0	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	
R	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	5	0	0	0	0	0	0	0	0	0	0	
S	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	0	0	0	0	0	0	
T	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	11	0	0	0	0	0	0	0	0	0	
U	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	0	0	0	0	0	0	0	
V	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	3	0	0	0	0	0	0	0	0	0	
W	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	0	1	0	0	0	0	
X	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	0	0	1	0	0	0	
Y	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	0	0	0	1	0	0	0
Z	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	0	0	0	0	2	0	0
1	1	3	4	0	1	6	6	0	4	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	219	

Obrázek B.2: Kontingenční tabulka. Pět vstupních příznaků, $\epsilon = 1 \times 10^{-7}$, počet iterací 2780, úspěšnost rozpoznání 64.4737%, čas trénování 229m24.992s.