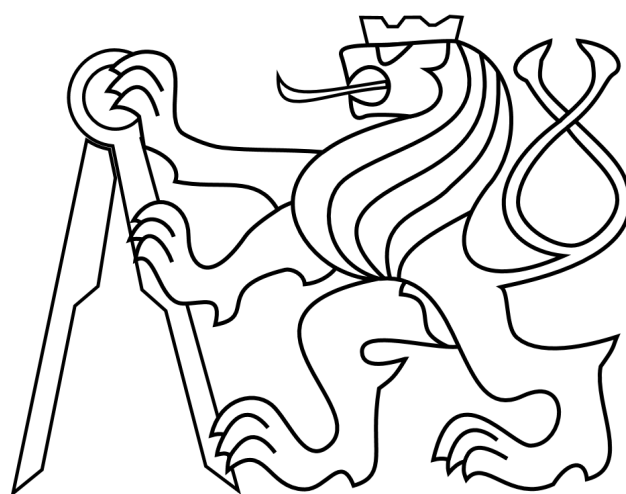


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra radioelektroniky

DIPLOMOVÁ PRÁCE



Viktoriia Lysenko

Účinné metody klasifikace barev vozidel

Efficient Methods for Vehicle Color Classification

Vedoucí práce: **Ing. Karel Fliegel**

PRAHA 2020

Poděkování

Ráda bych poděkovala Ing. Karlu Fliegelovi, Ph.D. za vedení mé práce, věcné připomínky a vstřícnost při konzultacích. Poděkování také patří firmě GoodVision za spolupráci a poskytnutí potřebných dat. V neposlední řadě patří obrovské poděkování mé rodině a přátelům za podporu v průběhu studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

Podpis.....

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Lysenko** Jméno: **Viktoriia** Osobní číslo: **434785**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**
Specializace: **Audiovizuální technika a zpracování signálů**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Účinné metody klasifikace barev vozidel

Název diplomové práce anglicky:

Efficient Methods for Vehicle Color Classification

Pokyny pro vypracování:

Podějte přehled účinných metod pro automatizovanou klasifikaci barev vozidel v dopravě. Vybrané metody popište, implementujte ve vhodném prostředí a na testovací anotované databázi snímků ověřte jejich účinnost.

Seznam doporučené literatury:

- [1] Gonzalez, R. C., Woods, R. E., Digital image processing, Pearson, 2018.
- [2] Šonka, M., Hlaváč, V., Boyle, R., Image processing, analysis, and machine vision, Cengage Learning, 2015.
- [3] Hsieh, J.-W., et al., Vehicle color classification under different lighting conditions through color correction, IEEE Sensors Journal, 15 (2), 2015.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Karel Fliegel, Ph.D., katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **07.02.2020**

Termín odevzdání diplomové práce: **22.05.2020**

Platnost zadání diplomové práce: **30.09.2021**

Ing. Karel Fliegel, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Josef Dobeš, CSc.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomantka bere na vědomí, že je povinna vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studentky

Abstrakt

Práce se zabývá problematikou klasifikace barev dopravních prostředků a segmentace obrazu jako součást problému. Tato práce obsahuje přehled, implementace a porovnání vybraných metod, které byly navíc rozšířeny o algoritmy zvyšující jejich účinnost. Pro segmentaci byla představena nová metoda založená na principu Bayesova klasifikátoru, jejíž výsledky překonaly ostatní srovnávané metody. Práce je zaměřena na klasické i moderní metody strojového učení. Součástí práce je příprava dvou datových sad pro trénování a testování klasifikačních algoritmů.

Klíčová slova

Segmentace obrazu, klasifikace barev vozidel, metody strojového učení, snímky dopravních prostředků, porovnání klasifikačních algoritmů, MATLAB.

Abstract

This thesis deals with the color classification of vehicles and segmentation as a part of the task. It contains overview, implementations, and comparison of the selected methods, which were extended for algorithms improving the accuracy. A new algorithm was introduced for segmentation based on Bayes classifier and the results surpassed other compared methods. This thesis is focussed on classical and modern algorithms of machine learning. Two datasets were prepared for training and testing the classification algorithms.

Key words

Image segmentation, vehicle color recognition, machine learning algorithms, vehicle images, classifiers comparison, MATLAB.

Obsah

1	Úvod	10
2	Současný stav problematiky klasifikace barev dopravních prostředků	12
3	Digitální reprezentace obrazu	15
3.1	Míchání barev	15
3.2	Barevný model RGB	15
3.3	Barevný model HSV	16
3.4	Barevný model CIELab	17
4	Metody segmentace obrazu	19
4.1	Detekce hran	19
4.1.1	Cannyho detektor	21
4.2	Metody prahování	22
4.3	Metody shlukování	23
4.4	Regionální metody	24
5	Metody klasifikace barev založené na strojovém učení	26
5.1	Metoda podpůrných vektorů (Support Vector Machine, SVM)	27
5.2	Metoda nejbližších sousedů (Nearest Neighbor)	28
5.3	Klasifikace pomocí umělých neuronových sítí	30
5.3.1	Vícevrstvé neuronové sítě	32
5.3.2	Vrstvy neuronových sítí	33
5.3.3	Učení neuronových sítí	34
6	Implementace metod a testování účinnosti	36
6.1	Datové soubory pro vyhodnocení účinnosti algoritmů	36
6.1.1	Datový soubor kaggle	36
6.1.2	Datový soubor traffic	38
6.2	Metrika pro vyhodnocení účinnosti algoritmů	39
6.3	Implementace segmentačních algoritmů	39
6.3.1	Implementace algoritmu k-means	39
6.3.2	Implementace algoritmu detekce hran	42
6.3.3	Implementace Bayesovy segmentace	44
6.4	Klasifikace pomocí barevných definic	46
6.4.1	TOP 2 a TOP 3 accuracy	47
6.5	Implementace algoritmu nejbližších sousedů	48
6.6	Implementace metody podpůrných vektorů (SVM)	49
6.6.1	TOP 2 a TOP 3 accuracy	51
6.7	Klasifikace barev pomocí konvoluční neuronové sítě	52
6.8	Klasifikace barev ve video záznamech	53
6.8.1	Metoda podpůrných vektorů pro video záznam	54
6.8.2	TOP 2 a TOP 3 accuracy pro video záznam	55
6.8.3	Metoda nejbližších sousedů pro video záznam	55
7	Porovnání metod	57
7.1	Porovnání segmentačních metod	57
7.2	Porovnání klasifikačních metod	58
7.2.1	Metoda barevných definic	58

7.2.2	Metoda nejbližších sousedů	59
7.2.3	Metoda podpůrných vektorů	61
7.2.4	Konvoluční neuronová síť	62
7.3	Výsledky porovnání klasifikačních metod	63
8	Závěr	65
A	Přílohy	66
A.1	Výsledky experimentů s metodou podpůrných vektorů	66
A.2	Výsledky experimentů s metodou nejbližších sousedů	70
A.3	Výsledky experimentů při klasifikaci barev ve videu	74
B	Seznam elektronických příloh	77
	Použitá literatura	82

Seznam obrázků

1	Metoda odstranění oken [30]	12
2	Ukázka výběru oblastí zájmu na základě registrační značky vozidla [83] . . .	13
3	Metoda transformace vzdálenosti: 1) binární maska, 2) oblasti zájmu [39] . .	13
4	RGB model [25]	16
5	HSV model [25]	17
6	Lab model [33]	17
7	Segmentační techniky	19
8	Typy hran: a) skoková hrana, b) rampová hrana, c) čára, d) střecha	20
9	Obrazová funkce: 1) původní funkce, 2) první derivace, 3) druhá derivace . .	21
10	Binární maska a segmentovaný obrázek pomocí metody Otsu	22
11	Binární strom hierarchické metody	23
12	Ukázka principu metody K-means [60]	24
13	Princip techniky šíření oblasti	24
14	Princip techniky dělení a spojování	25
15	Fáze strojového učení	26
16	Ukázka rozdělení dat do dvou tříd pomocí metody podpůrných vektorů [80]	27
17	Princip metody nejbližších sousedů	29
18	Rozdíl mezi Euklidovskou (zelená čára) a Manhattanskou metrikou	30
19	Struktura umělého neuronu	31
20	Typy aktivačních funkcí	32
21	Vícevrstvá neuronová síť	32
22	Proces konvoluční vrstvy	33
23	Princip maximálního a průměrného sdružování [78]	34
24	Ukázka neuronové sítě	34
25	Nejprodávanější barvy aut v roce 2018	36
26	Ukázka datového souboru <i>kaggle</i>	37
27	Ukázka datového souboru <i>traffic</i>	38
28	Histogramy průměrných pixelových rozdílů pro trénovací sadu snímků	40
29	Blokové schéma algoritmu rozhodujícího o typu vstupních dat	40
30	Binární masky pro barevné auto	41
31	Binární masky pro bílé auto	41
32	Úplná maska s vyznačenou oblastí pro výběr	41
33	Histogram přesností segmentační metody k-means	42
34	Ukázka výsledků detekce hran pomocí Sobelova a Cannyho detektoru	42
35	Ukázka výsledků dilatace hran	43
36	Ukázka výsledků vyplňování hran	43
37	Blokové schéma segmentační metody detekce hran	43
38	Histogram přesností metody detekce hran	44
39	Ukázka vytváření 2D histogramů	45
40	Ukázka výsledků Bayesovy segmentace	45
41	Histogram přesností Bayesovy segmentační metody	45
42	Složení HSV obrazu	46
43	Blokové schéma detektoru barev v HSV prostoru	47
44	Blokové schéma metody nejbližších sousedů	48
45	Blokové schéma metody podpůrných vektorů	50
46	Architektura konvoluční neuronové sítě [73]	52
47	Průběhy na (a) trénovacích a (b) testovacích datech pro sadu <i>kaggle</i>	53
48	Průběhy na (a) trénovacích a (b) testovacích datech pro sadu <i>traffic</i>	53
49	Extrakce snímků vozidel z celkové scény	54

50	Vzorová maska souboru <i>kaggle</i> (vlevo) a Bayesova maska (vpravo)	57
51	Závislost výpočetní rychlosti na přesnosti pro segmentační metody	58
52	Chybová matice metody barevných definic	59
53	Chybová matice metody nejbližších sousedů	60
54	Chybová matice metody podpůrných vektorů	61
55	Chybová matice konvoluční neuronové sítě	62

Seznam tabulek

1	Parametry klasifikační metody barevných definic nalezené pro datovou sadu <i>kaggle</i> a příklady vyskytujících se barev.	47
2	Přehled výsledků klasifikační metody barevných definic	48
3	Nejlepší dosažené výsledky pro datový soubor <i>kaggle</i>	49
4	Nejlepší dosažené výsledky pro datový soubor <i>traffic</i>	49
5	Nejlepší dosažené výsledky pro metodu podpůrných vektorů	50
6	TOP N výsledky pro datový soubor <i>kaggle</i>	51
7	TOP N výsledky pro datový soubor <i>traffic</i>	51
8	Nejlepší dosažené výsledky metody podpůrných vektorů pro video záznam .	54
9	TOP N výsledky pro video záznam	55
10	Nejlepší dosažené metody nejbližších sousedů pro video záznam	56
11	Přesnosti klasifikačních metod pro jednotlivé barvy datové sady <i>kaggle</i> . . .	63
12	Přesnosti klasifikačních metod pro jednotlivé barvy datové sady <i>traffic</i> . . .	63

Zkratky

ANN	Artificial Neural Network
CMY	Cyan Magenta Yellow
CMYK	Cyan Magenta Yellow Black
HSV	Hue Saturation Value
KNN	K-Nearest Neighbors
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
NN	Nearest Neighbors
ReLU	Rectified Linear Unit
RGB	Red Green Blue
RoI	Region of Interest
SVM	Support Vector Machine

1 Úvod

Cílem této diplomové práce je návrh účinné metody detekce a klasifikace barev dopravních prostředků na snímcích z dopravních kamerových systémů pro následnou analýzu těchto dat v oblasti dopravního inženýrství. Dopravní inženýrství je technická disciplína, mezi jejíž hlavní činnosti patří dopravní průzkumy a prognózy, navrhování a projektování dopravní infrastruktury v kontextu technického řešení i územního plánování, návrhy opatření k zajištění plynulosti a bezpečnosti provozu, vliv dopravy na životní prostředí, uplatňování nových technologií v dopravě atd. Znalost barev některých dopravních prostředků umožňuje klasifikovat jejich typ a účel použití. Například ve Spojených státech amerických většina vozidel taxi a školních autobusů je žluté barvy. Tyto informace pomáhají dopravním inženýrům lépe porozumět situaci na určitém úseku dopravní infrastruktury.

Tato práce vznikla ve spolupráci s firmou GoodVision¹, která se zabývá vývojem softwarové platformy pomáhající dopravním inženýrům a analytikům rychleji získávat a analyzovat dopravní data. Informace o dopravních intenzitách pro jednotlivé typy dopravních prostředků, včetně chodců, jsou získávány z dopravních video záznamů pomocí hlubokých neuronových sítí. Další důležitou informací, kterou může uživatel platformy získat, jsou trajektorie pohybu detekovaných objektů ve scéně. Jako jedno z rozšíření této platformy bylo rozhodnuto zavést možnost filtrace dopravních prostředků podle barev. Škála klasifikovaných barev je omezena na 10 - bílou, černou, šedou, stříbrnou, červenou, zelenou, modrou, hnědou, žlutou a oranžovou. Znalost barvy konkrétního vozidla bude také nápomocna při propojení jednotlivých částí jeho trajektorie v případě, že se např. kvůli překrývání se s jiným objektem rozpojí.

Vzhledem k tomu, že metoda pro detekci a klasifikaci barev bude v praxi běžet zároveň s neuronovou sítí, kladou se na její výběr určité nároky. Musí být rychlá a výpočetně co nejméně náročná. Další podmínkou je, že algoritmus se musí počítat na procesoru a ne na grafické kartě, aby nezpomaloval detekční neuronovou síť. Proto je táto práce zaměřena více na klasické metody počítačového vidění a ne na neuronové sítě, které se běžně používají pro řešení této problematiky. I přesto budou neuronovým sítím v dalším textu věnovány dvě kapitoly 5.3 a 6.7 pro úplnost přehledu používaných metod.

Klasifikace barev se na první pohled může zdát jako jednoduchý úkol, ale není tomu tak. Existuje velké množství faktorů včetně počasí, denní doby, kvality snímacího zařízení a nastavení jeho parametrů (např. vyvážení bílé), které mají dramatický vliv na vjem barev. Kromě toho výrobci automobilů nabízejí tak širokou škálu barev a jejich odstínů, že často i pro člověka je obtížné je klasifikovat. Aby byly barvy detekovány a klasifikovány správně, je potřeba přesně definovat oblast zájmu, která je následně vyhodnocena. Dělá se to různými způsoby, ale nejčastějším je použití segmentačních metod. Dosáhnout dokonalé segmentace objektů v obraze také není snadné, proto jedním z úkolů v rámci této práce je vyzkoušení různých metod segmentace a nalezení takového řešení, které by splňovalo účel a dávalo co nejlepší výsledky (viz kap. 4 a 6.3).

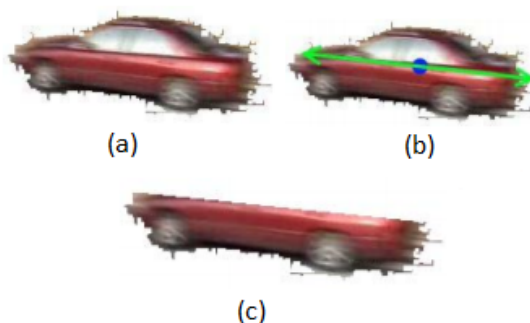
V dalším textu bude podán přehled vybraných metod segmentace obrazu a klasifikace barev včetně teoretické části potřebné pro porozumění problematice (viz kap. 3, 4 a 5). Praktická část zahrnuje popis použitých datových sad s ukázkami (viz kap. 6.1), metrik pro vyhodnocení účinností algoritmů (viz kap. 6.2) a implementace jednotlivých metod včetně

¹Oficiální stránka firmy GoodVision: <https://goodvisionlive.com/>

nejlepších dosažených výsledků (viz kap. 6). Implementace všech metod jsou realizovány v programovacím prostředí MATLAB a jsou k dispozici v elektronické příloze.

2 Současný stav problematiky klasifikace barev dopravních prostředků

Rozpoznávání údajů o vozidle je klíčovou součástí inteligentních dopravních systémů [3]. Jedním z takových údajů je barva vozidla, jejíž klasifikace je v dnešní době velmi aktuálním problémem. Kromě již zmíněných oblastí použití informací o barvách konkrétních vozidel to může být také nápomocné v zajištění bezpečnosti měst, např. při sledování pohybu odcizeného nebo podezřelého auta na dopravních kamerách.



Obrázek 1: Metoda odstranění oken [30]

Existuje velké množství metod a algoritmů, které se snaží barvy dopravních prostředků co nejlépe klasifikovat. Zdroj [30] popisuje eliminaci vlivu světelných podmínek a špatného nastavení parametrů snímacího zařízení pomocí mapování (přenesení) barev mezi obrázky [75] jako způsob jejich korekce. Ke krokům této metody patří odstranění nejenom pozadí (segmentace), ale také části obrazu, která obsahuje okna vozidla (viz obr. 1). Světlo odražené od oken nenese žádnou informaci o barvě určitého auta, způsobuje falešné detekce, a negativně tím ovlivňuje vyhodnocení výsledné barvy. Pro extrakci a modelování pozadí se používá technika kódových knih [46]. Klasifikace je pak realizována pomocí metody podpůrných vektorů (SVM - Support Vector Machine).

Existuje řada metod, které k detekci barev využívají znalost o umístění registrační značky vozidla [83, 14, 16]. Oblast zájmu (RoI - Region of Interest) se pak neurčuje pomocí segmentace a představuje *výřez* určitých rozměrů z původního obrazu. Tento *výřez* je umístěn v blízkém okolí registrační značky (viz obr. 2). Zdroj [16] pro účely klasifikace do sedmi barev použil metodu nejbližších sousedů (NN - Nearest Neighbors), umělou nerunovou síť (ANN - Artificial Neural Network) a metodu podpůrných vektorů. Nejvyšší přesnosti 83,5% se podařilo dosáhnout pomocí umělé neuronové sítě, jejíž vstupem byl barevný histogram omezený na 8 intervalů. Zdroj [83] převádí hodnoty RoI do barevného prostoru HSV, počítá střední hodnotu obrazových bodů pro jednotlivé kanály, čímž získá tři hodnoty popisující barvu vozidla. Pro klasifikaci se pak používá jenom parametr H nabývající hodnot 0-360°, který v HSV prostoru nese informaci o barvě. Taková metoda vyhledávání oblastí zájmu ale má velké množství nevýhod. Je funkční jenom pro snímky přední nebo zadní strany vozidla pod určitým úhlem, registrační značka musí mít na snímku stanovené rozměry atd.

Další zajímavou metodu popisuje [39]. Tato metoda řeší problém gradace barev způsobené světelnými podmínkami. Např. kapota auta může obsahovat gradient barev od úplně tmavé ve stínu do bílé v místech, kde se odráží slunce. Pro nalezení oblastí zájmu se používá technika detekce hran (Edge Detection) a transformace vzdálenosti (Distance Transform)



Obrázek 2: Ukázka výběru oblastí zájmu na základě registrační značky vozidla [83]

[19]. Výstupem těchto technik je binární maska, která znázorňuje vzdálenost pixelů od detekovaných hran. Čím světlejší barvu má pixel, tím větší je jeho vzdálenost od hrany. Nejdálčenější pixely následně slouží jako centroidy jednotlivých oblastí zájmu (viz obr. 3). Klasifikace je následně realizována pomocí multiadaboostu na základě vektorů příznaků jednotlivých oblastí zájmů v barevném prostoru HSV. Multiadaboost je způsob sloučení několika binárních klasifikátorů pro klasifikaci příznaků do více tříd [47].



Obrázek 3: Metoda transformace vzdálenosti: 1) binární maska, 2) oblasti zájmu [39]

Metoda popsaná ve zdroji [45] pracuje hlavně s histogramem obrazu v barevném modelu HSI (hue, saturation, intensity) a počtem jeho intervalů. Barevný histogram se používá jako vektor příznaků, který je následně porovnáván se vzorovými vektory/histogramy detekovaných barev. Porovnáním je výpočet Euklidovské a Manhattan vzdálenosti. Čím více intervalů má histogram, tím podrobnější je barevná informace, ale zároveň se zpomaluje zpracování vektoru příznaků. Objektu je pak přiřazena taková barva, ke které je nejbližší jeho vektor. Metoda dává nejlepší výsledky při použití 4, 16 a 8 intervalů pro histogramy jednotlivých kanálů barevného prostoru HSI. Tato metoda je velmi podobná metodě nejbližších sousedů.

Široce používané pro řešení problému klasifikace barev jsou konvoluční neuronové sítě [73, 1, 31]. Principu jejich fungování a podrobnějšímu popisu neuronové sítě [73] jsou věnovány kapitoly 5.3 a 6.7.

Pro usnadnění klasifikace barev a odstranění vlivu pozadí je vhodné mít segmentační masku vozidla. Pro segmentaci na videích se používá například metoda odečítání pozadí (Background Subtraction) [71] nebo optického toku (Optical Flow) [29]. Poslední dobou mají v segmentaci velmi dobré výsledky také neuronové sítě [66, 27, 72], které jsou bohužel výpočetně velmi náročné.

Komerční firmy nabízející řešení problému detekce barev často spolu s rozpoznáváním

modelu a registrační značky [41, 61] používají buď hluboké neuronové sítě² nebo zmíněnou metodu založenou na znalosti polohy registrační značky³.

²Car Color Recognition, dostupné z: <http://spectrico.com/car-color-recognition.html>

³Eyedeas recognition, dostupné z: <https://cloud.eyedeas.cz/api/VCL>

3 Digitální reprezentace obrazu

V reálném světě se neustále setkáváme s velkým množstvím barev. Budeme-li chtít některou z nich reprodukovat na jiném zařízení (např. na displeji počítače), je potřeba ji jednoznačně definovat pomocí barevného modelu. Tento matematický model popisuje způsoby, jakými mohou být všechny barvy viditelného spektra reprezentovány jako n -tice čísel, obvykle jsou to tři až čtyři hodnoty nebo barevné komponenty. Barevné modely popis barev zjednodušují a přizpůsobují lidskému vnímání, technickým a dalším potřebám. Existuje velké množství barevných modelů pro různé aplikace, které se od sebe liší hlavně základními barvami nebo způsobem jejich míchání. V odborné literatuře zabývající se problematikou klasifikace barev je často nejlepšího výsledku dosaženo reprezentací vstupních dat pomocí barevného prostoru HSV nebo HSI [12, 45, 39, 52]. Barevný prostor RGB a CIE Lab je více používaný v segmentačních algoritmech [88, 17, 20]. Následující podkapitoly budou věnovány typům míchání barev a barevným prostorům, které jsem použila v této práci pro definici požadovaných barev nebo reprezentaci vstupních dat v klasifikačních algoritmech.

3.1 Míchání barev

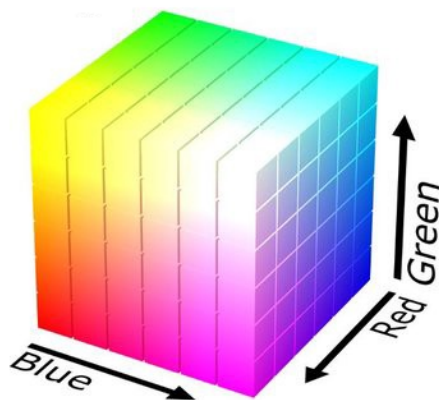
Rozlišují se dva základní typy míchání barev - **aditivní** a **subtraktivní**. V obou případech je míchání obvykle popisováno pomocí tří základních a tří sekundárních (doplňkových) barev, které jsou vytvořeny kombinací dvou základních barev ve stejném poměru.

Aditivní míchání barev je vlastnost barevného modelu, která funguje na principu sčítání jednotlivých barev - červené, zelené a modré (např. barevný model RGB). Pokud jsou všechny tři základní barevné složky v obraze zastoupeny rovnoměrně s maximální intenzitou, je získaná barva bílá. Naopak smícháním barev s minimální intenzitou získáme černou. Červená, zelená a modrá se spojují po dvou, čímž se vytvoří aditivní sekundární azurová, purpurová a žlutá [36]. Aditivní míchání se používá pro návrh a testování televizních a počítačových monitorů, včetně displejů chytrých telefonů, pro reprodukci široké škály barev.

Subtraktivní míchání barev je naopak založeno na principu odečítání - s každou další přidanou barvou se ubírá část původního světla. Světlo prochází jednotlivými barevnými vrstvami a je stále více pohlcováno. Výsledná barva se skládá z vlnových délek, které zbudou po odrazu nebo průchodu filtrem [36]. Tři primární barvy, obvykle používané v systémech subtraktivního míchání barev, jsou azurová, purpurová a žlutá, což odpovídá barevným modelům CMY a CMYK široce používaným při barevném tisku. Při subtraktivním míchání barev rovnoměrné zastoupení všech tří primárních barevných složek a nepřítomnost bílé vytváří neutrální tmavě šedou nebo černou. Sekundární barvy jsou stejné jako základní barvy pro aditivní míchání. Subtraktivní míchání se používá v tiskárnách nebo při malování na papír.

3.2 Barevný model RGB

RGB (red - červená, green - zelená, blue - modrá) je aditivním barevným modelem (viz kap. 3.1), ve kterém se jednotlivé barevné složky sčítají různými způsoby pro reprodukci široké škály barev (např. 16 777 216 barevných odstínů pro 24-bitovou reprezentaci). Tento model je založen na teorii lidského vnímání barev, která říká, že lidské oko je nejvíce citlivé na tři barvy - červenou, zelenou a modrou. Ostatní barvy jsou dány jejich sytostí.



Obrázek 4: RGB model [25]

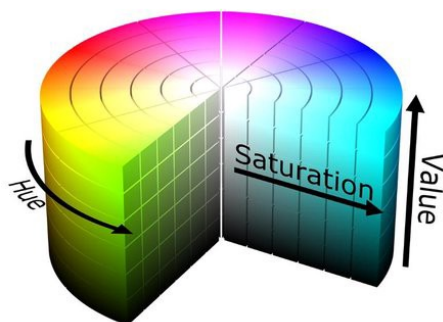
Model RGB je reprezentován ve tvaru krychle v kartézské soustavě souřadnic, kde každý vrchol odpovídá jedné z primárních nebo sekundárních barev. V počátku souřadnicového systému je černá barva (0,0,0) a v protilehlém rohu krychle je barva bílá (1,1,1). RGB triplet představuje trojrozměrnou souřadnici bodu (r, g, b nebo x, y, z) dané barvy uvnitř krychle nebo na jejích okrajích, která vyjadřuje každou barvu nebo její odstín jako trojici kladných čísel v intervalu $< 0, 1 >$ na základě intenzity [24]. Pro 24-bitovou reprezentaci obrazu je horní hranice intervalu 255. Tento přístup umožňuje výpočet podobností dvou barev pomocí euklidovské vzdálenosti mezi nimi: čím je vzdálenost menší, tím vyšší je podobnost.

3.3 Barevný model HSV

HSV (hue - odstín, saturation - sytost, value - světelnost) je alternativní reprezentací barevného modelu RGB, navrženou tak, aby se ještě více přiblížila lidskému vnímání barev. Je to proto, že ani aditivní, ani subtraktivní barevné modely nedefinují barevné vztahy stejným způsobem jako lidské oko. V tomto modelu jsou všechny barvy a jejich odstíny uspořádány do válce kolem centrální osy neutrálních barev, která sahá od černé ve spodní části až po bílou v horní části (viz obr. 5). Úhel kolem osy odpovídá odstínu, vzdálenost od osy odpovídá saturaci a vzdálenost podél osy odpovídá světlosti (hodnotě jasu).

- **Odstín** je převládající barva odražená nebo procházející objektem. Měří se jako poloha na standardním barevném kole (0 až 360).
- **Sytost** je příměs jiné barvy. Někdy též chroma, síla nebo čistota barvy, představuje množství šedi v poměru k odstínu, měří se v procentech od 0% (šedá) do 100% (plně sytá barva).
- **Světelnost (hodnota jasu)** - množství bílého světla. Relativní světlost nebo tmavost barvy. Jas vyjadřuje, kolik světla barva odráží [24].

Barevné složky R, G a B se dají snadno přepočítat na hodnoty H, S a V pomocí následu-



Obrázek 5: HSV model [25]

ující vzorců:

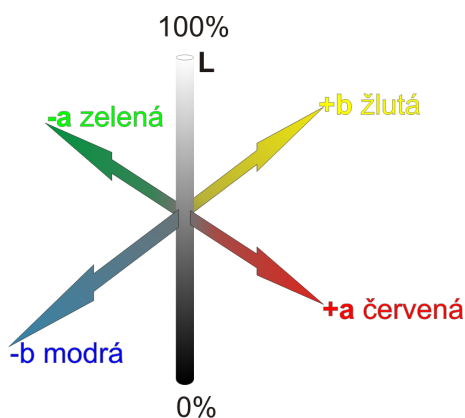
$$H = 60^\circ \times \begin{cases} 0 + \frac{G-B}{\max-\min}, & \text{pro } \max = R, \\ 2 + \frac{B-R}{\max-\min}, & \text{pro } \max = G, \\ 4 + \frac{R-G}{\max-\min}, & \text{pro } \max = B \end{cases} \quad (1)$$

$$S = \frac{\max - \min}{\max} \quad (2)$$

$$H = \max \quad (3)$$

kde \max je nejvyšší hodnota z R, G, B a \min je nejmenší. HSV model se často používá v počítačovém vidění a analýze obrazu pro detekci prvků nebo segmentaci obrazu [12, 45, 39, 52]. Kvůli tomu, že barevné složky R, G a B určitého objektu v digitálním obraze jsou ovlivněné množstvím světla, které na tento objekt dopadá, je definice barev pomocí tohoto modelu pro rozpoznávání objektů o něco náročnější.

3.4 Barevný model CIELab



Obrázek 6: Lab model [33]

Lab je zkrácený název pro barevný model specifikovaný Mezinárodní komisí pro osvětlení (CIE – Commission Internationale de l'Eclairage). Tento model je navržen tak, aby popisoval všechny barvy viditelného spektra a také byl nezávislý na zařízení. Stejně jako u předchozích modelů, zkratka Lab odpovídá názvům tří kanálů (složek), pomocí kterých jsou definovány jednotlivé barvy. Písmeno **L** (luminance) odpovídá **jasové složce**, která je

v tomto modelu zcela oddělena od barevných. Díky tomu je možné např. upravovat barvy snímků bez ovlivnění světelnosti, což by mohlo způsobit nežádoucí artefakty. Jas L může nabývat hodnot od 0% do 100%, kde nula odpovídá nejtmaší černé a sto nejsvětleší bílé barvě [33].

Složka **a**, jak je vidět na obr. 6, odpovídá barevné škále od zelené v záporném směru až po červenou barvu v kladném. Složka **b** představuje modro žluté barvy, s modrou v záporném směru a žlutou v kladném. Oba kanály nabývají hodnot v rozsahu -128 až +127. Čím vyšší (případně nižší) je hodnota, tím sytější je pak barva, přičemž nula představuje odstín šedi.

V praxi se barevný prostor CIELAB často používá pro konverzi mezi modely RGB a CMYK, jelikož pro to neexistují jednoduché metody a Lab zahrnuje gamuty obou modelů.

4 Metody segmentace obrazu

Pro vyšší přesnost klasifikace barev je dobré umět rozeznat vozidlo od pozadí. Segmentace je metoda automatického rozdělení obrazu na oblasti, nazývané segmenty, které mají nějaké společné vlastnosti - pixely, hodnotu jasu, barvu, texturu atd. Výsledky segmentace se používají pro kompresi obrazu [18], rozpoznávání a detekci objektů [27, 72], zpracování lékařských obrazových dat [7, 65] nebo automatické systémy řízení dopravy [62, 87], kde není efektivní zpracovávat celý obraz. Cílem segmentace je tedy extrakce určité části obrazu pro následné zpracování.

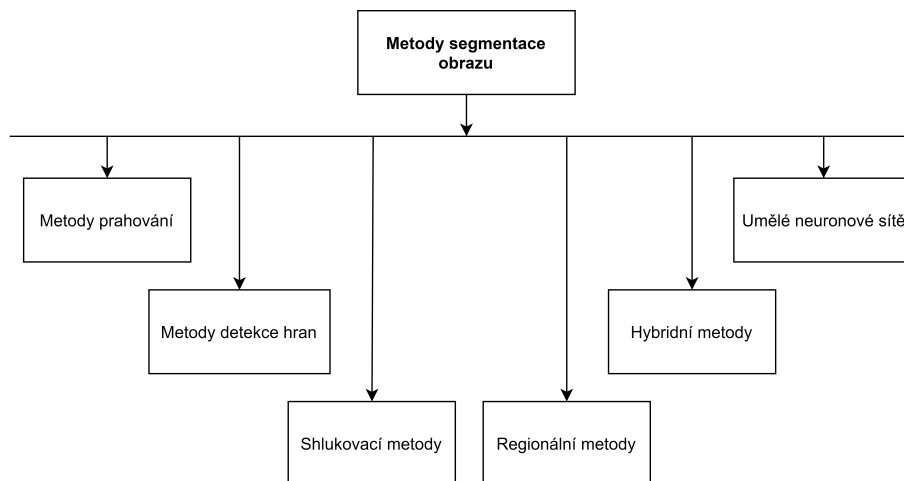
Podle [35] se rozlišují dva základní typy segmentačních metod:

- **Local segmentation** (místní segmentace) - týká se konkrétní části obrazu
- **Global segmentation** (globální segmentace) - zpracovává se celý obraz, skládající se z velkého počtu pixelů.

Podle vlastností obrazu se segmentační metody dělí na:

- **Přístup založený na detekci diskontinuity** - tohoto přístupu využívá technika detekce hran (angl. edge detection), která detekuje hrany vytvořené v důsledku rozdílů intenzity, a vytváří takto hranice objektů.
- **Přístup založený na detekci podobnosti** - tento přístup je základem několika segmentačních technik: metody prahování (angl. thresholding), regionální metody (angl. region-based methods) a metody klastrování (angl. clustering methods).

V dalších odstavcích budou podrobněji popsány jednotlivé metody segmentace, jejich výhody a nevýhody.



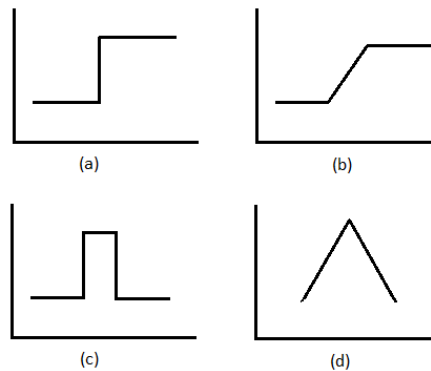
Obrázek 7: Segmentační techniky

4.1 Detekce hran

Detekce hran zahrnuje řadu matematických metod, které se zaměřují na identifikaci bodů v digitálním obrazu, ve kterých dochází k výrazné změně jasu. Body, ve kterých se jas mění ostře, jsou obvykle uspořádány do sady zakřivených úseček nazývaných hrany. Místa

v obraze odpovídající významným hranám nesou nejvíce informace o objektech. Zbytek se může zanedbat a ve většině případů to nijak neovlivní porozumění příslušné scéně. Výsledkem metod detekce hran je takzvaná **mapa hran**, pomocí které je následně vypočten konečný segmentovaný obraz.

Hranu tedy můžeme chápat jako vlastnost obrazového bodu započteného jako funkce obrazu v okolí tohoto bodu. Je reprezentována velikostí a směrem. Změny či přerušení v jasu jsou jedny z nejzákladnějších charakteristik, protože naznačují fyzické rozmístění objektů v obraze [34]. Ideální hrana je vyjádřena **skokovou** (angl. step) funkcí. V reálných obrazech není změna jasu nikdy skoková, ale postupná - používá se **rampová funkce** (angl. ramp). Kombinací zmíněných dvou funkcí vznikají další typy hran - **čára** (angl. line) a **střecha** (angl. roof).



Obrázek 8: Typy hran: a) skoková hrana, b) rampová hrana, c) čára, d) střecha

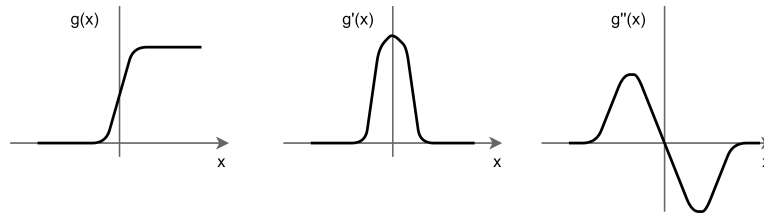
Pro detekci hran se používají metody založené na **první** nebo **druhé derivaci**, techniky **srovnávání se vzorem** a některé **statistické metody**.

Metody **první derivace** jsou takzvané **gradientní metody**. Nejpoužívanějším způsobem výpočtu jednotlivých složek tohoto gradientu je použití hranového operátoru - Sobelova, Prewittova, Robinsonova nebo Kirschova [79, 44]. Jednotlivé hranové operátory se liší konvolučním jádrem, které se následně aplikuje na originální obraz pro výpočet požadované složky gradientu. Při použití první derivace je výsledný hranový gradient porovnán s prahem, který určuje, zda se jedná o hrany či nikoli [34].

Jádro hranového operátoru udává, které body se pro výpočet gradientu použijí a jaký budou mít vliv. Velikost jádra významně ovlivňuje citlivost operátoru na šum, proto jejich rozměry nejsou omezeny pouze na 3×3 , ale používají se i větší např. 5×5 nebo 7×7 , které pracují s větším okolím.

Metody založené na výpočtu **druhé derivace** místo detekce maximální změny intenzity detekují průchody obrazové funkce nulou (zero-crossing). Tyto metody jsou považovány za spolehlivější díky strmosti přechodu. Hlavní nevýhodou metod je větší citlivost na šum [64].

Metody druhé derivace většinou pracují s Laplaceovým operátorem [79, 54], jehož vlastností je, že klade důraz na středové body a součet všech jeho prvků je roven nule. Je vhodný pro doostřování obrazu a není výpočetně náročný. Nevýhodou však je výraznější citlivost na šum a dvojitě odezvy na některé hrany v obraze.



Obrázek 9: Obrazová funkce: 1) původní funkce, 2) první derivace, 3) druhá derivace

4.1.1 Cannyho detektor

Cannyho hranový detektor [79] si zaslouží zvláštní pozornost, protože je považován za jeden z nejlepších [58]. Cannyho detektor se liší od ostatních zmíněných detektorů (operátorů) hlavně tím, že nepředstavuje jenom určité konvoluční jádro, ale je algoritmem zahrnujícím několik kroků pro dosažení co nejlepšího výsledku.

Hlavními požadavky na Cannyho detektor jsou: minimální chybovost, co největší přesnost lokalizace polohy hran a jednoznačná odezva na hranu. Aby všechny tyto požadavky byly splněny, je Cannyho algoritmus realizován pomocí následujících kroků:

1. Odstranění šumu, většinou pomocí Gaussova filtru
2. Výpočet první derivace (určení gradientu) obrazové funkce nejčastěji pomocí Sobelova operátoru.
3. Nalezení lokálních maxim - ztenčení hran (angl. thinning) a získávání jednoznačné odezvy na hranu.
4. Odstranění nevýznamných hran prahováním

Odstranění šumu pomocí Gaussova filtru je založeno na konvoluci obrazu s konvolučním jádrem. Hodnoty konvolučního jádra se určují pomocí rovnice Gaussovy funkce. Gaussovův filtr se většinou používá pro eliminaci Gaussova šumu mírným rozostřením původního snímku.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} \quad (4)$$

$$G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (5)$$

Ztenčení hran (angl. thinning) potlačuje hodnoty (pixely), které nejsou lokálními maximy. Prakticky si to můžeme představit tak, že skutečná hrana prochází určitým pixelem, který musí mít z obou stran sousedy s nižší hodnotou gradientu, což lze matematicky popsat nerovnicí $E1 < E > E2$. Aby byl libovolný pixel uznán jako skutečná hrana, musí splňovat zmíněnou nerovnici. Pokud tomu tak není, tato hrana se odstraní. Více informací o metodách prahování a jejich principech je v další kapitole.

4.2 Metody prahování

Prahování (angl. tresholding) je řada jednoduchých metod pro segmentaci obrazu, které jsou založené na rozdělování obrazových bodů podle úrovně jejich intenzity. Jejich principem je nalezení takové hodnoty (prahu) v histogramu, pro kterou bude platit, že všechny hodnoty jsou nižší než práh odpovídají pozadí, zatímco všechny hodnoty vyšší než práh odpovídají popředí [40]. Tyto metody mají největší účinnost na obrazech, kde objekty v popředí mají výrazně jinou intenzitu než pozadí. Existují tři typy prahování:

- **Globální prahování** - pro tuto metodu se najde jakákoli vhodná prahová hodnota T , která je následně konstantní pro celý obraz. Pomocí T lze výstupní segmentovaný obraz $q(x, y)$ získat ze vstupního $p(x, y)$ takto:

$$q(x, y) = \begin{cases} 1, & \text{pro } p(x, y) > T \\ 0, & \text{pro } p(x, y) \leq T \end{cases} \quad (6)$$

- **Variabilní prahování** - u tohoto typu prahování se hodnota prahu T může v obrazu lišit [50]. Pokud je T závislé na okolí bodů x a y , mluvíme o **lokálním prahování**. Pokud je ale T funkcí x a y , v tom případě jde o prahování **adaptivní**. Adaptivní prahování se používá pro nehomogenní obrazy obsahující různé jasné úseky, které neumožňují určit globální práh přímo [40].
- **Vícenásobné prahování** - tento typ prahování [50] vyžaduje určení více prahových hodnot jako např. T_0 a T_1 , pomocí kterých se výstupní obrázek $q(x, y)$ vypočítá jako:

$$q(x, y) = \begin{cases} m, & \text{pro } p(x, y) > T_1 \\ n, & \text{pro } p(x, y) \leq T_1 \\ o, & \text{pro } p(x, y) \leq T_0 \end{cases} \quad (7)$$

Hodnoty prahů se počítají z vrcholů histogramu obrazu pomocí jednoduchých algoritmů. Nejvíce používané jsou metody maximální entropie (maximum entropy threshold) [89], vyvážené prahování histogramu (angl. balanced histogram thresholding) [4], metoda Otsu (maximální rozptyl) a k-means shlukování [55, 57]. Výstupem zmíněných algoritmů je binární maska, která se následně aplikuje na originální obraz.



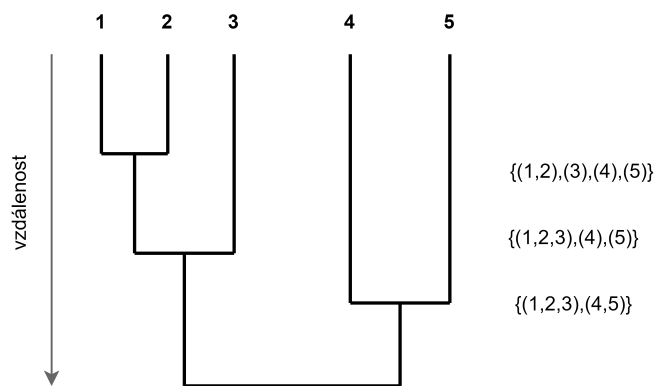
Obrázek 10: Binární maska a segmentovaný obrázek pomocí metody Otsu

Nevýhodou těchto metod je, že jsou velmi závislé na vrcholech histogramu a neberou v úvahu prostorové detaily. Nejsou účinné pro obrázky, které obsahují více objektů a mají nehomogenní pozadí. Proto pro účely této diplomové práce není nejlepší volbou jako samostatná metoda.

4.3 Metody shlukování

Techniky založené na shlukování (angl. clustering) rozdělují obrazové body do požadovaného počtu shluků (skupin) podle jejich vlastností (jas, barva atd.) tak, aby si obrazové body náležející do jednoho shluku byly podobnější než ty z ostatních skupin [35]. Existují dvě základní kategorie shlukových metod: **hierarchické** a **nehierarchické**.

Hierarchické metody [35] jsou založené na konceptu binárního stromu, kde kořen stromu představuje celou databázi a jednotlivé uzly představují shluky. Jak je vidět na obr. 11, na začátku hierarchického shlukování máme n jednoprvkových datových shluků, které jsou následně postupně sloučeny na základě jejich vzájemné podobnosti. Proces sloučení se zastaví v momentě, kdy zbude pouze jeden shluk. Výška stromu, ve kterém jsou sloučeny dva shluky, odpovídá jejich vzdálenosti v datovém prostoru [68]. Mezi hierarchické metody patří: metoda nejbližšího/nejvzdálenějšího souseda (Simple/Complete linkage), centroidní metoda (Centroid linkage) [84] a metoda průměrné vazby (Average linkage) [38].

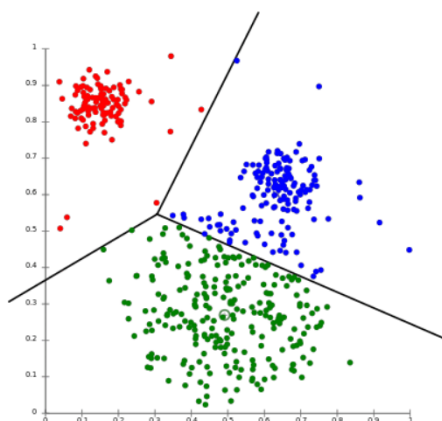


Obrázek 11: Binární strom hierarchické metody

Na druhé straně **nehierarchické metody** používají optimalizační metody s cílem minimalizovat nebo maximalizovat nějakou charakteristiku rozkladu. V nehierarchických shlukovacích metodách je počet shluků obvykle předem dán, i když se v průběhu výpočtu může změnit. Zůstává-li počet shluků zachován, hovoříme o *nehierarchických metodách s konstantním počtem shluků*, v opačném případě o *nehierarchických metodách s optimalizovaným počtem shluků* [2].

Nejpoužívanější a nejznámější nehierarchickou metodou je **k-means** [57]. Jednotlivé kroky metody:

1. Výběr počtu shluků
2. Určení (automaticky nebo manuálně) těžiště jednotlivých shluků
3. Přiřazení každého pixelu ke shluku, jehož těžiště je nejbližší
4. Pokud se vybraný shluk nerovná původnímu, přemístění pixelu do něj a přepočítání těžiště
5. Proces pokračuje, dokud už nedochází k žádným změnám shluků

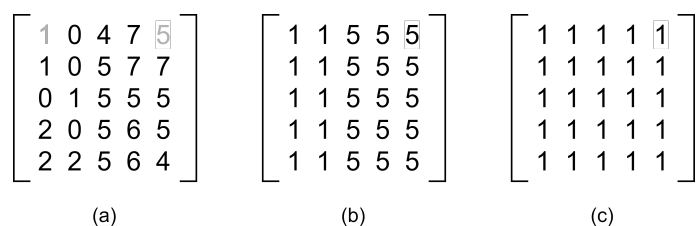


Obrázek 12: Ukázka principu metody K-means [60]

4.4 Regionální metody

Regionální metody (region based method) vycházejí z předpokladu, že sousední pixely v jedné oblasti mají podobné hodnoty. Tyto metody jsou efektivnější pro složitější obrázky a jsou více odolné proti šumu než např. metody detekce hran. Je-li v obraze hodně šumu, hranové operátory obtížně detekují hrany. Hlavním segmentačním kritériem pro detekci oblasti v obraze je její homogenita. Kritériem homogenity mohou být: úroveň šedi, barva, textura, tvar, model, apod [34]. Regionálních metod využívají dvě techniky - **šíření oblastí** (region growing) a **dělení a spojování oblastí** (split and merge).

Podstatou techniky **šíření oblastí** je vytvoření celého segmentu postupně od jednoho pixelu. Jako první krok se v obraze rozmístí počáteční (semínkové) body, které jsou na obr. 13 označené šedou barvou. Tyto body se vybírají ručně (na základě předchozí znalosti) nebo automaticky (na základě konkrétní aplikace). Následně se provede seskupování podobných pixelů kolem počátečního do jedné oblasti na základě určitého kritéria. V případě, který je uveden na obr. 13, byla použita podmínka, že pokud je absolutní hodnota rozdílu mezi pixelem X a počátečním bodem menší než prahová hodnota T , tak se pixel X zahrne do oblasti, ve které je tento počáteční bod umístěn.

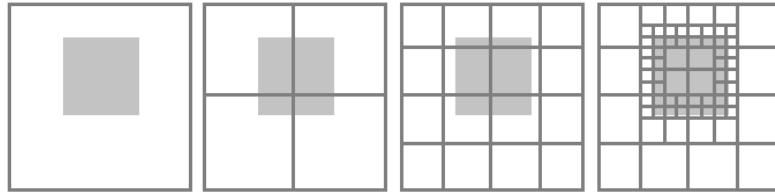


Obrázek 13: Princip techniky šíření oblastí

Na obr. 13 (b) je znázorněn výsledek pro hodnotu prahu $T = 3$; pixely jsou rozdělené do dvou oblastí. Obr. 13 (c) je výsledkem šíření regionu pro $T = 6$; všechny pixely patří jedné oblasti [85]. Výběr prahu je tedy velmi důležitý.

Technika **dělení a spojování** má principiálně opačný princip. Využívá pyramidové reprezentace obrazu – je realizováno v rámci čtvercových oblastí pyramidové datové struktury [37]. Na začátku je originální obraz jedinou oblastí, která se pak postupně dělí do menších a menších podoblastí podle určitého kritéria homogenity, a sousední oblasti se naopak spo-

jují dohromady, mají-li shodné vlastnosti (viz obr. 14). Kritéria pro dělení a pro spojování regionu se mohou lišit.



Obrázek 14: Princip techniky dělení a spojování

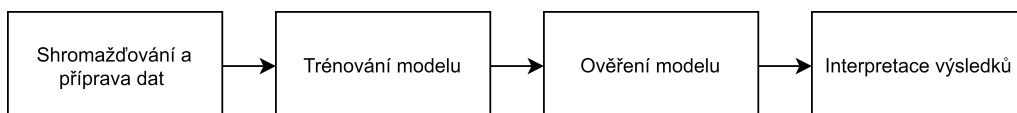
5 Metody klasifikace barev založené na strojovém učení

V dnešní době technologií a velkého množství informací se problematika detekce a klasifikace objektů, barev, snímků atd. nejčastěji řeší pomocí metod strojového učení (Machine learning). Strojové učení je odvětvím umělé inteligence (Artificial intelligence), které se zabývá technikami a algoritmy trénování počítačů pomocí dat tak, aby napodobovaly lidskou inteligenci. Co si máme představit pod pojmem učení? Hlavním principem strojového učení je hledání podobností a vzorů ve vstupních datech, které se následně uloží do tzv. datového modelu. Na základě datového modelu (svých znalostí) dokáže algoritmus generovat předpovědi. Přesnost předpovědi závisí na množství a rozmanitosti vstupních dat a použitém algoritmu. Dnes se strojové učení používá v nejrůznějších odvětvích jako např. medicína [65], zemědělství [53], zákaznické služby [48], doprava [87, 12] atd.

Techniky strojového učení se dělí na 3 hlavní skupiny [70]:

- **Učení bez učitele** (Unsupervised Learning) - principem této metody je nechat stroj samostatně najít vzory ve vstupních datech [57, 55]. Správná třída není známa.
- **Učení s učitelem** (Supervised Learning) - trénovací data obsahují také předdefinovanou ukázkou výsledků, které se očekávají na výstupu [27, 72, 6]. Nevýhodou této metody je, že je časově náročnější a náchylnější k chybám. Při učení s učitelem může také dojít k tzv. **přeučení stroje** (angl. overfitting) - přílišné přizpůsobení systému množině trénovacích dat. V tomto stavu není stroj schopen rozhodovat a selhává na testovací množině. Může to být způsobeno malým rozsahem trénovacích dat, složitostí klasifikátoru nebo vysokou dimenzí extrahovaných příznaků (např. příliš dlouhý vektor příznaků).
- **Zpětnovazebné učení** (Reinforcement Learning) - během učení stroj (agent) interaguje s prostředím pomocí akcí a dostává zpětnou vazbu, na jejímž základě mění svoje chování tak, aby byl dosažen co nejlepší výsledek [42, 81].

Na obr. 15 jsou znázorněny fáze strojového učení. Prvním a nejdůležitějším krokem je **příprava vstupních dat** - rozdělení dat na disjunktní trénovací a testovací množiny. Vstupní data se mohou lišit podle účelu a oblasti použití natrénovaného modelu. V případě detekce barev jsou jako vstupní data použity sady snímků dopravních prostředků vybraných barev. Z připravených dat je potřeba extrahovat jejich příznaky (angl. features), na kterých se bude stroj učit, např. histogram. Další fází je **trénování modelu** - stroj se na základě trénovacích dat naučí požadovanou činnost pomocí jedné z technik popsanych výše. Pokračuje **ověření účinnosti** natrénovaného modelu pomocí testovací sady. Pokud přesnost modelu nespĺňuje očekávání, pokračuje se v ladění parametrů nebo trénovací množiny dat a celý proces se opakuje. Posledním krokem je **vyhodnocení výsledků** konečného modelu, formulování závěrů a předpověď výsledků.



Obrázek 15: Fáze strojového učení

Mezi problémy, které lze vyřešit pomocí strojového učení a které jsou aktuální pro tuto

práci, patří **klasifikace** a **shlukování**. Klasifikace vstupních dat se nejčastěji řeší pomocí techniky učení s učitelem. Cílem je roztrdit vstupní data tak, aby patřila do konkrétních tříd. Existují tři typy klasifikace:

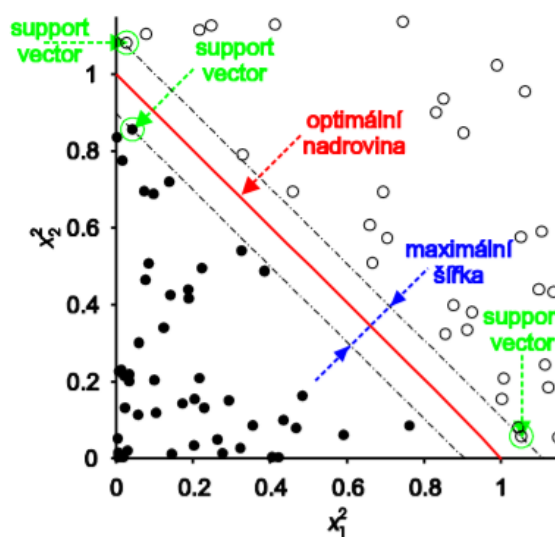
- **Binární klasifikace** - vstupní množina dat je rozdělena do dvou tříd.
- **Multi-class klasifikace** - jak již je patrné z názvu tohoto typu klasifikace, jde o rozdělování dat do N kategorií ($N > 2$). Důležité je, že jeden klasifikovaný objekt může patřit jenom k jedné z kategorií. Tento typ klasifikace byl použit při testování vybraných modelů strojového učení.
- **Multi-label klasifikace** - na rozdíl od multi-class klasifikace může objekt patřit zároveň k několika třídám nebo také k žádné.

Shlukování se naopak řeší pomocí techniky učení bez učitele a vstupní data se dělí do shluků na základě podobnosti, o kterých zdroj rozhoduje sám.

V následujících podkapitolách budou podrobněji vysvětleny dva vybrané klasifikační modely strojového učení, které byly testovány v rámci této práce - metoda podpůrných vektorů (angl. support vector machine) a metoda nejbližších sousedů.

5.1 Metoda podpůrných vektorů (Support Vector Machine, SVM)

Metoda podpůrných vektorů [9] je metoda strojového učení s učitelem a jejím hlavním cílem je najít takovou nadrovinu v N -dimenzionálním prostoru, která optimálně rozdělí vstupní data patřící do definovaných tříd a zároveň bude mít maximální vzdálenost mezi jejími datovými body (viz maximální šířka na obr. 16). Datovým bodům rozhodujícím o tom, které rozdělení je správné, se říká podpůrné vektory (support vectors). Tyto vektory leží v nejmenší vzdálenosti od vektoru patřícího do jiné třídy dat. Jsou také významné tím, že se následně ukládají pro další zpracování a ostatní vektory se zanedbávají, díky čemuž dochází k optimalizaci a kompresi dat. Odstraněním podpůrných vektorů se změní poloha dělicí roviny.



Obrázek 16: Ukázka rozdělení dat do dvou tříd pomocí metody podpůrných vektorů [80]

V případě základního algoritmu metody SVM, kde jsou datové body lineárně separované do dvou tříd, lze tuto dělicí nadrovinu popsat rovnicí:

$$\mathbf{x} \cdot \mathbf{w} + b = 0 \quad (8)$$

kde \mathbf{w} je normálový vektor a b je konstanta. Pokud je trénovací množina ve tvaru dvojic (\mathbf{x}_i, y_i) , kde \mathbf{x}_i je vektor příznaků a y_i je vzor od učitele nabývající hodnot ± 1 , pak platí:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \quad \text{pro } y_i = +1 \quad (9)$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \quad \text{pro } y_i = -1 \quad (10)$$

Proces učení spočívá v hledání takového b a \mathbf{w} , které odpovídají rovnicím 9 a 10. Popsanému způsobu dělení datových bodů se také říká binární klasifikace. Na jejím principu jsou založeny některé techniky multi-class klasifikace, která je pro problematiku detekce barev vhodnější. Tyto techniky zjednodušují problém rozdělení objektů do více tříd tak, že ho rozdělí na více podproblémů, které se dají snadno vyřešit pomocí binární klasifikace. Lze je rozdělit do kategorií **One vs All** (jeden proti všem) a **One vs One** (jeden proti jednomu).

Principem techniky **One vs All** je vytvoření binárního klasifikátoru pro každou z definovaných tříd. Například klasifikátor pro červenou barvu přiřadí hodnotu 1 všem snímkům s červenými objekty a hodnotu -1 všem ostatním. Stejným způsobem se počítají modely pro ostatní třídy. Celkový modul provádí předpovědi pomocí všech binárních klasifikátorů a následně vybírá tu, která má nejvyšší skóre spolehlivosti - pravděpodobnost, že spadají do dané třídy. Znalost pravděpodobnosti náležitosti je velkou výhodou této techniky, která umožňuje s daty dál pracovat, např. prahovat, rozšířit výsledky o top N accuracy atd.

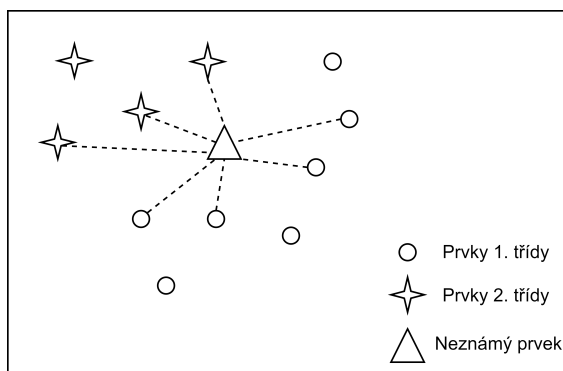
Druhá technika **One vs One** vytváří klasifikátory pro všechny možné páry tříd, tedy máme $|C|(|C| - 1)/2$ modelů, kde C je počet definovaných skupin. Opět se provádí předpověď pomocí všech klasifikátorů a vybírá se ta třída, která byla zvolena většinou modelů. Nevýhodou této techniky oproti předchozí je, že je výpočetně náročnější.

Případ nelineární klasifikace (kernel SVM) a klasifikace lineárně neseperabilních dat v tomto textu popsán nebude. Více informací k nalezení zde [59].

5.2 Metoda nejbližších sousedů (Nearest Neighbor)

Metoda nejbližších sousedů [10] je jednou z nejstarších a nejjednodušších metod. Patří mezi metody učení s učitelem a je nebinárním klasifikátorem - umožňuje klasifikaci do více tříd. Princip algoritmu je založen na porovnání extrahovaných příznaků (features) z testovacích instancí se vzory natrénovaného modelu. Jak již bylo zmíněno v předchozí podkapitole, příznakem snímku, který nese nejvíce informace, může být histogram ve vybraném barevném prostoru.

Jak přesně probíhá proces porovnání? Pro každý klasifikovaný objekt se vypočte jeho vzdálenost od všech objektů trénovací sady. Vzdálenost nepřímo úměrně vyjadřuje míru podobnosti jednotlivých dat. Následně se klasifikovanému objektu přiřadí třída objektu, ke kterému je nejbliž. Matematicky se dá metodu nejbližších sousedů popsat vzorcem:



Obrázek 17: Princip metody nejbližších sousedů

$$y = \arg \min_{x_i, y_i \in T} D(\mathbf{x}, \mathbf{x}_i) \quad (11)$$

kde x_i a y_i je trénovací objekt a jeho třída, T je trénovací množina, D je metrika a y je klasifikovaná třída.

Pro měření vzdáleností se používá velké množství různých metrik. Nejčastěji se ale používá Euklidovská nebo Hammingova vzdálenost. V odborné literatuře se také často vyskytuje tzv. Manhattaná vzdálenost [11].

Hammingova vzdálenost je vhodná jenom pro specifické případy, kdy vektory příznaků jsou od sebe odlišné jen minimálně. Například se používá pro porovnání binárních slov (čísel), kde výslednou vzdáleností je počet bitů, které jsou od sebe odlišné. Bohužel, v případě porovnání dvou snímků nebo histogramů mohou být rozdíly tak velké, že daná metrika nebude dostatečně účinná.

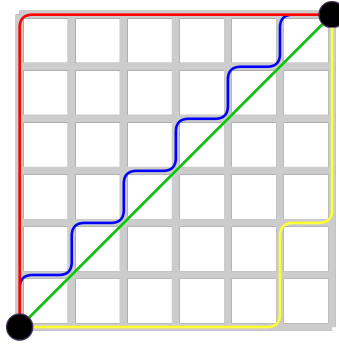
Euklidovská vzdálenost je metrika, kterou lze snadno odvodit pomocí Pythagorovy věty. Pro vektory příznaků v N -rozměrném prostoru např. $a = (a_1, a_2, \dots, a_n)$ a $b = (b_1, b_2, \dots, b_n)$ spočítáme jejich vzdálenost pomocí vzorce:

$$d_2(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (12)$$

V porovnání s ostatními zmíněnými metrikami je účinnější, protože klade větší důraz na velké rozdíly v konkrétních dimenzích.

Manhattanská vzdálenost [11] vychází z předpokladu, že nejkratší vzdálenost mezi body A a B je taková, kterou je třeba ujít mezi dvěma křižovatkami na Manhattanu, mezi kterými se lze pohybovat jen po na sebe kolmých ulicích ve směru obou os [22]. Matematicky je vyjádřena vzorcem:

$$d_1(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^n |a_i - b_i| \quad (13)$$



Obrázek 18: Rozdíl mezi Euklidovskou (zelená čára) a Manhattanskou metrikou

Kosinová podobnost je založena na výpočtu skalárního součinu a vyjadřuje míru podobnosti dvou vektorů, která se získá výpočtem kosinu úhlu těchto vektorů. Výhodou této metriky je, že pokud je příznakovým vektorem nenormovaný histogram, pak na rozlišení snímku nezáleží. Matematicky je vyjádřena následujícím vzorcem:

$$D_{cos} = \cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}} \quad (14)$$

kde $\|\mathbf{a}\|$ a $\|\mathbf{b}\|$ jsou normy (délky) vektorů.

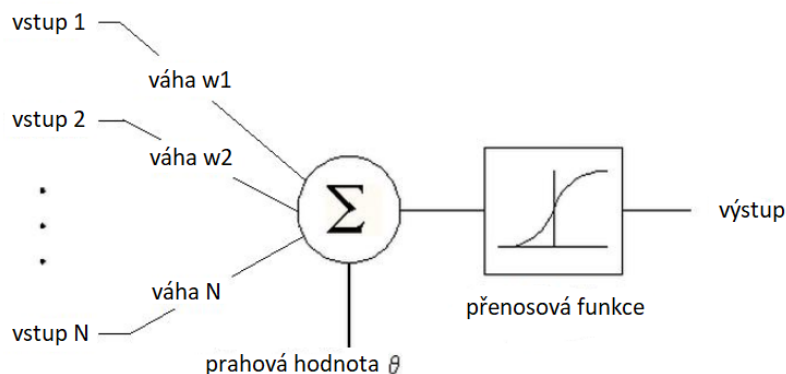
Zdroj [63] uvádí, že „nevýhodou metody nejbližšího souseda je její značná citlivost na odlehle hodnoty. Obzvláště v situaci, kdy se třídy částečně překrývají, zpravidla nedává dobré výsledky. Proto se v praxi častěji používá její zobecnění, což je metoda **k-nejbližších sousedů**, při níž zařadíme subjekt, který chceme klasifikovat, do té třídy, která převažuje mezi jeho nejbližšími sousedy“. Počet sousedů k je zpravidla liché číslo. Pokud k bude sudé, může snaze dojít k situaci, kdy počet sousedů v okolí klasifikovaného objektu bude stejný ze všech tříd, a proto nebude možné jednoduše rozhodnout, do které třídy patří vstupní objekt. Pokud by taková situace nastala, většinou se subjekt zařadí náhodně do jedné z tříd nebo případně do té třídy, která je rizikovější [63]. Hodnota k není známa dopředu a není univerzální, zjišťuje se experimentální metodou pro každou vstupní sadu zvlášť.

Výhodou metody nejbližších sousedů je jednoduchost a snadné implementování algoritmu. Metoda nevyžaduje trénování složitého modelu, kterému je těžké porozumět, nebo nekoněné vyladování parametrů, ani provádění dalších předpokladů. Algoritmus je univerzální pro řešení různorodých problémů. Mezi nevýhody, kromě již zmíněné citlivosti na odlehle hodnoty a snížení přesnosti, patří také zpomalení algoritmu s rostoucím objemem používaných dat.

5.3 Klasifikace pomocí umělých neuronových sítí

Tato kapitola bude věnována dalšímu odvětví umělé inteligence, které se široce používá pro řešení problému klasifikace velkých objemů dat - umělé neuronové sítě (Artificial Neural Network, ANN). ANN je výpočetní model inspirovaný nervovou soustavou člověka tvořenou neurony, které jsou zodpovědné za příjem, zpracování a přenos signálů potřebných pro funkci organismu. Jak je již patrné z názvu, základní jednotkou matematického modelu neuronových sítí je umělý (formální) neuron neboli také perceptron [21]. Jednotlivé neurony

jsou navzájem propojené tak, aby výstup jednoho neuronu byl vstupem do dalších - tedy jeden neuron může mít více vstupů, ale jenom jeden výstup. Na obr. 19 je znázorněna struktura formálního neuronu, kde $\mathbf{x} = (x_1, \dots, x_n)$ je vektor vstupů a $\mathbf{w} = (w_1, \dots, w_n)$ je vektor synaptických vah, kterými se násobí jednotlivé vstupy.



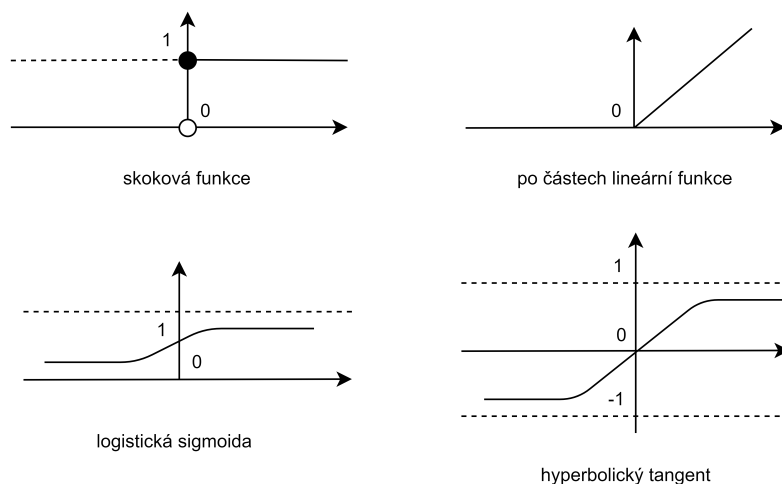
Obrázek 19: Struktura umělého neuronu

Velikost vah w_i vyjadřuje uložení zkušeností do neuronu. Čím je hodnota vyšší, tím je daný vstup důležitější. Uvnitř neuronu se následně provede skalární součin vstupu a vah. Výsledek se transformuje předem definovanou přenosovou (aktivační) funkcí a pošle se na výstup y . Tento proces lze matematicky popsat vzorcem:

$$y = f\left(\sum_{i=1}^n w_i x_i - \theta\right) \quad (15)$$

kde θ je práh a f je přenosová funkce. Přenosová (aktivační) funkce se může lišit v závislosti na typu neuronu, vrstvy, nebo celé neuronové sítě. Mezi nejvíce používané funkce patří:

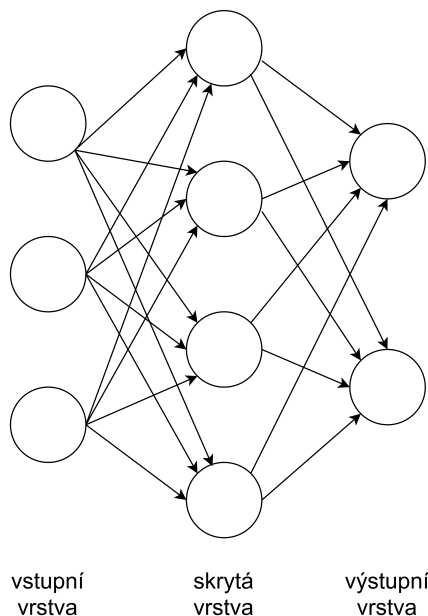
- **Skoková funkce** - pro tuto funkci platí, že pokud je vstup menší než práh, pak $f(x) = 0$. Pokud je naopak větší než práh, pak $f(x) = 1$. Používá se hlavně pro binární klasifikátory. Její derivace je ale nulová.
- **Po částech lineární funkce (ReLU)** [86] ve tvaru $f(x) = \max(0, x)$ převádí záporné hodnoty x na 0. Používá se pro svou nízkou výpočetní náročnost a jednoduchou derivaci.
- **Sigmoidální funkce (logistická)** má tvar $f(x) = \frac{1}{1+e^{-x}}$. Nabývá hodnot od 0 v minus nekonečnu do 1 v plus nekonečnu. Její výhodou oproti skokové funkci je, že má ve všech bodech spojitě derivace. Průměrnou hodnotou této funkce je 0,5. Nejčastěji se používá pro vícevrstvé neuronové sítě.
- **Funkce hyperbolické tangenty** má tvar $f(x) = \frac{2}{1+e^{-x}} - 1$. Tvarem je velmi podobná sigmoidální funkci, ale x nabývá hodnot $(-1, 1)$. Průměrnou hodnotou této funkce je 0. Výhodou funkce oproti ostatním je její jednoduchost v aproximaci a vhodnost z hlediska normalizace.



Obrázek 20: Typy aktivačních funkcí

5.3.1 Vícevrstvé neuronové sítě

Pro případ binární klasifikace si většinou vystačíme s jedním perceptronem, ale pro řešení komplexnějších problémů je potřeba použít **vícevrstvou neuronovou síť** (MLP – Multi Layer Perceptron). Počet neuronů a jejich vzájemné propojení v síti určuje architekturu (topologii) neuronové sítě. Z hlediska využití rozlišujeme v síti **vstupní**, **pracovní** (skryté, mezilehlé, vnitřní) a **výstupní** neurony [76]. Vstupní vrstva slouží výhradně k přenosu informace do první skryté vrstvy a nevykonává žádnou výpočetní funkci. Počet skrytých vrstev závisí na složitosti řešeného problému a rozsáhlosti vstupního souboru dat.



Obrázek 21: Vícevrstvá neuronová síť

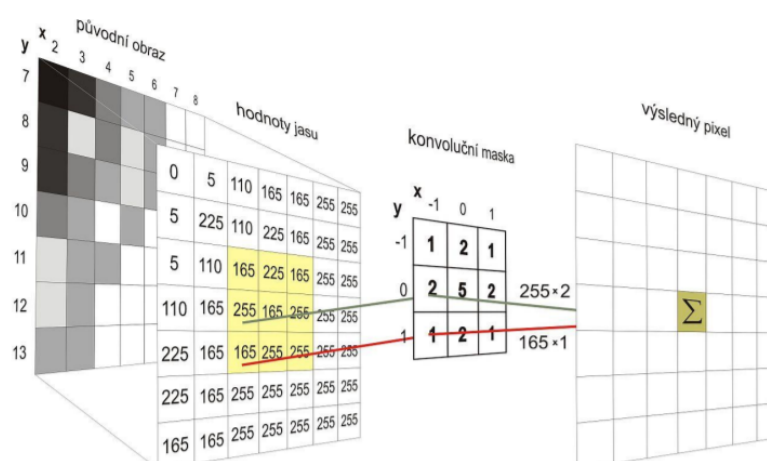
Podle typu architektury se vícevrstvé sítě se dělí na dva typy: **dopředné** a **zpětnova-zebné**.

Dopředná (acyklická) neuronová síť je nejpoužívanějším typem a umožňuje přenos informace jenom jednou cestou - od vstupu k výstupu. Uspořádání neuronů takové sítě je znázorněno na obr. 21. Tato architektura neobsahuje žádnou zpětnou vazbu, tzn. že výstup jakékoli vrstvy má vliv jenom na další vrstvu, nikoliv na tu stejnou. Tento typ sítě se široce používá pro řešení úkolů detekce [56, 74], klasifikace [26, 82], rozpoznávání vzorů [5] atd.

Ve **zpětnovazebné (rekurentní) neuronové síti** se informace může pohybovat oběma směry mezi jednotlivými vrstvami (např. od vyšších k nižším) nebo se vracet do stejného neuronu [28]. Tato zpětná vazba se realizuje prostřednictvím pomocných rekurentních neuronů ve vstupní a skryté vrstvě. Síť tohoto typu jsou dynamické - jejich *stav* se neustále mění. Tyto sítě se používají ve zpracování řeči a jiných úlohách, ve kterých vstupní data závisejí na čase.

5.3.2 Vrstvy neuronových sítí

Jak již bylo zmíněno, neuronová síť má tři základní vrstvy - vstupní, několik skrytých a výstupní. Každá skrytá vrstva ale může vykonávat odlišné funkce. V případě klasifikace snímku bývá první skrytá vrstva **konvoluční**. Aplikuje na vstupní obraz konvoluční masku (kernel) nejčastěji o rozměrech $3 \times 3 \times 3$, kde poslední číslo odpovídá hloubce masky pro tři barevné vrstvy. Jednotlivé pixely snímku se postupně násobí hodnotami masky, výsledky se sečtou a uloží do nové matice v podobě jednoho pixelu. Proces konvoluce je podrobněji znázorněn na obr. 22. Pro 3D obraz probíhá operace konvoluce na každé vrstvě zvlášť a jednotlivé výsledky se sečtou a uloží do **mapy příznaků**.



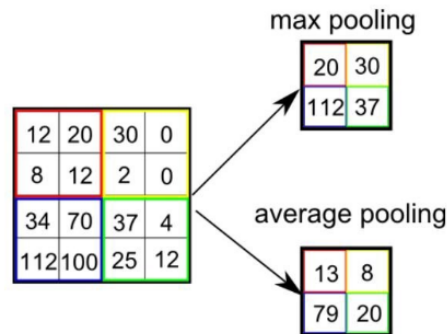
Obrázek 22: Proces konvoluční vrstvy ⁴

Cílem konvoluční vrstvy je extrahovat ze vstupního obrazu více vysokoúrovňových příznaků, které popisují sémantiku scény. Neuronová síť nemusí být omezena pouze na jednu konvoluční vrstvu. První vrstva je obvykle zodpovědná za extrakci příznaku nízké úrovně jako jsou hrany, barva, textura atd. S přidáním dalších vrstev síť rozšiřuje svoje znalosti o složitější příznaky - **hluboké učení** (deep learning) [78].

Vytvořená mapa příznaků pokračuje do sdružující vrstvy (pooling), která je zodpovědná za zmenšení její velikosti, a tak i za snížení výpočetního výkonu potřebného ke zpracování dat. Zachovává při tom ale nejdůležitější informace potřebné pro provedení předpovědi.

⁴https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=18431

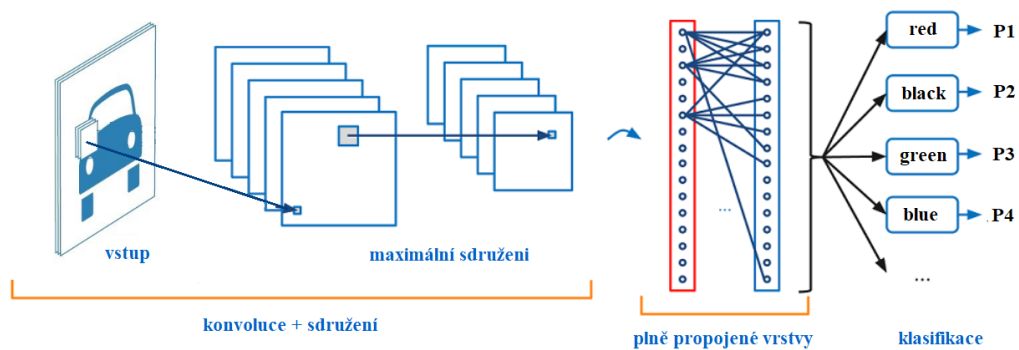
Redukce je realizována opět pomocí jádra o velikosti nejčastěji 2×2 , které se aplikuje na mapu příznaků s definovaným krokem. Existují dva hlavní typy sdružování: **maximální sdružování** (max pooling) a **průměrné sdružování** (average pooling). Maximální sdružování počítá maximální hodnotu z části obrazu pokryté jádrem. Tento typ sdružovací vrstvy se v dnešní době používá nejčastěji. Mezi jeho výhody patří, že také odstraňuje šum z obrázku a je méně výpočetně náročný. Průměrné sdružování, jak je již patrné z názvu, počítá průměrnou hodnotu všech hodnot, na které je aplikováno jádro.



Obrázek 23: Princip maximálního a průměrného sdružování [78]

V závislosti na složitosti vstupních dat může být počet vrstev zvýšen pro další zachycení detailů, avšak za cenu většího výpočetního výkonu.

Následuje **plně propojená** (fully connected) vrstva. Termín *plně propojený* znamená, že každý neuron v předchozí vrstvě je spojen s každým neuronem v další vrstvě. Tato vrstva transformuje vstupní data v jediný N-dimenzionální vektor. Tato vrstva se často používá jako poslední (klasifikační vrstva), kde N je počet tříd, do kterých chceme data rozdělit. Účelem této vrstvy je tedy použití vysokoúrovňových příznaků na výstupu předchozích vrstev pro klasifikaci obrazu (nejčastěji pomocí techniky **Softmax** [15]) do různých tříd na základě trénovacího souboru dat. Každé číslo v tomto N-dimenzionálním vektoru představuje predikovanou pravděpodobnost, se kterou vstupní snímek patří do určité třídy [13].



Obrázek 24: Ukázka neuronové sítě⁵

5.3.3 Učení neuronových sítí

Důležitou částí celého procesu vytvoření neuronové sítě je její učení (trénování). Všechny trénovací metody lze stejně jako u technik strojového učení rozdělit na tři základní skupiny:

⁵<https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

učení s učitelem, bez učitele a zpětnovazební (viz kap. 5). Je potřeba připravit trénovací a testovací sady dat včetně informace od učitele. Pro jednodušší pochopení lze učení neuronové sítě přirovnat k učení dítěte, kterému rodiče ukazují, jak vypadají jednotlivé objekty kolem něj, a pojmenovávají je. Podobným způsobem ukazujeme síti trénovací data a pomocí informace od učitele říkáme, do které třídy patří. V tomto textu bude popsán nejpopulárnější algoritmus pro učení neuronových sítí - **backpropagation** (zpětná propagace). Algoritmus zpětné propagace se používá hlavně pro trénování dopředných neuronových sítí pro učení s učitelem. Zpětnou propagaci lze rozdělit do 4 různých kroků: dopředný průchod (feed forward), vypočítání ztrátové funkce (loss function), zpětný průchod (backpropagation) a úprava parametrů (weights update). Během dopředného průchodu projde vstupní obraz všemi vrstvami neuronové sítě až na výstup. Po prvním průchodu, vzhledem k tomu, že váhy byly na začátku inicializovány náhodně, dostaneme různé pravděpodobnosti pro všechny třídy, tzn. že síť není schopna učinit předpoklad o tom, do jaké třídy patří vstupní snímek. Následuje výpočet ztrátové funkce, která vyhodnocuje odlišnost predikce od referenčních hodnot. Může být definována mnoha různými způsoby, ale nejčastěji používanou metrikou je MSE (střední kvadratická chyba):

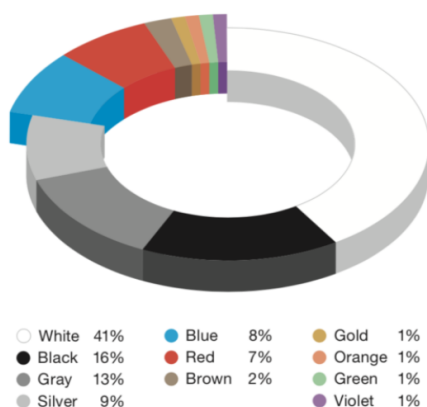
$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - y'_t)^2 \quad (16)$$

kde n je počet predikovaných hodnot, y_t je referenční hodnota a y'_t je predikovaná hodnota. Cílem dalšího kroku (zpětné propagace) algoritmu je minimalizace chybové hodnoty úpravou vah, které nejvíce přispěly ke ztrátě sítě. Pro každý parametr je vypočtena parciální derivace chyby podle váhy v konkrétní vrstvě. Dále následuje poslední krok, kterým je optimalizace parametrů. Zde se všechny váhy vypočtené zpětnou propagací aktualizují tak, aby se změnil v opačném směru gradientu ztrátové funkce.

Tento proces se opakuje pořád dokola, dokud se chybová hodnota neustálí. Ve chvíli, kdy je dokončena aktualizace parametrů, lze neuronovou síť považovat za natrénovanou a vypočítat chybu pro testovací data. Chyba na testovací množině hodnotí schopnost neuronové sítě generalizovat.

6 Implementace metod a testování účinnosti

Hlavním úkolem této diplomové práce je co nejpřesnější klasifikace barev různých dopravních prostředků - osobních aut, dodávek, autobusů, nákladních vozidel (kamionů) atd. Po prostudování různých zdrojů, které se zabývají statistickými údaji o nejprodávanějších barvách automobilů, bylo rozhodnuto omezit detekovanou barevnou škálu na 10 barev - bílá, černá, šedá, stříbrná, červená, zelená, modrá, hnědá, žlutá a oranžová. Algoritmy pro detekci barev jsou navrženy tak, aby byly schopné pracovat s libovolnými vstupními snímky bez jakékoliv předchozí znalosti.



Obrázek 25: Nejprodávanější barvy aut v roce 2018 ⁶

V rámci praktické části bylo vyzkoušeno několik používaných metod jak segmentace, tak klasifikace. Jak již bylo zmíněno v úvodu, práce je více zaměřena na klasické metody počítačového vidění. Během implementace těchto metod bylo provedeno velké množství experimentů cílených na zvýšení jejich účinnosti. Pro vyhodnocení účinnosti metod byl vytvořen rozsáhlý datový soubor z volně dostupných dopravních záznamů a streamů. Nejlepší dosažené výsledky budou uvedeny v příslušných kapitolách dalšího textu popisujících jednotlivé metody. Výsledky ostatních experimentů jsou k dispozici v příloze A této práce

6.1 Datové soubory pro vyhodnocení účinnosti algoritmů

Příprava vstupních dat je velmi časově náročným a důležitým krokem v řešení problému klasifikace barev. Velikost, kvalita a rozmanitost datové sady má přímý vliv na účinnost navrhovaného algoritmu. Jedním z cílů této práce je najít univerzální metodu, která bude schopna pracovat jak s kvalitními snímky, tak se snímky z dopravních kamer, kde obraz může být různě degradován např. vlivem počasí nebo parametrů kamery. Proto jsem připravila dvě datové sady, které pokrývají oba případy. Každou sadu jsem dále rozdělila na disjunktí skupiny trénovacích a testovacích dat.

6.1.1 Datový soubor kaggle

První datový soubor je veřejně přístupný v soutěži *kaggle*⁷[8], obsahuje velké množství snímků vozidel ve velmi dobré kvalitě. Jednou z výhod této sady je, že snímky všech vozidel byly pořízeny z různých uhlů, a proto např. algoritmy strojového učení měly možnost

⁶autobible.euro.cz/barvy-aut-roce-2018-svetu-stale-vladne-bila-evropane-jsou-sede-mysi/

⁷www.kaggle.com/c/carvana-image-masking-challenge/data

se naučit, jak může stejná barva vypadat při různém dopadu světla. Další výhodou této sady jsou anotované binární masky, díky kterým jsem měla možnost vyhodnocovat účinnost implementovaných algoritmů klasifikace bez ohledu na nepřesnosti segmentace a také vyhodnotit účinnost segmentačních metod. Nevýhodou je absence anotací barev.



Obrázek 26: Ukázka datového souboru *kaggle*

Z nabízeného množství dat jsem vybrala 1659 snímků tak, abych měla pro každou detekovanou barvu přibližně 100 snímků. Při výběru jsem se snažila pokrýt co nejširší škálu odstínů jednotlivých barev. Vzhledem k tomu, že všechny algoritmy v této práci jsou navrženy tak, aby spolupracovaly s neuronovými sítěmi pro detekci dopravních prostředků, předpokládá se, že ze snímků nebo framů videa budou extrahovány výřezy odpovídající těmto detekcím. Proto dalším krokem přípravy dat bylo jejich oříznutí tak, aby to odpovídalo předpokládanému vstupu. Oříznutí jsem uskutečnila automaticky z přiložených segmentačních masek tak, aby výsledný výřez byl co nejmenší, ale zároveň obsahoval celou masku. Příklad je znázorněn na obr. 26.

- Datový soubor - celkově 1659 snímků
- Trénovací (kalibrační) sada - 999 snímků
- Testovací sada - 655 snímků
- Podmínky - studio

Abych podvědomě nezanesla své znalosti funkcí klasifikačních algoritmů do anotací barev,

všechny obrázky z trénovací a testovací datové sady anotoval kolega. Proces anotace snímků probíhal jenom jednou pro každou datovou sadu. Součástí testovací sady tedy je referenční textový soubor obsahující jména snímků a příslušné barvy. Trénovací soubor na základě anotace byl rozdělen do jednotlivých složek odpovídajících konkrétní barvě.

6.1.2 Datový soubor traffic

Druhá sada byla vytvořena speciálně pro účely této práce z reálných dopravních záznamů a streamů⁸, které jsou volně dostupné na různých internetových stránkách a část videí jsem natáčela ručně. Jednotlivé obrázky byly extrahovány pomocí zmíněné neuronové sítě pro detekci dopravních prostředků, kterou pro účely této práce zprostředkovala firma Good-Vision⁹. Stejně jako u prvního souboru bylo potřeba z celkového množství dat na výstupu neuronové sítě vytvořit menší sady snímků pro každou z detekovaných barev. Získanou datovou sadu jsem následně rozdělila na disjunktní skupiny trénovacích a testovacích dat. Ukázka dat je na obr. 27.



Obrázek 27: Ukázka datového souboru *traffic*

- Datový soubor - celkově 1538 snímků
- Trénovací (kalibrační) sada - 975 snímků

⁸<https://www.insecam.org/en/bytag/Traffic/>

⁹<https://goodvisionlive.com/>

- Testovací sada - 563 snímků
- Podmínky - denní světlo

Anotace barev probíhala stejně jako pro datovou sadu *kaggle* (viz kap. 6.1.1).

6.2 Metrika pro vyhodnocení účinnosti algoritmů

Pro vyhodnocení účinnosti algoritmů byla zvolena metrika **accuracy** (přesnost) - počet správně klasifikovaných barev ku celkovému počtu testovacích snímků. Této metrice se také říká TOP 1 accuracy. Aby se mohlo vyhodnotit, kolik barev je detekováno správně, byl vytvořen referenční textový soubor, který obsahuje názvy jednotlivých testovacích snímků a příslušné barvy. Přesnost algoritmu je vyjádřena v procentech.

$$acc(h) = \frac{\text{správně klasifikované snímky}}{\text{všechny snímky}} \quad (17)$$

kde h je klasifikátor.

Vzhledem k tomu, že klasifikace barev dopravních prostředků je složitý problém a je obtížné odhadnout správnou barvu na první pokus kvůli vlivu velkého množství faktorů, bylo rozhodnuto rozšířit metriku accuracy o **TOP N** [49], přesněji o **TOP 2** a **TOP 3**. Klasifikátor predikuje pro každý vstup N tříd, kterým přiřadil nejvyšší skóre. Pokud se správná třída vyskytuje mezi predikovanými, pak se tato predikce považuje za správnou. V opačném případě za špatnou. Tato metrika usnadňuje klasifikaci v případech, kdy např. má auto dvě barvy, nebo kde i člověk váhá nad správnou anotací mezi 2 třídami.

6.3 Implementace segmentačních algoritmů

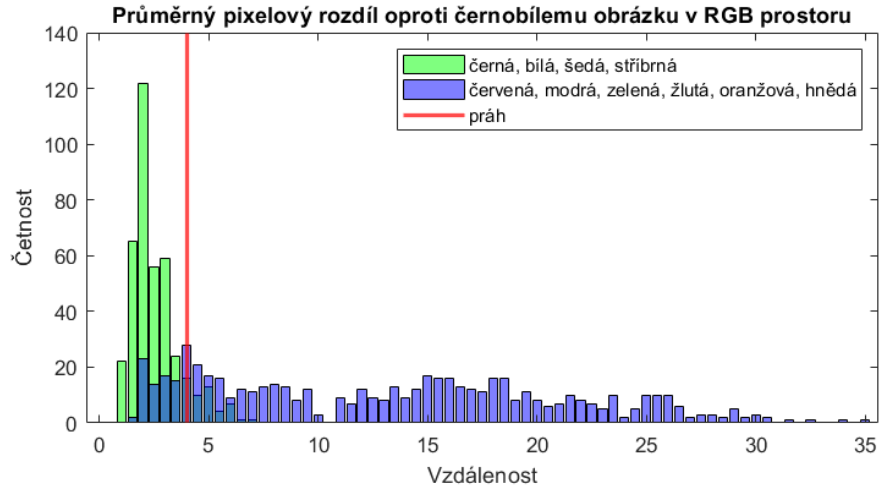
Jak již bylo zmíněno v teoretickém úvodu této práce, segmentace je jednou z důležitých součástí procesu detekce barev. Segmentační algoritmy definují oblast zájmu, ve které se určitá barva bude vyhledávat, nebo která v případě strojového učení předdefinuje informaci, ze které se následně bude systém učit. Teoreticky se dá obejít i bez segmentace [6], ale velmi často to způsobí pokles přesnosti navrhovaného algoritmu (viz kap. 6.6, 6.5).

V dalším textu budou podrobněji popsány implementace segmentačních algoritmů včetně použitých funkcí v prostředí MATLAB a ukázky výstupů. Také budou zmíněny jejich výhody a nevýhody.

6.3.1 Implementace algoritmu k-means

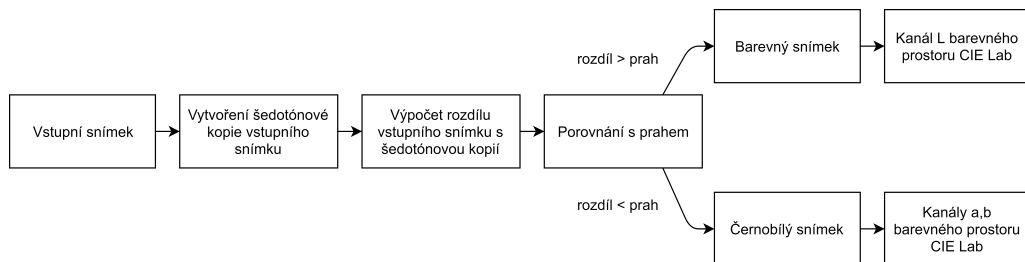
Programovací prostředí MATLAB nabízí speciální knihovnu, která zahrnuje velké množství funkcí pro zpracování obrázků. Jednou z takových funkcí pro implementaci metody k-means je `imsegkmeans(a, b)`, kde **a** je vstupní snímek a **b** je počet shluků, do kterých se rozdělují obrazové body. Po řadě experimentů s barevnými prostory RGB, Lab a HSV se ukázalo, že nejlepšími výsledky může být dosaženo reprezentací vstupních dat pomocí modelu Lab (viz kap. 3.4). Problém ale spočívá v tom, že segmentace černobílých a barevných aut se výrazně liší. Při segmentaci barevných aut vstupují do funkce `imsegkmeans` pouze 2. a 3. kanály odpovídající barevným složkám. Popředí a pozadí jsou většinou navzájem

dostatečně barevně kontrastní, aby je algoritmus mohl rozlišit, a nepotřebuje k tomu informaci o jasů. V případě černobílých aut se naopak barvy v obraze skoro nevyskytují a nás zajímá zejména informace o jasů, aby bylo možné rozlišit odstíny šedi. Vzhledem k tomu, že výsledný algoritmus pro klasifikaci barev musí být univerzální pro libovolný vstup, bylo navrženo rozšíření segmentačního algoritmu k-means o rozlišení těchto dvou případů.



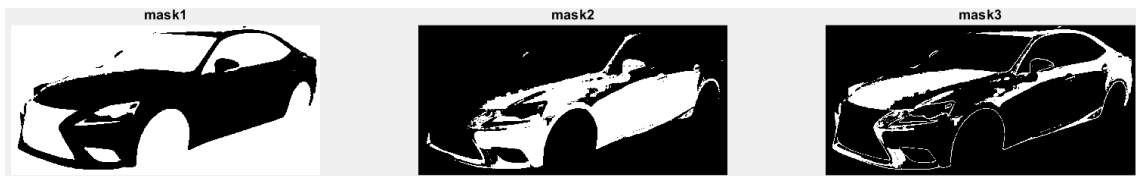
Obrázek 28: Histogramy průměrných pixelových rozdílů pro trénovací sadu snímků

Podmínkou tohoto rozšíření bylo navrhnout algoritmus, který klasifikuje černobílá a barevná auta. Za černobílé barvy považujeme černou, šedou, stříbrnou a bílou. Ostatní barvy považujeme za barevné. Vyvinutý algoritmus spočívá v nalezení takové hodnoty prahu, na základě které by se rozhodovalo o tom, zda se jedná o černobílé nebo barevné auto a které kanály mají být vstupem do algoritmu k-means. Proto byly vypočteny průměrné pixelové rozdíly všech snímků z trénovací sady oproti jejich šedotónové podobě v RGB prostoru. Následně získané hodnoty byly vykresleny ve tvaru dvou histogramů, které se v určitém bodě protínaly (viz obr. 28). Tento bod byl považován za hledanou prahovou hodnotu. Tedy pro každý vstupní snímek bude vypočtena jeho vzdálenost od jeho šedotónové varianty. Pokud tato vzdálenost bude např. menší než prahová hodnota, snímek bude považován za černobílý a do algoritmu k-means vstoupí jenom jasový kanál. Blokové schéma tohoto algoritmu je znázorněno na obr. 29.

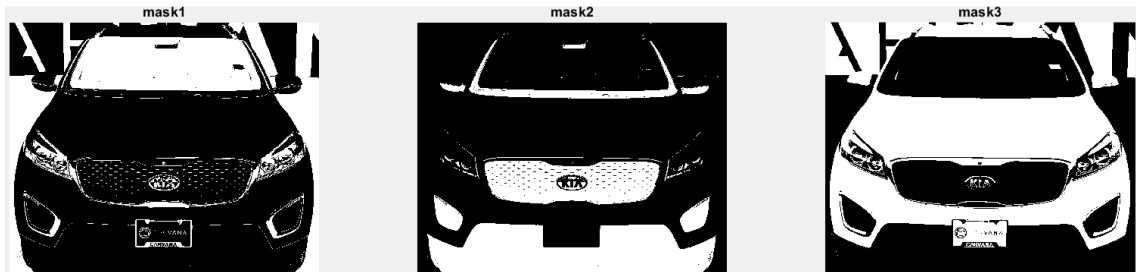


Obrázek 29: Blokové schéma algoritmu rozhodujícího o typu vstupních dat

Jako optimální bylo vybráno rozdělení obrazových bodů do třech shluků. Proč ne do dvou, když nás zajímá jenom pozadí a popředí? Experimenty ukázaly, že kvůli nerovnoměrnosti barvy auta vlivem odrazů světla je lepší rozdělovat obrazové body do více shluků a následně spojit ty, které odpovídají popředí. Výstupem funkce k-means jsou potom tři binární masky odpovídající jednotlivým skupinám pixelů (viz obr. 30 a obr. 31).



Obrázek 30: Binární masky pro barevné auto



Obrázek 31: Binární masky pro bílé auto

Dalším, neméně důležitým krokem je výběr správné masky, která bude aplikována na originální snímek a určí oblast zájmu. Na obr. 30 je vidět, že v případě barevných aut je jedna z masek negovanou kombinací ostatních dvou a zároveň je tou maskou, kterou hledáme. Tady ale nastává problém, že tato maska může pokaždé odpovídat jinému shluku. Proto se algoritmus *podívá* na oblast 20×20 pixelů v levém horním rohu úplné masky (viz obr. 32), která obsahuje všechny shluky, a vybere ten shluk, který je v této oblasti nejvíce zastoupen. Vybraná maska je prohlášena za pozadí a kombinace ostatních dvou za popředí.

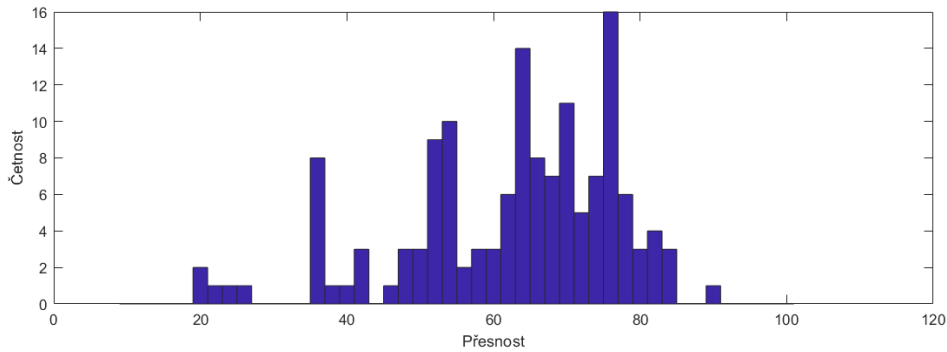


Obrázek 32: Úplná maska s vyznačenou oblastí pro výběr

Pro černobílá auta se princip vybírání segmentační masky liší, protože pozadí bude odděleno od popředí o něco hůř (viz obr. 31). V tomto případě se spíše zaměříme na hledání shluku, který odpovídá popředí. Proto se algoritmus *podívá* na definovanou oblast uprostřed úplné masky a opět vybere tu, která je zastoupena nejvíce.

Výhodou navržené metody využívající algoritmus k-means je její jednoduchost a nízká výpočetní náročnost (viz kap. 7.1). Dobře funguje pro případy, kdy popředí a pozadí je navzájem kontrastní. Naopak účinnost klesá, když na vstupním obrázku je např. šedé auto na šedé silnici, s čímž se částečně vypořádá popsáné rozšíření. Další nevýhodou je, že účinnost algoritmu závisí na správnosti odhadu binární masky. Účinnost metody byla vyhodnocena na sadě 143 snímků ze sady *kaggle*, která má k dispozici vzorové segmentační masky. Průměrná přesnost je **62%**. Může se zdát, že přesnost metody je poměrně nízká, ale z velké části je to způsobené tím, že při vyhodnocení účinnosti se získaná binární maska porovnávala s úplnou vzorovou maskou vozidla, která zahrnuje kola a skla. Pro klasifikace

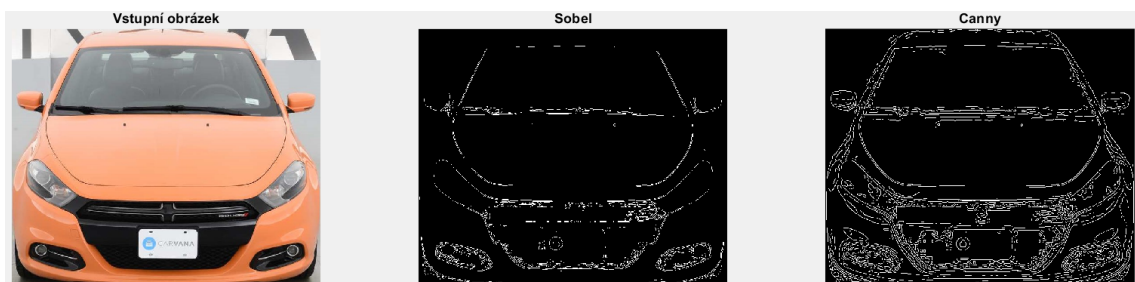
barev ale tyto prvky nejsou důležité, spíše naopak mohou negativně ovlivnit rozhodnutí algoritmu o výsledné barvě. Přesnost může být také ovlivněna chybami při vybírání segmentační masky. Na obr. 33 je znázorněn histogram přesnosti metody k-means. Z tohoto histogramu můžeme posoudit, že většina snímku testovací sady *kaggle* byla segmentována s přesností 60 až 80 procent. Výsledná přesnost metody k-means je průměrem znázorněných hodnot.



Obrázek 33: Histogram přesností segmentační metody k-means

6.3.2 Implementace algoritmu detekce hran

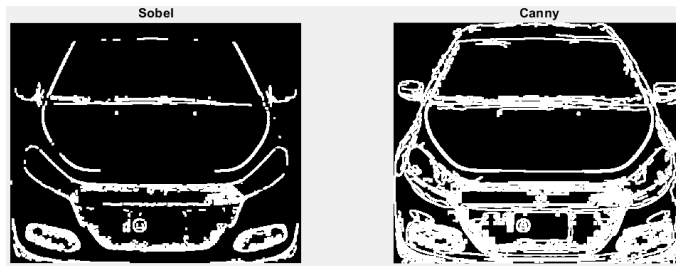
Během implementace algoritmu detekce hran jsem experimentovala s operátorem Sobela a jeho rozšířením - Cannyho detektorem, který je považován v současné době za jeden z nejlepších (viz kap. 4.1). V prostředí MATLAB k tomu slouží funkce `edge(I,method)`, kde `I` je vstupní snímek a `method` je vybraný detektor. Vzhledem k tomu, že tato funkce umí pracovat jenom s jednorozměrnými snímky, bylo potřeba najít vhodnou reprezentaci pro vstupní data. Nejlepších výsledků tak bylo dosaženo při použití barevného modelu HSV (viz kap. 3.3), ale stejně jako v případě s algoritmem k-means došlo k problému, že pro černobílá a barevná auta bylo potřeba použít odlišné vstupy. Pro barevná auta více vyhovuje druhý kanál, odpovídající sytosti, a pro černobílá třetí, odpovídající světelnosti. Jako řešení tohoto problému byl opět použit algoritmus rozhodující o typu vstupních dat na základě barevnosti vstupního snímku (viz kap. 6.3.1). Výstup funkce `edge` s použitím Sobelova a Cannyho detektoru je znázorněn na obr. 34.



Obrázek 34: Ukázka výsledků detekce hran pomocí Sobelova a Cannyho detektoru

Mapa hran ale pro segmentaci nestačí, proto je potřeba z něj pomocí dalších kroků vytvořit binární masku. Jak je patrné z obr. 34, detekované hrany jsou velmi tenké a často nejsou mezi sebou spojené tak, aby tvořily spojitý okraj objektu. Abychom s nimi mohli dále pracovat, je potřeba je nějakým způsobem rozšířit. Proto byla použita funkce MATLABu `bwmorph(I,'dilate',X)`, kde `I` je mapa hran a `X` je kladné číslo, které odpovídá míře

zvětšení hran. Tomuto procesu rozšíření se také říká dilatace. Na obrázcích výše je vidět, že Cannyho detektor dává zřetelně lepší výsledky a jeho výsledná hranová mapa perfektně opakuje linie objektu.



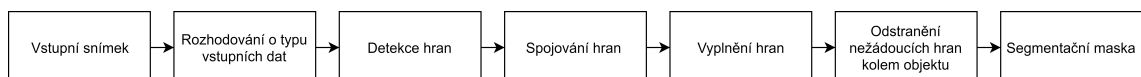
Obrázek 35: Ukázka výsledků dilatace hran

Dalším krokem algoritmu je vyplnění prostoru mezi hranami, aby se vytvořila segmentační maska. Proto byla použita funkce `imfill(I, 'holes')`, kde `I` je mapa hran po dilataci. Pokud byly v předchozím kroku hrany spojitě dobře, dostaneme na výstupu funkce správnou segmentaci jako v případě Cannyho detektoru na obr. 36.



Obrázek 36: Ukázka výsledků vyplňování hran

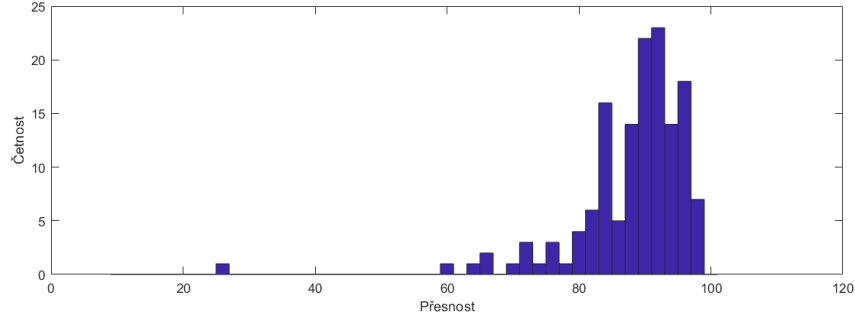
Posledním krokem je odstranění nadbytečných hran kolem objektu požitím operací eroze a dilatace. V prostředí MATLAB k tomu slouží funkce `imerode(A, se)`, kde `A` je binární maska a `se` je strukturující prvek definující oblast, na kterou budou operace aplikovány. Blokované schéma segmentační metody detekce hran je znázorněno na obr. 37.



Obrázek 37: Blokované schéma segmentační metody detekce hran

Jak je již patrné z předchozích textů a obrázků, Cannyho detektor je vhodnějším řešením pro případ segmentace. Mezi výhody této metody patří jednoduchost implementace, vysoká výpočetní rychlost (viz kap. 7.1) a průměrná úspěšnost, která dosahuje **88%** na datové sadě *kaggle*. Na rozdíl od metody k-means je úspěšnost výrazně vyšší díky odlišnému principu metody, který do segmentační masky zahrnuje všechny součásti vozidla (viz obr. 36). Dalším rozdílem je mnohem menší závislost na kontrastu mezi pozadím a popředím. Nevýhodou metody je, že účinnost metody může být negativně ovlivněna v případě neúspěšného propojení hran, což způsobí vyplnění špatné oblasti snímku.

Na obr. 38 je znázorněn histogram přesnosti metody detekce hran. Z tohoto histogramu je patrné, že většina snímku testovací sady *kaggle* byla segmentována s přesností 80 až 100 procent. Výsledná přesnost metody detekce hran je průměrem znázorněných hodnot.



Obrázek 38: Histogram přesností metody detekce hran

6.3.3 Implementace Bayesovy segmentace

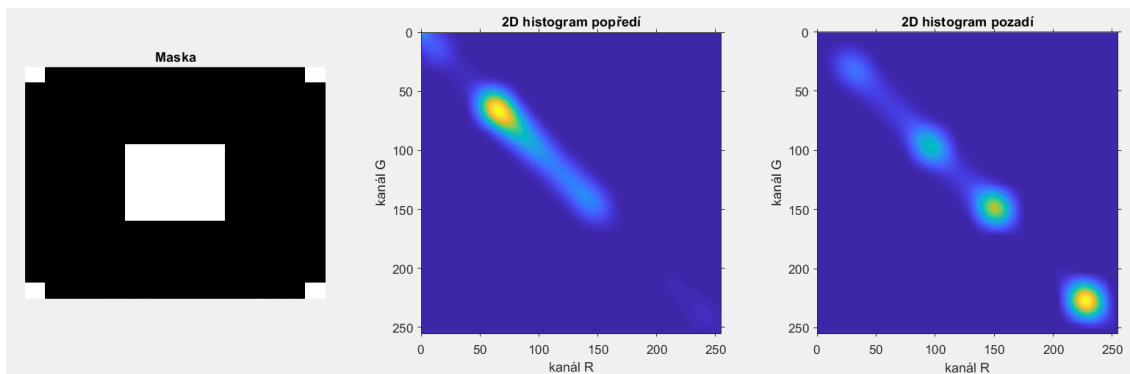
V rámci této diplomové práce byl navržen algoritmus, který byl inspirován obecným Bayesovým klasifikátorem. Metodu jsem pojmenovala podle Bayesova klasifikátoru [70], jehož principu využívám k rozdělení obrazových bodů do dvou tříd, v našem případě popředí a pozadí. Bayesův klasifikátor je založen na výpočtu aposteriorní podmíněné pravděpodobnosti zatřídění objektu (v našem případě barvy) X do třídy Y , jež se značí jako $P(Y|X)$. Při výpočtu se vychází z Bayesova vzorce:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \quad (18)$$

kde $P(X|Y)$ je podmíněná hustota pravděpodobnosti výskytu barvy X ve třídě Y , $P(Y)$ je apriorní pravděpodobnost třídy Y a $P(X)$ je celková hustota pravděpodobnosti rozložení barvy X v celém prostoru [23]. Vzhledem k tomu, že klasifikujeme jenom do dvou tříd, Y nabývá hodnot 0 (pro pozadí) a 1 (pro popředí). Pro případ segmentace jako podmíněná hustota pravděpodobnosti výskytu slouží 2D histogramy obou tříd (viz obr. 39). Vytváří se z obrazových bodů určitých oblastí vstupního snímku, které jsou definovány pomocí masky znázorněny na obr. 39. Umístění masky popředí vychází z předpokladu, že na snímcích extrahovaných z detekcí neuronové sítě se objekt vždy nachází uprostřed snímku. Odpovídajícím způsobem budou na krajích snímků pravděpodobně pixely patřící pozadí. Protože vytvářený histogram je dvourozměrný a vstupní snímek má tři barevné kanály, bylo potřeba rozhodnout, které z nich použít. Experiment ukázal, že nejlepšího výsledku dosáhneme, když je histogram tvořen kanály R a G.

Dalším a nejdůležitějším krokem po vytvoření histogramů (pravděpodobností) je klasifikace ostatních pixelů do definovaných tříd. Z matematického hlediska lze tento problém popsat pomocí vzorce (20), kde $P(barva)$ můžeme zanedbat, protože v $argmax$ na něm nezáleží. Dále předpokládáme, že $P(Y = 1) \cong P(Y = 0)$, proto zanedbáváme i $P(Y)$. Výsledný vzorec je (21). Prakticky hledáme maximální pravděpodobnost, se kterou konkrétní pixel patří do jedné ze tříd. Rozdělením všech pixelů do skupin dostaneme dvě binární masky. Masku popředí pak považujeme za segmentační a aplikujeme ji na každý z kanálů vstupního snímku.

$$C(barva) = \underset{y \in \{f, b\}}{argmax} P(Y = y | X = barva) \quad (19)$$



Obrázek 39: Ukázka vytváření 2D histogramů

$$C(\textit{barva}) = \operatorname{argmax}_{y \in \{f, b\}} \frac{P(X = \textit{barva} | Y = y) P(Y = y)}{P(X = \textit{barva})} \quad (20)$$

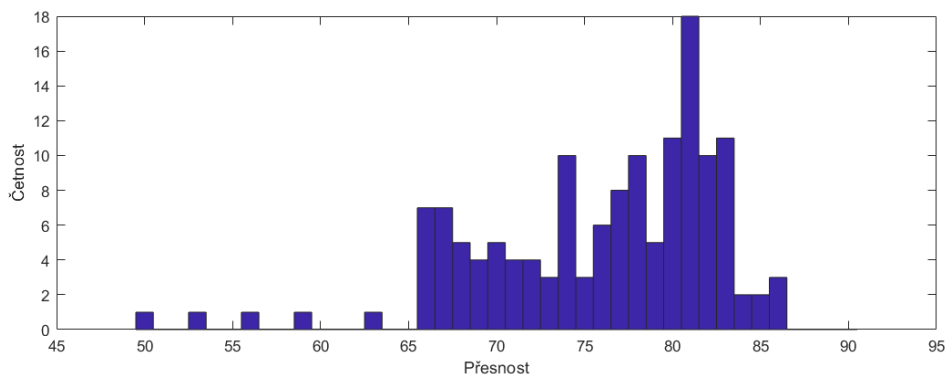
$$C(\textit{barva}) = \operatorname{argmax}_{y \in \{f, b\}} P(X = \textit{barva} | Y = y) \quad (21)$$

kde C je klasifikátor, f je popředí (foreground) a b je pozadí (background).



Obrázek 40: Ukázka výsledků Bayesovy segmentace

V porovnání s předchozími metodami k-means a detekcí hran je tato metoda o něco náročnější na implementaci. Má ale vysokou výpočetní rychlost (viz kap. 7.1) a na rozdíl od ostatních metod není závislá na kontrastu mezi objektem a pozadím. Její průměrná přesnost na množině 143 snímků ze sady *kaggle* je **76%**. Úspěšnost je opět ovlivněna tím, že získaná segmentační maska neobsahuje všechny součásti vozidla, jako jsou kola a skla. Hlavní nevýhodou metody je, že účinnost je přímo závislá na správnosti odhadu oblasti pozadí a popředí, jinak budou obrazové body rozdělené špatně.

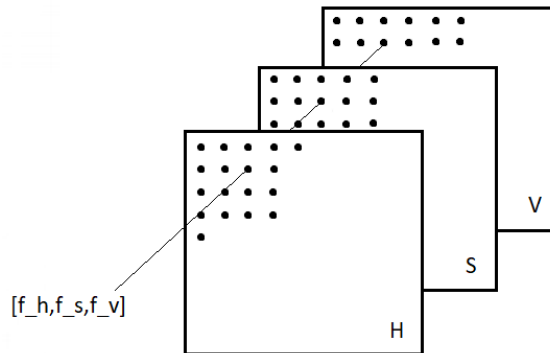


Obrázek 41: Histogram přesností Bayesovy segmentační metody

Na obr. 41 je znázorněn histogram přesnosti Bayesovy metody. Z tohoto histogramu je patrné, že většina snímků testovací sady *kaggle* byla segmentována s přesností 65 až 85 procent. Výsledná přesnost metody detekce hran je průměrem znázorněných hodnot.

6.4 Klasifikace pomocí barevných definic

Každý barevný obraz je polem $M \times N \times 3$ hodnot, které jsou zároveň informacemi o barvě. Jak už bylo zmíněno v kap. 3, barvy jsou ve většině barevných modelů definovány pomocí trojice čísel (např. r,g,b nebo h,s,v). Každé z těchto čísel udává hodnotu základních parametrů tvořících výslednou barvu. Vzhledem k tomu, že barevné prostory jsou často reprezentovány jako prostorové těleso (krychle, válec atd.), lze jednotlivé barvy a jejich odstíny definovat pomocí množiny bodů s určitými souřadnicemi uvnitř tohoto tělesa.



Obrázek 42: Složení HSV obrazu

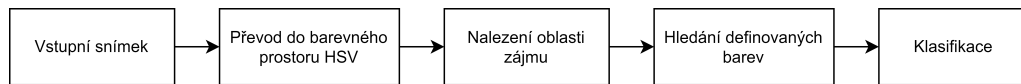
Na principu takových definic je založena tato metoda klasifikace, kde je každá barva definována pomocí logických nerovnic. Následně se barva vyhledává v určité oblasti snímku. Prvním krokem algoritmu je nalezení oblasti zájmu aplikací binární masky získané pomocí segmentace na vstupní obraz. Tento krok je velmi důležitý, protože pokud oblast zájmu nebude definována správně, účinnost metody bude téměř nulová. Dále následuje časově nejnáročnější krok - sestavování rovnic. Nejprve bylo potřeba rozhodnout, v jakém barevném prostoru budou reprezentována vstupní data. Pokud se chceme vyhnout komplikovaným nerovnicím v prostoru RGB a Lab, přirozený popis barev je v prostoru HSV. Vzhledem k tomu, že tento barevný model je reprezentován ve tvaru válce, bylo potřeba pro každou složku h , s a v určit spodní a horní hranici. Hodnoty těchto hranic byly nalezeny pomocí nástroje *colours*¹⁰, který zobrazuje barvy pro různá nastavení parametrů, a doladěny experimentální metodou. Barvy a jejich odstíny jsou tak definovány pomocí trojice čísel (h,s,v) , kde h je úhel od 0 do 360 stupňů, s a v nabývají hodnot od 0 do 1. Rovnice pro bílou barvu v modelu HSV vypadají následovně:

$$\begin{aligned}
 hue &= (hue \geq 0) \cap (hue \leq 360) \\
 saturation &= (saturation \geq 0) \cap (saturation \leq 0.4) \\
 value &= (value \geq 0.85) \cap (value \leq 1) \\
 &hue \cap saturation \cap value
 \end{aligned} \tag{22}$$

To znamená, že bílou barvu si můžeme představit jako menší válec uvnitř velkého, který obsahuje celé spektrum všech odstínů od 0 do 360 stupňů. Šířka válce od osy je hodnota

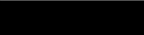

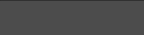








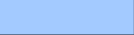








¹⁰<https://observablehq.com/@anbnyc/color-on-the-web>

sytosti - 0 až 0,4. Výška válce odpovídá hodnotě světelnosti, která pro bílou barvu a její odstíny je od 0,85 do 1. Podobným způsobem jsou definovány i ostatní barvy. Jak je vidět na obr. 5, pro černou, bílou a odstíny šedi nenesou hodnota h odpovídající barevnému odstínu prakticky žádnou informaci, proto ji můžeme zanedbat a definovat tyto barvy pomocí jen dvou hodnot. Poslední kroky algoritmu jsou: vyhledávání definovaných barev v nalezené oblasti zájmu, výpočet jejich procentuálních zastoupení v obraze a výběr výsledné barvy podle nejvyššího skóre. Na obr. 43 je znázorněno blokové schéma algoritmu.



Obrázek 43: Blokové schéma detektoru barev v HSV prostoru

Hlavní výhodou této klasifikační metody je její velmi jednoduchá implementace a výpočetní rychlost. Proces hledání hraničních parametrů pro jednotlivé barvy je ale časově velice náročný a musí se opakovat pro různé typy vstupních dat. Například parametry nalezené pro datovou sadu *kaggle* nebyly stejně účinné pro snímky ze sady *traffic*. Další nevýhodou je, že metoda není funkční bez segmentačního algoritmu a je velmi závislá na jeho účinnosti. Nejvyšší dosažená přesnost této metody je **69%** pro datovou sadu *kaggle* v kombinaci se segmentačním algoritmem detekce hran.

Barva	Hue [°]	Saturation	Value	Min	Max
Černá	0-360	0-0.4	0-0.3		
Šedá	0-360	0-0.3	0.3-0.6		
Stříbrná	0-360	0-0.2	0.6-0.85		
Bílá	0-360	0-0.4	0.85-1		
Červená	0-10, 330-360	0.1-1	0.1-1		
Modrá	170-300	0.1-1	0-1		
Zelená	60-170	0.05-1	0.1-1		
Žlutá	40-65	0.2-1	0.75-1		
Oranžová	10-40	0.1-1	0.75-1		
Hnědá	0-30, 320-360	0.1-0.5	0.1-1		

Tabulka 1: Parametry klasifikační metody barevných definic nalezené pro datovou sadu *kaggle* a příklady vyskytujících se barev.

6.4.1 TOP 2 a TOP 3 accuracy

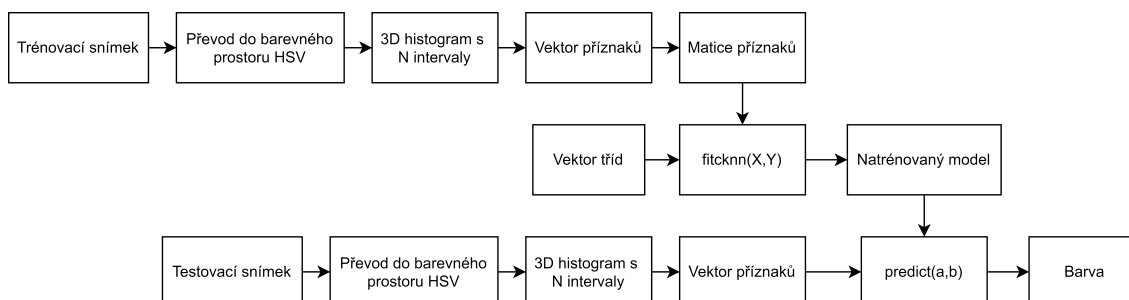
Vyhodnocení účinnosti dané metody bylo rozšířeno o metriku TOP N (viz kap. 6.2). Tato metrika je velmi nápomocna v případech, kdy musí klasifikátor rozhodovat mezi dvěma třídami, jejichž skóre se liší velmi málo, a nakonec vybere tu špatnou. Podobná situace může nastat, když se např. sluneční světlo odráží od nalakovaného povrchu černého auta, jehož barva se pak může jevit jako bílá nebo šedá. Maximální přesnosti TOP 2 pro datovou sadu *kaggle* bylo dosaženo při použití segmentačního algoritmu detekce hran, jejíž hodnota dosahuje **84%** (zvýšení **15%** oproti TOP 1). Metrika TOP 3 ale překvapivě nabyla své maximální hodnoty **91%** (zvýšení až **24%** oproti TOP 1) při použití Bayesovy segmentační metody. Výsledky pro ostatní segmentační metody jsou znázorněny v tab. 2.

Segmentace/ přesnost	Detekce hran	Bayesova	K-means
TOP 1	69%	67%	65%
TOP 2	84%	83%	81%
TOP 3	90%	91%	85%

Tabulka 2: Přehled výsledků klasifikační metody barevných definic

6.5 Implementace algoritmu nejbližších sousedů

Pro implementaci klasifikátoru nejbližších sousedů nabízí programovací prostředí MATLAB funkci `fitcknn(X,Y)` s dalšími proměnnými `NumNeighbors` - počet sousedů a `Distance` - vzdálenost, které, jak již bylo zmíněno v kap. 4.2, mají největší vliv na účinnost metody. Jako vstupní data byly použity sady 100 snímků pro každou z vybraných barev (viz kap. 6.1). Následně byly snímky převedeny do barevného prostoru HSV (viz kap. 3.3). Pro vytvoření vektoru příznaků se spočítá 3D histogram s omezeným počtem intervalů (binů), ze kterého vznikne pomocí zig-zag načítání hodnot výsledný vektor. Stejným způsobem se počítají vektory příznaků pro všechny snímky trénovací sady a ukládají se do jedné matice, kterou považujeme za vzorový model. Tato matice je prvním vstupem X do funkce `fitcknn`. Druhým vstupem Y je vektor tříd, kde každá pozice odpovídá skutečné barvě pro jednotlivé vektory příznaků, podle kterých se bude algoritmus rozhodovat. V případě, že se nepoužije žádná jiná proměnná, hledá algoritmus automaticky jednoho souseda pomocí Euklidovské vzdálenosti. Výsledek získáme pomocí funkce `predict(a,b)`, kde a je natrénovaný model a b je vektor příznaků testovacího snímku.



Obrázek 44: Blokové schéma metody nejbližších sousedů

Pro dosažení co nejlepších výsledků jsem experimentovala s různými metodami výpočtu vzdálenosti - Euklidovská, Manhattanská a kosinová (viz kap. 5.2). Také jsem měnila počet hledaných sousedů (1 až 4) a počet intervalů histogramů. S ohledem na předchozí výsledky [45] byly vybrány následující kombinace intervalů pro jednotlivé kanály H, S a V: 4,4,4; 8,4,4; 8,8,8; 16,8,8 a 16,16,16. Experimentovala jsem také se vstupními daty - s použitím segmentace i bez ní. Výsledky všech experimentů jsou uvedeny v příloze A.2.

Pro testovací sadu *kaggle*, jejíž snímky nebyly segmentované, bylo nejvyšší přesnosti **79%** dosaženo v kombinaci s výpočtem Euklidovské vzdálenosti, třemi sousedy a osmi intervaly histogramu pro každý z kanálů H, S a V. V případě segmentovaných dat pro tuto sadu ukázal experiment, že počet hledaných sousedů a vybraná metrika nemá výrazný vliv na výslednou přesnost. Výsledky se lišily jenom při použití různých kombinací intervalů histogramů. Nejvyšší účinnost je rovna **82%**.

Pro testovací sadu *traffic* bez segmentace dosáhla nejvyšší přesnost hodnoty **57%**. Tato hodnota se opakovala při výpočtu Manhattanské vzdálenosti pro jednoho a tři hledané sousedy. Počet intervalů histogramů byl také stejný pro oba případy a byl roven 16 pro každý z kanálů H, S a V. Ze všech segmentačních metod se jako nejlepší ukázala Bayesova metoda v kombinaci se třemi sousedy, opět Manhattanskou vzdáleností a kombinací 16,8,8 intervalů histogramu - **69%**.

Nesegmentovaná data					Segmentovaná data			
Počet sousedů/ počet intervalů	1	2	3	4	1	2	3	4
4, 4, 4	76%	77%	77%	77%	78%	75%	77%	76%
8, 4, 4	77%	77%	78%	78%	79%	80%	78%	79%
8, 8, 8	78%	77%	79%	78%	82%	82%	82%	82%
16, 8, 8	73%	71%	72%	72%	78%	77%	76%	77%

Tabulka 3: Nejlepší dosažené výsledky pro datový soubor *kaggle*

Nesegmentovaná data					Segmentovaná data			
Počet sousedů/ počet intervalů	1	2	3	4	1	2	3	4
4, 4, 4	55%	50%	50%	51%	62%	64%	63%	64%
8, 4, 4	54%	51%	53%	54%	66%	66%	67%	67%
8, 8, 8	52%	52%	56%	55%	62%	64%	63%	64%
16, 8, 8	52%	50%	54%	54%	67%	65%	69%	69%
16, 16, 16	57%	57%	57%	56%	67%	67%	67%	67%

Tabulka 4: Nejlepší dosažené výsledky pro datový soubor *traffic*

6.6 Implementace metody podpůrných vektorů (SVM)

Metodu podpůrných vektorů jsem v programovacím prostředí MATLAB realizovala pomocí funkce `fitcecoc(X,Y)`, kde X je matice příznaků a Y informace od učitele. Automaticky se používá technika One vs One (viz kap. 5.1), ale pomocí proměnné `Coding` ji lze v případě potřeby změnit na One vs All nebo jiné. Výsledky pro testovací data se vyhodnocují funkcí `predict(a,b)`, kde a je natrénovaný model a b je vektor příznaků.

Při ladění modelu bylo provedeno velké množství experimentů se vstupními daty. Zkoušela jsem různé velikosti trénovacích sad, barevné modely, způsoby vytváření vektorů příznaků a kombinace intervalů histogramů. Bohužel změny velikosti trénovacích sad neposkytovaly očekávané výsledky tzn. že čím větší byla sada, tím horší byla přesnost, což neodpovídá logice strojového učení. Bylo to způsobeno pravděpodobně tím, že jednotlivé skupiny snímků nebyly vyvážené. Například podsada s bílými auty obsahovala 400 snímků, a podsada s oranžovými jenom 96. Je to proto, že některé barvy vozidel jsou méně používané a vyskytují se velmi málo (viz obr. 26). Proto jsem se rozhodla omezit trénovací sadu na 100 snímků pro každou barvu. Dalším experimentem byl výběr barevného prostoru mezi RGB a HSV (viz kap. 3.2 a 3.3). Výsledná přesnost ukázala, že vhodnější je prostor HSV. Nejlepší dosažené přesnosti pro datové sady *kaggle* a *traffic* jsou znázorněny v tab.5. Výsledky všech experimentů jsou uvedeny v příloze A.1.

Vektory příznaků se vytvářely dvěma způsoby. První z nich již byl popsán v předchozí podkapitole, a to výpočet pomocí 3D histogramu. Princip vytvoření 3D histogramu je po-

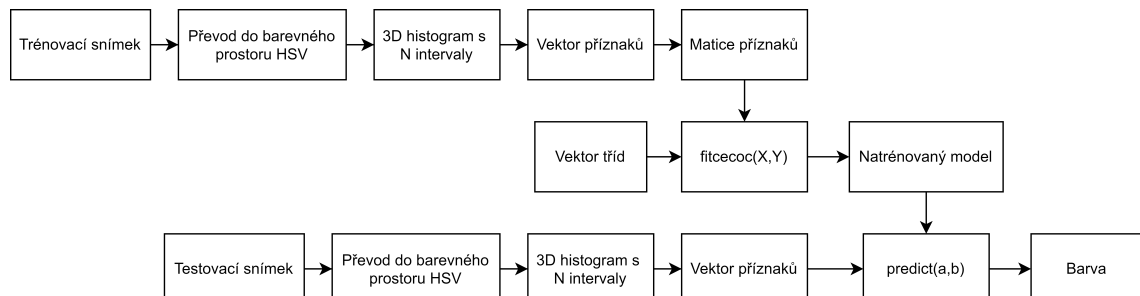
Datový soubor kaggle		
Počet intervalů	Nesegmentovaná data	Segmentovaná data
4, 4, 4	80%	79%
8, 4, 4	82%	82%
8, 8, 8	72%	83%
16, 8, 8	69%	75%
Datový soubor traffic		
4, 4, 4	57%	62%
8, 4, 4	63%	67%
8, 8, 8	61%	66
16, 8, 8	65%	70%
16, 16, 16	60%	69%

Tabulka 5: Nejlepší dosažené výsledky pro metodu podpůrných vektorů

drobně popsán zde [45]. Druhý způsob je výpočetně méně náročný a je založen na výpočtu 2D histogramů pomocí funkce `imhist` pro jednotlivé kanály, které jsou následně poskládány za sebe do jednoho vektoru. Posledním z experimentů bylo omezení histogramu na určitý počet binů (intervalů). Proč vůbec histogram omezovat, když může dojít ke ztrátě informace? Jedním z mých prvních pokusů bylo, že jako vektor příznaků vstupního obrázku jsem zvolila úplný vektor vytvořený z histogramů a při vyhodnocení účinnosti takto natrénovaného modelu jsem narazila na problém přeučení (overfitting) systému (viz kap. 5). Proto jsem se rozhodla experimentovat také s různými kombinacemi počtů intervalů jako 4,4,4; 8,4,4; 8,8,8 a 16,8,8. Blokové schéma výsledného algoritmu je znázorněno na obr. 45.

Nejvyšší přesnost **82%** pro nesegmentovaná data souboru *kaggle* má model, jehož vstupní data jsou v barevném prostoru HSV, vektory příznaků jsou vytvořené z 3D histogramů a počet intervalů těchto histogramů je 8,8,8 pro jednotlivé kanály. Pro segmentovaná data je nejvyšší dosažená přesnost **83%** s použitím Bayesovy segmentační metody.

Při aplikaci metody podpůrných vektorů na snímky ze souboru *traffic* dosáhla nejvyšší přesnost pro nesegmentovaná data hodnoty **65%**. Vektory příznaků byly vytvořeny z 3D histogramů s velikostí intervalů 16,8,8 pro kanály H, S a V. Data segmentovaná Bayesovou metodou ukázala výsledek **70%**. Způsob tvoření vektorů příznaků a velikosti intervalů zůstal nezměněn.



Obrázek 45: Blokové schéma metody podpůrných vektorů

6.6.1 TOP 2 a TOP 3 accuracy

Metoda podpůrných vektorů nabízí možnost rozšířit vyhodnocení její účinností o metriku TOP N (viz kap. 6.2), jejíž výsledky jsou pro obě datové sady znázorněny v tab. 6 a tab. 9.

TOP 2 accuracy		
Počet intervalů	Nesegmentovaná data	Segmentovaná data
4, 4, 4	88%	89%
8, 4, 4	90%	90%
8, 8, 8	90%	94%
16, 8, 8	88%	93%
TOP 3 accuracy		
4, 4, 4	91%	91%
8, 4, 4	94%	94%
8, 8, 8	93%	96%
16, 8, 8	92%	94%

Tabulka 6: TOP N výsledky pro datový soubor *kaggle*

Výpočet TOP 2 accuracy pro nesegmentovaná data souboru *kaggle* zvýšil již dosaženou přesnost o **8%**, což ve výsledku dává **90%**. Tento výsledek se opakoval pro kombinace velikostí intervalů 8,4,4 a 8,8,8. Pro segmentovaná data bylo maximální navýšení přesnosti až **11%** pomocí Bayesovy metody, což celkově činí **94%**. Nejúčinnější velikost intervalů histogramů byla opět 8,8,8 pro jednotlivé kanály H, S a V.

Výsledky metriky TOP 3 accuracy se velmi blíží ke stoprocentní přesnosti. Pro nesegmentovaná data činí tato hodnota **94%** pro kombinaci intervalů 8,4,4, to je o **12%** vyšší než TOP 1. V případě segmentovaných dat sahá přesnost až k **96%** při použití Bayesovy segmentační metody a osmi intervalů histogramů pro kanály H, S a V.

TOP 2 accuracy		
Počet intervalů	Nesegmentovaná data	Segmentovaná data
4, 4, 4	71%	81%
8, 4, 4	76%	82%
8, 8, 8	76%	80%
16, 8, 8	79%	80%
16, 16, 16	78%	82%
TOP 3 accuracy		
4, 4, 4	81%	91%
8, 4, 4	86%	90%
8, 8, 8	85%	88%
16, 8, 8	85%	86%
16, 16, 16	87%	87%

Tabulka 7: TOP N výsledky pro datový soubor *traffic*

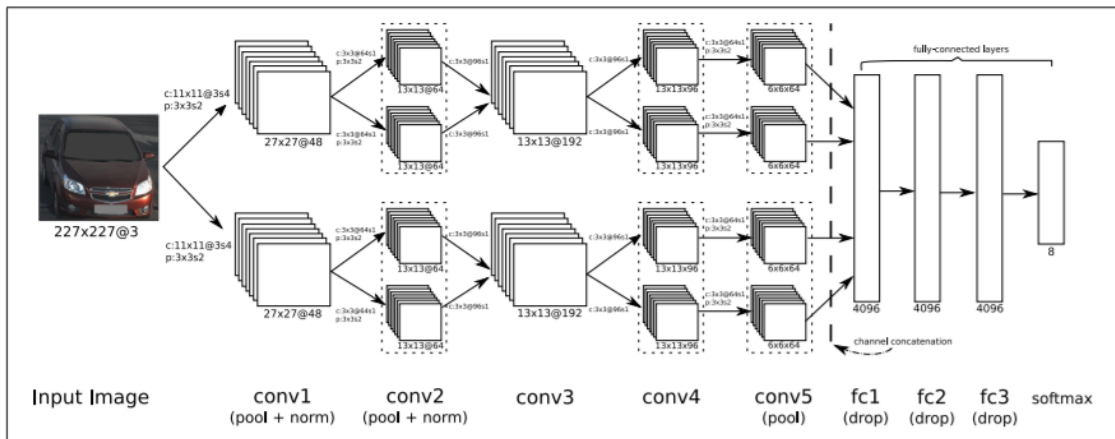
V případě souboru *traffic* také metrika TOP N výrazně zvýšila výslednou přesnost. Pro nesegmentovaná data činí rozdíl mezi přesnostmi TOP 1 a TOP 2 až **14%**, celková přesnost je tedy **79%**. Stejně tak se přesnost pro data segmentovaná Bayesovou metodou zvýšila

o **12%**, což dává **82%**. Tento výsledek se opakoval pro velikosti intervalů 3D histogramů 8,4,4 a 16,16,16.

Přesnosti dosažené pomocí metriky TOP 3 se opět blíží ke sto procentům. Hodnota přesnosti algoritmu podpůrných vektorů bez použití segmentace dosáhla **87%**, což je o **22%** vyšší než TOP 1. Pro algoritmus se segmentací činí tento rozdíl **21%** a výsledná přesnost se pak rovná **91%**. V předchozím textu a tabulkách jsou znázorněny jen nejlepší dosažené výsledky pro danou metodu. Tabulky výsledků ostatních experimentů jsou uvedeny v příloze A.1 této práce.

6.7 Klasifikace barev pomocí konvoluční neuronové sítě

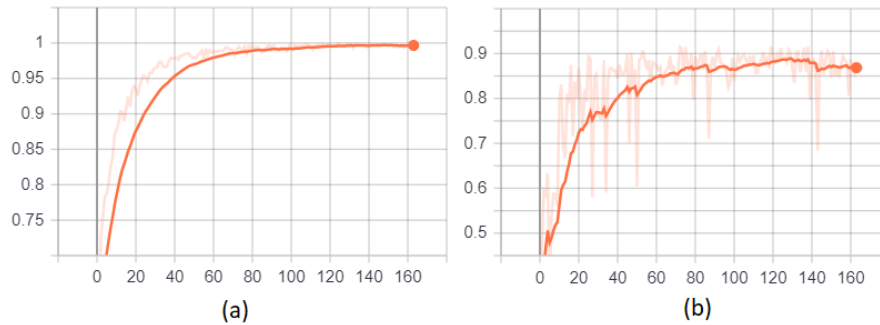
Vzhledem k tomu, že se klasifikace barev objektů v dnešní době běžně řeší pomocí neuronových sítí, nesmí zmínka o nich v této práci chybět. Jak již bylo řečeno v úvodu, cílem mé diplomové práce je najít účinnou metodu klasifikace barev, která bude následně spolupracovat s neuronovou sítí pro detekci a sledování (tracking) dopravních prostředků. Z implementačních a výpočetních důvodů bohužel dvě neuronové sítě nemohou běžet současně, proto se hledá řešení pomocí klasických metod počítačového vidění a neuronové sítě jsou v tomto textu popsány jen pro úplnost porozumění dané problematice. Po nastudování odborné literatury a dostupných implementací jsem se rozhodla pro konvoluční neuronovou sít' [73, 6] (implementace vlastní architektury je nad rámec zadání). Architektura této neuronové sítě obsahuje dvě základní sítě s osmi vrstvami pro každou (viz obr. 46). První až pátá vrstva jsou konvoluční a liší se hlavně tím, že některé obsahují navíc procesy normalizace a maximálního sdružování. Jako aktivizační funkce byla použita po částech lineární funkce ReLU. Poslední tři vrstvy jsou plně propojené a klasifikace probíhá pomocí Softmax aktivace [15]. Oficiální implementace [6] této neuronové sítě je realizována v programovacím jazyce Python v knihovně keras [32].



Obrázek 46: Architektura konvoluční neuronové sítě [73]

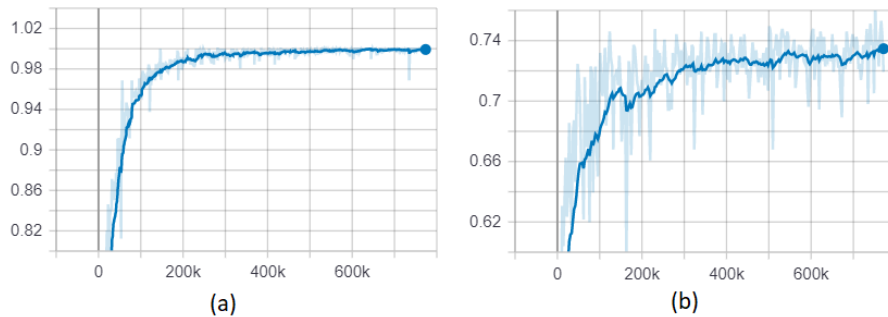
Sít' jsem trénovala po 160 tisíc kroců pro datovou sadu *kaggle* a 770 tisíc kroců pro datovou sadu *traffic* (viz kap. 6.1). Za krok se považuje dopředný průchod, zpětný průchod a aktualizace vah. Součástí tohoto procesu je také předzpracování vstupních dat změnou rozměrů na 244 × 244 pixelů. Počet vstupních snímků se uměle navyšuje vytvořením horizontálně zrcadlené kopie jednotlivých snímků - čím rozsáhlejší je trénovací sada, tím vyšší bude přesnost. Do vrstev neuronové sítě potom vstupují sady o 32 snímcích. Hyperparametr learning rate je roven 0,001. Tento parametr se během procesu učení zmenšuje s exponen-

ciálním poklesem vah (angl. decay) 10^{-6} . Autoři článků [73] experimentovali s různými barevnými prostory jako je RGB, HSV, CIE Lab a CIE XYZ a nejvyšší přesnosti dosáhli u modelu RGB, proto je tento model použit v implementaci. Je to celkem netypické, protože ve většině odborných článků se nejčastěji jako nejlepší vyskytuje model HSV.



Obrázek 47: Průběhy na (a) trénovacích a (b) testovacích datech pro sadu *kaggle*

Pro testovací sadu *kaggle*, která obsahuje celkově 660 snímků ve velmi dobré kvalitě a se stejným osvětlením (ukázka je znázorněna v kap. 6.1), byla dosažena přesnost **88%**. V případě datové sady *traffic*, která obsahuje 563 snímků z dopravních kamer, které byly pořízené při denním světle, ale jejich kvalita je ovlivněna počasím, nastaveními snímacího zařízení atd., je nejvyšší dosažena přesnost **73%**. Průběhy na trénovacích a testovacích množinách pro obě sady jsou znázorněny na obr. 47 a obr. 48.



Obrázek 48: Průběhy na (a) trénovacích a (b) testovacích datech pro sadu *traffic*

6.8 Klasifikace barev ve video záznamech

Ve všech předchozích kapitolách byla řeč jen o klasifikaci barev na statických snímcích, stejné techniky lze ale aplikovat i na video záznamy. Díky spolupráci s firmou GoodVision, která pro účely této práce zprostředkovala výstupy své neuronové sítě, jsem měla k dispozici snímky jednotlivých vozidel po dobu jejich přetrvávání ve scéně (tracku, trajektorie). Na obr. 49 je znázorněno, jak vypadají takové detekce a trajektorie aut ve scéně. Algoritmy klasifikace barev se pak aplikují na vybraný počet snímků jednoho auta a za výslednou se považuje ta barva, která byla detekována na většině z nich. Klasifikovat barvu pro všechny snímky jednoho tracku by bylo výpočetně náročné, proto jsem se rozhodla omezit počet snímků na 10 tak, aby byly rovnoměrně rozprostřené po celém tracku. Tento způsob vybírání snímku z celkového množství dává možnost zvýšit výpočetní rychlost a zároveň mít výběr rozmanitých snímků daného vozidla např. z různých stran, při odlišném osvětlení, v různé vzdálenosti atd. Jako trénovací sada byla použita sada *traffic*. Anotace barev probíhala stejně jako pro datovou sadu *kaggle* (viz kap. 6.1.1). Vyhodnocení nejčastěji

opakující se barvy pro jeden track bylo provedeno pomocí funkce MATLABu `mode(A)`. Tento způsob vyhodnocení zvyšuje pravděpodobnost správné klasifikace.



Obrázek 49: Extrakce snímků vozidel z celkové scény

Pro klasifikaci barev ve video záznamu jsem použila výše popsané algoritmy nejbližšího souseda a podpůrných vektorů. Oba algoritmy byly vyhodnocovány bez segmentace a se segmentací. Vyhodnocovalo se desetiminutové video nahrané z veřejně přístupného dopravního streamu¹¹. Tento video záznam je k dispozici v elektronické příloze.

6.8.1 Metoda podpůrných vektorů pro video záznam

Na základě předchozích zkušeností, při aplikaci metody podpůrných vektorů, byl pro vstupní data použit barevný prostor HSV a vektory příznaků se vytvářely z 3D histogramů. Experimentovala jsem jen s počty intervalů histogramů a segmentačními metodami. Výsledky experimentů jsou uvedeny v příloze A.3. Nejlepší dosažené výsledky jsou znázorněny v tab. 8. Z tabulky je patrné, že segmentace měla spíše negativní vliv na výslednou přesnost. Může to být způsobeno tím, že došlo k chybným segmentacím během procesu trénování, což ovlivnilo následnou klasifikaci na testovacích datech. Jako nejlepší se ale tentokrát ukázala segmentační metoda k-means. Nejvyšší přesnost **71%** tak byla dosažena pro nesegmentovaná data a kombinaci 16,16,16 intervalů histogramů.

Počet intervalů	Nesegmentovaná data	Segmentovaná data
4, 4, 4	63%	49%
8, 4, 4	66%	65%
8, 8, 8	65%	65%
16, 8, 8	68%	60%
16, 16, 16	71%	64%

Tabulka 8: Nejlepší dosažené výsledky metody podpůrných vektorů pro video záznam

¹¹Traffic video stream. Dostupno z: <http://85.186.25.54/mjpg/1/video.mjpg?timestamp=1586103584532>

6.8.2 TOP 2 a TOP 3 accuracy pro video záznam

Vyhodnocení metrik TOP 2 a TOP 3 pro série snímků bylo o něco náročnější vzhledem k celkovému počtu predikovaných tříd pro jednu trajektorii. Z celkového počtu predikcí je potřeba následně vybrat jen dvě nebo tři třídy v závislosti na metrice. Proto se používají teorie veřejné volby [43], kdy se každému hlasu (v našem případě predikci) přiděluje určitá váha na základě jeho preferenčního pořadí. K nejpoužívanějším závěrům teorie veřejné volby patří Bordovo hlasování [69], jež bylo použito pro vyhodnocení metrik TOP N.

U metrik TOP 2 a TOP 3 můžeme opět pozorovat řádné zvýšení hodnot přesnosti. TOP 2 přesnost pro nesegmentovaná data tak dosáhla hodnoty **81%** (zvýšení **10%** oproti TOP 1) pro netypickou kombinaci intervalů 32,16,16. Tuto kombinaci jsem nepočítala při všech experimentech, protože ve většině případů byl patrný pokles hodnot přesností s rostoucím počtem intervalů. V této situaci ale došlo k výjimce, což potvrzuje, že počet intervalů není univerzální a hledá se vždy experimentální metodou. V případě segmentovaných dat byla nejvyšší účinnost opět dosažena při použití metody k-means a je **85%** (zvýšení až **20%** oproti TOP 1). Výsledky experimentů se segmentačními metodami jsou uvedeny v příloze A.3.

TOP 3 accuracy pro nesegmentovaná data má hodnotu **89%** (zvýšení **18%** oproti TOP 1) pro kombinaci intervalů 16,16,16. Nejvyšší dosažená přesnost pro segmentovaná data je **92%** (zvýšení až **27%** oproti TOP 1) s kombinací intervalů 8,4,4.

TOP 2 accuracy		
Počet intervalů	Nesegmentovaná data	Segmentovaná data
4, 4, 4	69%	66%
8, 4, 4	70%	85%
8, 8, 8	73%	83%
16, 8, 8	79%	81%
16, 16, 16	79%	81%
32, 16, 16	81%	-
TOP 3 accuracy		
4, 4, 4	81%	80%
8, 4, 4	81%	92%
8, 8, 8	84%	90%
16, 8, 8	87%	90%
16, 16, 16	89%	89%

Tabulka 9: TOP N výsledky pro video záznam

6.8.3 Metoda nejblížešších sousedů pro video záznam

Jak již bylo zmíněno v kap. 6.5, tato metoda dává velký prostor pro experimenty s počtem hledaných sousedů, výpočty vzdáleností, segmentační metody atd. Barevný prostor HSV a způsob tvoření vektorů příznaků z 3D histogramu byly vybrány na základě předchozí zkušenosti s implementací dané metody. Proto jsem experimentovala jen s počty intervalu histogramu, počty hledaných sousedů a segmentačními metodami. Výsledky těchto experimentů jsou vedeny v příloze A.3.

Z tab. 10 je patrné, že nejvyšší přesnosti **77%** bylo dosaženo pro nesegmentovaná data dvakrát. Tento výsledek se opakoval při hledání jednoho a tři sousedů v kombinaci s osmi

intervaly histogramu pro každý barevný kanál. Mezi vektory příznaků se počítala Manhattanová vzdálenost.

Pro segmentovaná data Bayesovou metodou je nejvyšší přesnost **78%**. Tohoto výsledku jsem dosáhla při hledání čtyř sousedů v kombinaci s počty intervalů histogramu 8,4,4. Mezi vektory příznaků se opět počítala Manhattanová vzdálenost.

Nesegmentovaná data					Segmentovaná data			
Počet sousedů/ počet intervalů	1	2	3	4	1	2	3	4
4, 4, 4	74%	69%	76%	75%	70%	71%	71%	70%
8, 4, 4	76%	70%	75%	71%	77%	70%	77%	78%
8, 8, 8	77%	71%	77%	75%	73%	70%	77%	75%
16, 8, 8	76%	69%	76%	74%	74%	74%	76%	77%
16, 16, 16	76%	69%	76%	74%	75%	71%	76%	72%

Tabulka 10: Nejlepší dosažené metody nejbližších sousedů pro video záznam

7 Porovnání metod

Tato kapitola se věnuje podrobnějšímu porovnání implementovaných metod jak segmentace, tak klasifikace včetně grafického znázornění a příslušných tabulek výsledků.

7.1 Porovnání segmentačních metod

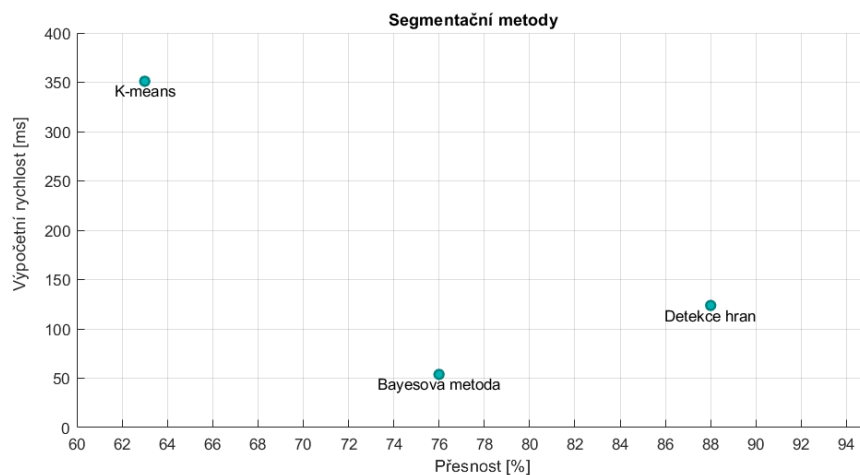
V rámci praktické části byly implementovány tři segmentační metody založené na velmi odlišných principech - k-means, detekce hran a Bayesova metoda. Metoda k-means a detekce hran využívá hlavně funkce MATLABu, které jsou součástí Image Processing toolboxu. Jsou ale rozšířeny o vlastní části algoritmu cílené na zvýšení účinnosti daných metod, např. algoritmus rozhodující mezi dvěma typy vstupních dat na základě barevnosti snímku a nalezené prahové hodnoty. Bayesova segmentační metoda byla navržena speciálně pro účely této práce a je založena na principu Bayesova klasifikátoru. Účinnost jednotlivých metod byla vyhodnocena pomocí vzorových masek, které jsou součástí datového souboru *kaggle*. Kvůli časové náročnosti anotace referenčních masek byly segmentační algoritmy vyhodnoceny pouze na jedné datové sadě. Pro množinu 143 snímků byly průměrné přesnosti **63%** při použití metody k-means, **88%** u metody detekce hran a **76%** u Bayesovy metody. Jak již bylo zmíněno v kap. 6.3.1, tyto výsledky nejsou přesné, protože získaná segmentační maska často neobsahuje všechny součásti vozidla, jako jsou např. kola a přední skla (viz obr. 50), které nenesou v případě klasifikace barev žádnou informaci a spíše mohou negativně ovlivnit výsledky. Například v odborném článku [30] dokonce navrhli speciální metodu odstranění oken pro minimalizaci tohoto negativního vlivu.



Obrázek 50: Vzorová maska souboru *kaggle* (vlevo) a Bayesova maska (vpravo)

Z důvodu nepřesností výsledků vyhodnocení segmentačních metod je mnohem důležitější to, jak jsou nápomocné při klasifikaci barev a jak moc zvyšují účinnost jednotlivých metod. V případě metody barevných definic, jejíž přesnost je velice závislá na účinnosti segmentační metody, se nejlépe projevila metoda detekce hran. Pro ostatní metody byla nejvyšší přesnost dosažena pomocí Bayesovy segmentace. Porovnáme-li výpočetní rychlost pro zmíněnou sadu 143 snímků, Bayesova metoda vypočítá segmentační masku pro jeden snímek za **53,85 ms**, metoda detekce hran za **123,78 ms** a k-means za **351,05 ms**. Na obr. 51 je znázorněna závislost přesnosti jednotlivých segmentačních metod na výpočetní rychlosti pro datovou sadu *kaggle*.

Na základě těchto údajů lze prohlásit představenou metodu využívající principu Bayesova klasifikátoru za nejlepší ze segmentačních metod zmíněných v této práci. Její výhodou oproti ostatním metodám je výpočetní rychlost, nezávislost na kontrastu mezi popředím a pozadím a také na výběru správné segmentační masky.



Obrázek 51: Závislost výpočetní rychlosti na přesnosti pro segmentační metody

7.2 Porovnání klasifikačních metod

V předchozím textu byly podrobně popsány implementace tří klasifikačních metod strojového učení - podpůrných vektorů, nejbližších sousedů a konvoluční neuronové sítě. Tyto metody se v odborné literatuře objevují nejčastěji a mají většinou velice slibné výsledky. Byla také implementována jedna metoda nevyužívající principu strojového učení - metoda barevných definic. Všechny tyto metody jsou založeny na velmi odlišných principech a každá má jak svoje výhody, tak i nevýhody.

7.2.1 Metoda barevných definic

Jako první byla v předchozím textu popsána klasifikace pomocí barevných definic. Jak již bylo zmíněno, metoda je velmi jednoduše implementovatelná a její výpočetní rychlost v kombinaci se segmentační metodou detekce hran je **95,8 ms** pro jeden obrázek. Výpočetní rychlosti jsou v tomto textu uvedeny pro procesor Intel i5-8265U pro implementaci v prostředí MATLAB. Výpočetní rychlost metody bez použití segmentačního algoritmu není v tomto textu uvedena z toho důvodu, že samostatná metoda není účinná a v práci se nepoužívala.

Hlavní nevýhodou metody je, že její parametry nejsou univerzální pro různé typy vstupních dat a proces jejich nalezení vyžaduje dlouhé experimenty. Jsou dvě možnosti hledání těchto parametrů - manuální, kde člověk manuálně hledá hranice barev podle barevného modelu colours¹² a chybové matice na trénovací množině. Druhá metoda je automatizovaná, využívající gradientní sestup [77], která se neosvědčila vzhledem k dlouhé výpočetní době. Manuální metoda je také časově náročná vzhledem k počtu hledaných parametrů. Z tohoto důvodu byla účinnost metody vyhodnocena jenom na jedné datové sadě a pro metriku TOP 1 dosahuje **69%** pro 10 barev s manuálně nalezenými parametry. Chybová matice¹³ (confusion matrix) tohoto výsledku je znázorněna na obr. 52, kde na ose X leží třídy predikované danou metodou a na ose Y správné třídy předem anotované. Čísla ležící na hlavní diagonále odpovídají počtu snímků, které byly klasifikovány správně. Ostatní čísla znázorňují chyby a přesněji počet snímků, které byly zařazeny do jiné barevné třídy. Např.

¹²<https://observablehq.com/@anbnyc/color-on-the-web>

¹³Confusion matrix. Dostupno z: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

první řádek říká, že 48 barev bylo klasifikováno správně jako černá a 16 jako šedá. Z těchto údajů lze dopočítat přesnosti klasifikace pro jednotlivé barvy zvlášť (viz tab. 11 a tab. 12).

black	48			16						
blue	9	60	1	7				1	2	
brown	18	3	49	27			3			
gray	1		1	57				8		
green	4			10	32		1	1		
orange	4		11		1	16	27	1		
red							75			
silver	4							35	20	1
white	6			2				3	53	
yellow	1		2				5			30
	black	blue	brown	gray	green	orange	red	silver	white	yellow

Obrázek 52: Chybová matice metody barevných definic

Z chybové matice je vidět, že nejvíce špatných klasifikací vzniká mezi černou, bílou, šedou a stříbrnou barvou. Je to proto, že je velmi těžké definovat určitou hranici mezi těmito barvami vzhledem k tomu, že se mohou vizuálně navzájem zaměňovat při různém osvětlení. Stejná situace je s hnědou, červenou a oranžovou barvou. Nejvyšší přesnost (nad 80%) je pro bílou, šedou a červenou barvu.

Ze všech těchto údajů lze odvodit, že tato metoda není pro řešení tak komplexního problému není úplně vhodná. Neodpovídá jednomu z hlavních požadavků kladených na hledanou metodu, a to je univerzálnost a schopnost pracovat s libovolnými vstupními daty bez žádné předchozí znalosti.

7.2.2 Metoda nejbližších sousedů

Jako druhá byla popsána metoda nejbližších sousedů. Mezi její výhody patří, že na rozdíl od jiných metod strojového učení je celkem jednoduchá pro implementaci a porozumění jejímu principu. Nevyžaduje tak velký objem a rozmanitost trénovacích dat. Proces trénování modelu, spočívající v převodu obrázků na vektory příznaků, běží na rozdíl od neuronových sítí v řádu desítek sekund na datových sadách uvedených v této práci. Nabízí široký prostor pro experimenty s výpočtem různých vzdáleností a počty hledaných sousedů. Výpočetní rychlost metody je při použití již natrénovaného modelu **46,15 ms** pro jeden nesegmentovaný snímek. Při použití např. Bayesovy segmentace se výpočetní rychlost sníží téměř dvakrát a dosahuje **88,06 ms**. Uvedená rychlost platí pro kombinaci 8,8,8 intervalů histogramu, který je použit jako vektor příznaků. Výpočetní rychlost se ale snižuje s rostoucím počtem intervalů a naopak.

Na obr. 53 je znázorněna chybová matice nejlepšího dosaženého TOP 1 výsledku metody na datové sadě *kaggle* s použitím Bayesovy segmentační metody, jehož hodnota je **82%**. Už při vizuálním porovnání s chybovou maticí předchozí metody je patrné, že počet špatných

klasifikací se výrazně snížil. Nejvíce jich je opět mezi černou, bílou a odstíny šedé. Např. 25% stříbrných aut bylo klasifikováno jako bílá a přibližně 21% šedých bylo klasifikováno jako černá. Bílá barva tentokrát byla klasifikována se stoprocentní přesností. Nejvyšších přesností (nad 80%) bylo také dosaženo pro černou, žlutou, oranžovou a hnědou barvu. Z tab. 11 je patrné, že přesnosti klasifikace většiny barev se výrazně zvýšily. Například přesnost klasifikace oranžových aut se navýšila skoro o 65% a hnědých o 48%.

black	59		1	4						
blue	6	58	3	12					1	
brown		3	97							
gray	14		9	41	1			2		
green				8	38	1		1		
orange	2				1	55	1		1	
red	13		12				50			
silver				4				40	15	1
white									64	
yellow			1			6				31
	black	blue	brown	gray	green	orange	red	silver	white	yellow

Obrázek 53: Chybová matice metody nejbližších sousedů

Do tab. 12 jsou zaneseny přesnosti klasifikace jednotlivých barev pro datovou sadu *traffic*. Maximální úspěšnost metody na této datové sadě je **69%** pro segmentovaná data. Z tabulky je vidět, že výsledky pro reálná data z dopravních kamer jsou nižší. Nejlépe (nad 80%) byla klasifikována zelená a modrá barva. Nejhuř bohužel dopadla šedá, stříbrná a hnědá. Takový pokles účinností je způsoben tím, jak bylo již několikrát zmíněno, že na klasifikaci barvy auta má vliv nespočetné množství faktorů. Pokud se bílé auto nachází ve stínu nebo černé na slunci, vjem jeho barvy se výrazně mění, což je téměř nemožné vyřešit barevnou korekcí nebo jinými algoritmy předzpracování obrazu. Přesnost se ale dá zvýšit tím, že se nebude vyhodnocovat pouze jeden snímek, ale série snímků extrahovaná z trajektorie pohybu vozidla na silnici, jak bylo popsáno v kap. 6.8. Tímto způsobem byla dosažena přesnost **78%** pro segmentovaná data. Výpočetní rychlost na datové sadě *traffic* je přibližně třikrát vyšší díky menšímu rozlišení extrahovaných snímků z dopravní scény. Pro nesegmentovaný snímek dosahuje tato hodnota **14,04 ms** a při použití např. Bayesovy segmentace je **25,28 ms**.

Na základě uvedených údajů lze posoudit, že metoda nejbližších sousedů v kombinaci se segmentačním algoritmem je dostatečně úspěšná pro řešení úkolu klasifikace barev dopravních prostředků. Splňuje všechny kladené podmínky, jako je univerzálnost pro jakákoli vstupní data a nízká výpočetní náročnost. Vylepšením účinnosti metody by mohlo být zvětšení objemu trénovací sady, což nebylo v této práci z časových důvodů možné.

7.2.3 Metoda podpůrných vektorů

Následuje metoda podpůrných vektorů. Díky knihovně *Statistics and Machine Learning* programovacího prostředí MATLAB není tato metoda implementačně náročná, přestože patří mezi složitější metody strojového učení. Princip je méně intuitivní než předchozí metody, ale široce se používá pro řešení klasifikačních problémů [51, 67]. Metoda vyžaduje pro natrénování modelu rozsáhlou sadu dat, která musí být zároveň vyvážená, tzn. že jednotlivé podsady musejí obsahovat přibližně stejný počet objektů. Podle obecných principů strojového učení se účinnost metody má zvyšovat s rostoucím počtem trénovacích dat, což se v této práci bohužel nepovedlo ověřit jednak kvůli omezenému počtu vozidel určitých barev v sadě *kaggle*, ale také kvůli nižší frekvenci výskytu barevných aut na dopravních záznamech. Výpočetní rychlost metody pro jeden nesegmentovaný snímek technikou One vs All je **44,76 ms**. Při použití např. Bayesovy segmentační metody se rychlost opět téměř dvojnásobně snižuje - **84,2 ms**, což lze ale pořád ještě považovat za dostatečně rychlou metodu.

Z chybové matice na obr. 54 nejlepšího dosaženého výsledku této metody pro datovou sadu *kaggle*, který dosahuje **83%** v kombinaci s Bayesovou segmentací, je vidět, že výsledky jsou velmi podobné metodě nejbližších sousedů. Čísla na hlavní diagonále se ale pro některé třídy mírně zvýšila. Přesnost klasifikace bílých aut opět dosáhla maximálních sta procent a celkem sedm barev z deseti má přesnost vyšší než 80% (viz tab. 11). K nejvyšší chybě došlo při klasifikaci stříbrných aut, kde až 42% snímků bylo klasifikováno jako bílá.

black	60		1	3						
blue	11	62	1	2				3	1	
brown		3	90		3		3	1		
gray	8		12	42				5		
green				4	42	1		1		
orange					1	57	1		1	
red			9			1	65			
silver				1				33	25	
white									64	
yellow			1			6				
	black	blue	brown	gray	green	orange	red	silver	white	yellow

Obrázek 54: Chybová matice metody podpůrných vektorů

Přesnosti pro jednotlivé barvy datové sady *traffic* jsou znázorněny v tab. 12, z níž je opět patrný výrazný pokles hodnot oproti sadě *kaggle*. Celková přesnost pro segmentovanou testovací sadu je **70%**. Nejlépe (nad 80%) bylo klasifikováno pět barev z deseti, a to jsou: červená, zelená, modrá, žlutá a oranžová. Skoro všechna stříbrná auta byla klasifikována jako bílá, a proto je přesnost jenom 1,5%. Může to být způsobeno nedostatkem trénovacích dat, aby byla metoda schopna generalizovat vlastnosti těchto dvou tříd. Tato teorie se potvrdila i při vyhodnocení barev ze série snímků extrahovaných z video záznamu, kde celková přesnost se, na rozdíl od metody nejbližších sousedů, skoro vůbec nezměnila a je

71% pro nesegmentovaná data. Při vyhodnocení snímků videa segmentovaných jakoukoliv ze zmíněných metod došlo k nečekanému poklesu účinnosti oproti nesegmentovaným snímkům. Výpočetní rychlost pro tuto datovou sadu je přibližně čtyřnásobně vyšší oproti sadě *kaggle*. Pro nesegmentovaný snímek dosahuje hodnota rychlosti **11,55 ms** a při použití Bayesovy segmentace **21,39 ms**.

Z údajů získaných při vyhodnocení metody podpůrných vektorů lze udělat závěr, že tato metoda je dostatečně účinná pro řešení problému klasifikace barev jak samostatně, tak v kombinaci se segmentačními algoritmy. Klasifikační přesnost na datech *traffic* byla nižší, než se očekávalo, ale může to být vyřešeno rozšířením trénovací sady dat. Jinak metoda splňuje kladené požadavky - je rychlá, univerzální pro libovolná vstupní data, běží na procesoru a má dostatečně vysokou přesnost.

7.2.4 Konvoluční neuronová síť

Poslední klasifikační metodou této práce je konvoluční neuronová síť. Vzhledem k tomu, že návrh vlastní architektury neuronové sítě překračuje rámec zadání, byla vyzkoušena jedna z veřejně přístupných implementací [6] inspirovaná zdrojem [73]. Neuronové sítě jsou v dnešní době hlavním nástrojem pro řešení klasifikačních problémů díky své vysoké přesnosti. Bohužel jsou ale výpočetně velmi náročné a vyžadují velké množství trénovacích dat, které často nejsou k dispozici. Proces trénování neuronové sítě může trvat několik hodin až několik dní, a proto jsou jakékoliv experimenty velmi časově náročné. Jak již bylo několikrát zmíněno, z implementačních a výpočetních důvodů není neuronová síť vhodným řešením pro případ klasifikace barev vozidel, které jsou detekovány jinou neuronovou sítí, proto je v této práci uvedena jen pro porovnání s klasickými metodami.

black	60	2	1							
blue	1	77	1					1		
brown	6	3	52	28			11			
gray				65				2		
green					47	1				
orange			1		1	52	4	1		
red							75			
silver				1				52	6	1
white									64	
yellow						7	1			31
	black	blue	brown	gray	green	orange	red	silver	white	yellow

Predicted class

Obrázek 55: Chybová matice konvoluční neuronové sítě

Již při vizuálním porovnání chybové matice na obr. 55 s předchozími je patrné, že počet chyb kolem hlavní diagonály je výrazně menší. Tato matice znázorňuje klasifikace pro nejvyšší dosažený výsledek datové sady *kaggle*, jehož hodnota dosahuje **88%**. Přesnosti osmi barev z deseti přesahují 80%, dvě z nich mají stoprocentní přesnost klasifikace - červená a bílá (viz tab. 11). K největší chybě došlo při klasifikaci hnědé barvy, kde bylo

28% snímků klasifikováno jako šedá a 11% jako červená.

V případě datové sady *traffic* se celková přesnost výrazně neliší od ostatních metod a dosahuje **73%** s tím, že šedá a zelená barva jsou klasifikovány s nulovou přesností. Je to nejspíše způsobeno nízkým počtem trénovacích dat. Kompenzací je ale, že až šest barev z deseti má přesností nad 80%. Pro efektivnější trénování a vyhodnocení neuronové sítě by celý proces měl běžet na grafické kartě. Vzhledem k tomu, že můj počítač nemá dostatečně výkonný hardware, využila jsem jiný počítač s grafickou kartou GeForce GTX 1080, na níž je výpočetní rychlost klasifikace jednoho snímku **15,62 ms**. Při výpočtu na procesoru Intel i7-8700K stejného počítače je rychlost klasifikace jednoho snímku **21,8 ms**. Tyto hodnoty jsou stejné pro obě datové sady, protože součástí procesu trénování i predikce je předzpracování obrazu změnou jeho rozměrů na 244×244 pixelů.

Daná konvoluční neuronová síť dosáhla nejvyšších celkových přesností jak pro datovou sadu *kaggle*, tak pro sadu *traffic* i přes to, že množství trénovacích dat bylo celkem malé. To potvrzuje tvrzení, že neuronové sítě jsou nejlepším nástrojem pro řešení problémů klasifikace, pokud k tomu máme dostatek dat a výpočetního výkonu.

7.3 Výsledky porovnání klasifikačních metod

Přehledné porovnání klasifikačních metod je zobrazeno v tabulkách 11 a 12.

Barva/Metoda	Barevné definice	KNN	SVM	CNN
Bílá	82,8%	100%	100%	100%
Černá	75%	92,19%	93,75%	95,24%
Šedá	85,1%	61,2%	62,69%	97,01%
Stříbrná	58,3%	66,67%	55%	86,67%
Červená	100%	66,67%	86,67%	100%
Zelená	66,7%	79,17%	87,5%	97,92%
Modrá	75%	72,5%	77,5%	97,47%
Žlutá	78,95%	81,59%	81,59%	79,49%
Oranžová	26,7%	91,67%	95%	88,14%
Hnědá	49%	97%	90%	52%

Tabulka 11: Přesnosti klasifikačních metod pro jednotlivé barvy datové sady *kaggle*

Barva/Metoda	KNN	SVM	CNN
Bílá	75%	78,13%	100%
Černá	78,46%	55,38%	92,31%
Šedá	40%	42,5%	0%
Stříbrná	18,2%	1,5%	53%
Červená	75,86%	90,8%	95,5%
Zelená	88,1%	95,24%	0%
Modrá	89,19%	94,59%	98,65%
Žlutá	79,03%	83,87%	88,71%
Oranžová	78,05%	90,24%	90,24%
Hnědá	40,9%	45,45%	27,27%

Tabulka 12: Přesnosti klasifikačních metod pro jednotlivé barvy datové sady *traffic*

Na základě porovnání jednotlivých metod považuji metodu podpůrných vektorů za nejlepší

pro řešení stanoveného problému. S výjimkou neuronové sítě dosahuje tato metoda nejvyšší celkové přesnosti pro obě datové sady. Její účinnost je dostatečně vysoká i bez použití segmentačních algoritmů a může se ještě zvýšit s rozšířením trénovací sady.

8 Závěr

V rámci této práce jsem se zabývala hledáním účinného řešení problému klasifikace barev dopravních prostředků na snímcích z dopravních záznamů. Požadavky kladené na hledanou metodu jsou: vysoká přesnost, malá výpočetní náročnost, výpočet na procesoru a univerzálnost pro libovolná vstupní data. Proto je práce více zaměřena na klasické metody strojového učení než na neuronové sítě, které se pro řešení těchto úkolů běžně používají. Bylo prostudováno velké množství odborných článků zabývajících se touto problematikou, jejichž přehled je obsahem kapitoly 2. Na základě tohoto studia byly vybrány a implementovány tři účinné metody segmentace obrazu a čtyři klasifikační metody. Všechny implementace jsou realizovány v programovacím prostředí MATLAB s výjimkou neuronové sítě, která je v jazyce python. Segmentační metody k-means a detekce hran byly implementovány pomocí funkcí knihovny Image Processing a rozšířeny o vlastní části algoritmu zvyšující celkovou účinnost daných metod. Pro účely této práce byla navržena Bayesova segmentační metoda. Byla inspirována principem Bayesova klasifikátoru. Pro vyhodnocení klasifikačních algoritmů byla vytvořena datová sada 1538 snímků z reálných dopravních video záznamů a streamů volně dostupných na internetových stránkách. Při implementaci klasifikačních algoritmů bylo provedeno velké množství experimentů s různými typy reprezentace vstupních dat - různé velikosti trénovacích sad, barevné prostory, kombinace intervalů histogramu a způsoby vytvoření vektoru příznaků. Klasifikační metody nejbližších sousedů a podpůrných vektorů jsou implementovány pomocí funkcí knihovny Statistics and Machine Learning. Účinnost klasifikačních metod byla vyhodnocena na dvou datových sadách, jednak na kvalitních snímcích ze studia, jednak na snímcích extrahovaných z dopravních záznamů. Vyvinutá Bayesova segmentační metoda překonala na základě podrobného porovnání ostatní zkoumané algoritmy. Za nejlepší z implementovaných klasifikačních metod byla prohlášena metoda podpůrných vektorů, jejíž nejlepším výsledkem na datové sadě *kaggle* je **83%** (96% TOP 3) a na sadě *traffic* **70%** (96% TOP 3).

Hlavním přínosem této práce je nalezení účinné metody klasifikace a návrh segmentační metody, které spolu s detekční neuronovou sítí budou součástí platformy poskytující dopravním inženýrům rychlejší a snadnější způsob zpracování dopravních dat. Znalost barev dopravních prostředků pomůže lépe sledovat jejich trajektorie pohybu ve scéně a také poskytnout informaci o typu a účelu použití.

Proces zpracování této práce byl velmi přínosný i pro mě. Rozšířila jsem svoje znalosti v oblasti zpracování a segmentace obrazu i v oblasti strojového učení, která přesahuje obor mého studia. Na základě získaných znalostí jsem navrhla vlastní segmentační metodu založenou na principu Bayesova klasifikátoru, která podle tab. 16 překonala ostatní implementované metody ve zvýšení přesnosti pro klasifikační algoritmy.

A Přílohy

Táto kapitola obsahuje výsledky všech experimentů provedených v rámci této diplomové práce. Nejlepší z dosažených výsledků jsou uvedeny v jednotlivých kapitolách (viz kap. 6).

A.1 Výsledky experimentů s metodou podpůrných vektorů

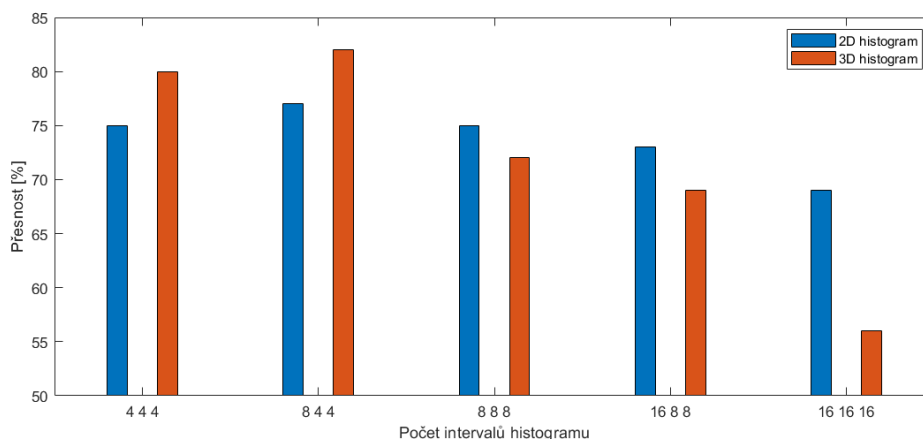
2D histogram			
Počet snímků/počet intervalů	100 snímků	200 snímků	400 snímků
4,4,4	75%	68%	57%
8,4,4	77%	72%	60%
8,8,8	75%	74%	63%
16,8,8	73%	73%	67%
16,16,16	69%	69%	66%
3D histogram			
Počet snímků/počet intervalů	100 snímků	200 snímků	400 snímků
4,4,4	80%	77%	71%
8,4,4	82%	76%	67%
8,8,8	72%	63%	61%
16,8,8	69%	64%	58%
16,16,16	56%	47%	48%

Tabulka 13: Znázornění změny přesnosti metody SVM (viz kap. 5.1) v závislosti na velikosti trénovací sady pro různé příznakové vektory (vytvořené z 2D a 3D histogramů) datové sady *kaggle* (viz kap. 6.1). Vstupní data nejsou segmentována a jsou reprezentována v barevném prostoru HSV (viz kap. 3.3).

2D histogram			
Počet snímků/počet intervalů	100 snímků	200 snímků	400 snímků
4,4,4	56%	51%	49%
8,4,4	53%	51%	48%
8,8,8	51%	49%	48%
16,8,8	49%	44%	45%
3D histogram			
Počet snímků/počet intervalů	100 snímků	200 snímků	400 snímků
4,4,4	69%	62%	62%
8,4,4	68%	60%	60%
8,8,8	73%	62%	59%
16,8,8	69%	57%	55%

Tabulka 14: Znázornění změny přesnosti metody SVM v závislosti na velikosti trénovací sady pro různé velké příznakové vektory (vytvořené z 2D a 3D histogramů) datové sady *kaggle* (viz kap. 6.1). Vstupní data nejsou segmentována a jsou reprezentována v barevném prostoru RGB (viz kap. 3.2).

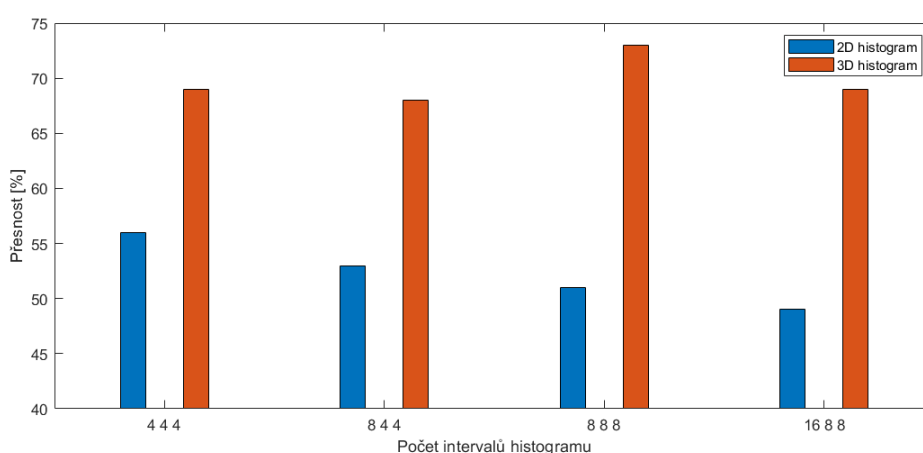
Na základě tab. 13 a tab. 14 jsem se rozhodla omezit datovou sadu *kaggle* a *traffic* na 100 snímků pro každou barvu, a vytvářet vektory příznaků pomocí 3D histogramů.



Obrázek 56: Tento graf vychází z tab. 13. Jsou do něj vyneseny přesnosti modelu SVM (viz kap. 5.1) natrénovaného na datové sadě *kaggle* obsahující 100 snímků pro každou z barev. Těchto přesností jsem dosáhla při použití příznakových vektorů vytvořených z 2D a 3D histogramů v barevném prostoru HSV.

Barevný model/počet intervalů	RGB	HSV	CIE Lab
4,4,4	69%	80%	65%
8,4,4	68%	82%	67%
8,8,8	73%	72%	71%
16,8,8	69%	69%	72%
16,16,16	60%	56%	77%

Tabulka 15: Znázornění závislosti přesnosti modelu SVM (viz kap. 5.1) na barevném modelu, ve kterém jsou reprezentována vstupní data. Na základě této tabulky jsem se rozhodla pro barevný prostor HSV při implementaci metody podpůrných vektorů a nejbližších sousedů. Uvedené přesnosti platí pro nesegmentovanou datovou sadu *kaggle* obsahující 100 snímků pro každou barvu, jejíž příznakové vektory jsou vytvořeny z 3D histogramů.



Obrázek 57: Tento graf vychází z tab. 14. Jsou do něj vyneseny přesnosti modelu SVM (viz kap. 5.1) natrénovaného na datové sadě *kaggle* obsahující 100 snímků pro každou z barev. Těchto přesností jsem dosáhla při použití příznakových vektorů vytvořených z 2D a 3D histogramů v barevném prostoru RGB.

Datová sada kaggle				
Metoda segmentace/počet intervalů	bez segm	Bayesova	k-means	detekce hran
4,4,4	80%	79%	73%	75%
8,4,4	82%	82%	76%	80%
8,8,8	72%	83%	75%	81%
16,8,8	69%	75%	68%	75%
16,16,16	56%	73%	65%	70%
Datová sada ttraffic				
Metoda segmentace/počet intervalů	bez segm	Bayesova	k-means	detekce hran
4,4,4	57%	62%	54%	57%
8,4,4	63%	67%	58%	61%
8,8,8	61%	66%	57%	60%
16,8,8	65%	70%	61%	64%
16,16,16	60%	69%	60%	64%

Tabulka 16: Znázornění přesnosti metody podpůrných vektorů (viz kap. 5.1) při použití různých segmentačních metod testovaných v této práci (viz kap. 6.3). Výsledky platí pro datovou sadu *kaggle* a *traffic* obsahující 100 snímků pro každou barvu, jejichž příznakové vektory byly vytvořeny z 3D histogramů.

Metoda segmentace/počet intervalů	bez segm	Bayesova	k-means	detekce hran
4,4,4	44%	57%	44%	51%
8,4,4	48%	61%	53%	55%
8,8,8	47%	59%	53%	55%
16,8,8	49%	63%	56%	58%
16,16,16	49%	64%	57%	58%

Tabulka 17: Znázornění přesnosti metody podpůrných vektorů (viz kap. 5.1) při použití segmentačních metod zmíněných v této práci (viz kap. 6.3). Výsledky platí pro datovou sadu *traffic* obsahující přibližně 100 snímků pro každou barvu, jejichž příznakové vektory byly vytvořeny z 2D histogramů.

Metoda segmentace/počet intervalů	bez segm	Bayesova	k-means	detekce hran
Metrika TOP2				
4,4,4	71%	81%	69%	74%
8,4,4	76%	82%	74%	75%
8,8,8	76%	80%	70%	77%
16,8,8	79%	80%	71%	77%
16,16,16	78%	82%	71%	82%
Metrika TOP3				
4,4,4	81%	91%	77%	89%
8,4,4	86%	90%	81%	89%
8,8,8	85%	88%	75%	85%
16,8,8	85%	86%	75%	86%
16,16,16	87%	87%	77%	90%

Tabulka 18: Znázornění přesnosti metrik TOP2 a TOP3 (viz kap. 6.2) pro metodu SVM (viz kap. 5.1) při použití segmentačních metod zmíněných v této diplomové práci (viz kap. 6.3). Výsledky platí pro datovou sadu *traffic*, jejíž příznakové vektory byly vytvořeny z 3D histogramů.

A.2 Výsledky experimentů s metodou nejbližších sousedů

Počet sousedů/počet intervalů	1	2	3	4
Euklidovská vzdálenost				
4,4,4	76%	77%	77%	77%
8,4,4	77%	77%	78%	78%
8,8,8	78%	77%	79%	78%
16,8,8	73%	71%	72%	72%
Manhattanská vzdálenost				
4,4,4	72%	70%	73%	73%
8,4,4	75%	73%	76%	74%
8,8,8	75%	73%	76%	75%
16,8,8	76%	65%	69%	68%
Kosinová vzdálenost				
4,4,4	76%	76%	77%	75%
8,4,4	77%	76%	76%	75%
8,8,8	75%	75%	77%	77%
16,8,8	73%	72%	72%	72%

Tabulka 19: Znázornění výsledků metody nejbližších sousedů (viz kap. 5.2) pro různý počet hledaných sousedů a různé vzdálenosti. Tyto výsledky platí pro nesegmentovaná data ze sady *kaggle* (viz kap. 6.1).

Počet sousedů/počet intervalů	1	2	3	4
Euklidovská vzdálenost				
4,4,4	78%	75%	77%	76%
8,4,4	79%	80%	78%	79%
8,8,8	82%	82%	82%	82%
16,8,8	78%	77%	76%	77%
Manhattanská vzdálenost				
4,4,4	77%	77%	78%	78%
8,4,4	80%	81%	80%	81%
8,8,8	82%	82%	82%	82%
16,8,8	80%	80%	80%	79%
Kosinová vzdálenost				
4,4,4	78%	75%	77%	76%
8,4,4	80%	79%	79%	79%
8,8,8	81%	82%	82%	82%
16,8,8	80%	79%	81%	79%

Tabulka 20: Znázornění výsledků metody nejbližších sousedů (viz kap. 5.2) pro různý počet hledaných sousedů a různé vzdálenosti. Tyto výsledky platí pro data ze sady *kaggle* segmentované Bayesovou metodou (viz kap. 6.3.3).

Počet sousedů/počet intervalů	1	2	3	4
Euklidovská vzdálenost				
4,4,4	74%	74%	75%	73%
8,4,4	75%	74%	75%	74%
8,8,8	78%	77%	78%	78%
16,8,8	75%	74%	74%	76%
16,16,16	72%	71%	74%	73%
Manhattanská vzdálenost				
4,4,4	71%	71%	72%	73%
8,4,4	74%	72%	74%	72%
8,8,8	78%	78%	77%	78%
16,8,8	78%	77%	78%	77%
16,16,16	78%	75%	76%	76%
Kosinová vzdálenost				
4,4,4	75%	75%	75%	75%
8,4,4	77%	75%	76%	75%
8,8,8	79%	79%	79%	80%
16,8,8	80%	78%	79%	79%
16,16,16	76%	75%	76%	74%

Tabulka 21: Znázornění výsledků metody nejblížešých sousedů (viz kap. 5.2) pro různý počet hledaných sousedů a různé vzdálenosti. Tyto výsledky platí pro data ze sady *kaggle* segmentované metodou detekce hran (viz kap. 6.3.2).

Počet sousedů/počet intervalů	1	2	3	4
Euklidovská vzdálenost				
4,4,4	65%	65%	66%	67%
8,4,4	70%	69%	71%	69%
8,8,8	67%	66%	66%	65%
16,8,8	62%	62%	63%	62%
16,16,16	60%	59%	60%	59%
Manhattanská vzdálenost				
4,4,4	63%	63%	64%	64%
8,4,4	65%	64%	66%	65%
8,8,8	67%	67%	67%	69%
16,8,8	67%	65%	68%	67%
16,16,16	64%	63%	65%	64%
Kosinová vzdálenost				
4,4,4	66%	65%	66%	66%
8,4,4	69%	67%	68%	68%
8,8,8	70%	70%	68%	70%
16,8,8	71%	69%	70%	70%
16,16,16	65%	66%	65%	66%

Tabulka 22: Znázornění výsledků metody nejblížešých sousedů (viz kap. 5.2) pro různý počet hledaných sousedů a různé vzdálenosti. Tyto výsledky platí pro data ze sady *kaggle* segmentované metodou k-means (viz kap. 6.3.1).

Počet sousedů/počet intervalů	1	3
Euklidovská vzdálenost		
4,4,4	50%	46%
8,4,4	46%	46%
8,8,8	37%	38%
16,8,8	38%	36%
16,16,16	33%	36%
Manhattanská vzdálenost		
4,4,4	55%	50%
8,4,4	54%	53%
8,8,8	52%	56%
16,8,8	52%	54%
16,16,16	57%	57%
Kosinová vzdálenost		
4,4,4	50%	46%
8,4,4	46%	48%
8,8,8	36%	34%
16,8,8	34%	36%
16,16,16	32%	34%

Tabulka 23: Znázornění výsledků metody nejbližších sousedů (viz kap. 5.2) pro různý počet hledaných sousedů a různé vzdálenosti. Tyto výsledky platí pro nesegmentovaná data ze sady *traffic* (viz kap. 6.1).

Počet sousedů/počet intervalů	1	3
Euklidovská vzdálenost		
4,4,4	61%	63%
8,4,4	63%	64%
8,8,8	58%	59%
16,8,8	64%	61%
16,16,16	58%	59%
Manhattanská vzdálenost		
4,4,4	62%	63%
8,4,4	66%	67%
8,8,8	62%	63%
16,8,8	67%	69%
16,16,16	67%	67%
Kosinová vzdálenost		
4,4,4	62%	64%
8,4,4	65%	67%
8,8,8	57%	57%
16,8,8	64%	64%
16,16,16	62%	63%

Tabulka 24: Znázornění výsledků metody nejbližších sousedů (viz kap. 5.2) pro různý počet hledaných sousedů a různé vzdálenosti. Tyto výsledky platí pro data ze sady *traffic* segmentované Bayesovou metodou (viz kap. 6.3.3).

Počet sousedů/počet intervalů	1	3
Euklidovská vzdálenost		
4,4,4	56%	56%
8,4,4	57%	59%
8,8,8	60%	59%
16,8,8	58%	58%
16,16,16	55%	56%
Manhattanská vzdálenost		
4,4,4	60%	59%
8,4,4	60%	63%
8,8,8	61%	63%
16,8,8	61%	64%
16,16,16	62%	64%
Kosinová vzdálenost		
4,4,4	58%	59%
8,4,4	58%	60%
8,8,8	56%	57%
16,8,8	63%	61%
16,16,16	61%	60%

Tabulka 25: Znázornění výsledků metody nejbližších sousedů (viz kap. 5.2) pro různý počet hledaných sousedů a různé vzdálenosti. Tyto výsledky platí pro data ze sady *traffic* segmentované metodou detekce hran (viz kap. 6.3.2).

Počet sousedů/počet intervalů	1	3
Euklidovská vzdálenost		
4,4,4	50%	48%
8,4,4	51%	52%
8,8,8	51%	50%
16,8,8	48%	50%
16,16,16	50%	48%
Manhattanská vzdálenost		
4,4,4	50%	50%
8,4,4	53%	53%
8,8,8	56%	55%
16,8,8	56%	56%
16,16,16	57%	57%
Kosinová vzdálenost		
4,4,4	50%	51%
8,4,4	52%	53%
8,8,8	52%	51%
16,8,8	52%	53%
16,16,16	54%	55%

Tabulka 26: Znázornění výsledků metody nejbližších sousedů (viz kap. 5.2) pro různý počet hledaných sousedů a různé vzdálenosti. Tyto výsledky platí pro data ze sady *traffic* segmentované metodou k-means (viz kap. 6.3.1).

A.3 Výsledky experimentů při klasifikaci barev ve videu

Metoda segmentace/počet intervalů	bez segm	Bayesova	k-means	detekce hran
4,4,4	63%	56%	49%	52%
8,4,4	66%	63%	65%	61%
8,8,8	65%	58%	65%	57%
16,8,8	68%	60%	60%	55%
16,16,16	71%	65%	64%	60%

Tabulka 27: Znázornění přesnosti metody podpůrných vektorů (viz kap. 5.1) v závislosti na použité segmentační metodě (viz kap. 6.3). Tyto výsledky platí pro snímky získané z dopravního video záznamu. Trénovací sada pro tento případ obsahovala přibližně 100 snímků pro každou barvu a vektory příznaků byly vytvořené z 3D histogramů v barevném modelu HSV

Metoda segmentace/počet intervalů	bez segm	Bayesova	k-means	detekce hran
Metrika TOP2				
4,4,4	69%	64%	66%	69%
8,4,4	70%	73%	85%	71%
8,8,8	73%	72%	83%	71%
16,8,8	79%	72%	81%	73%
16,16,16	79%	79%	81%	75%
32,16,16	81%	-	-	-
Metrika TOP3				
4,4,4	81%	82%	80%	85%
8,4,4	81%	84%	92%	86%
8,8,8	84%	87%	90%	86%
16,8,8	87%	86%	90%	91%
16,16,16	89%	89%	89%	87%

Tabulka 28: Znázornění přesnosti metrik TOP 2 a TOP 3 (viz kap. 6.2) pro metody podpůrných vektorů (viz kap. 5.1) jak pro segmentovaná, tak pro nesegmentovaná data získaná z dopravního video záznamu. Tyto výsledky platí pro trénovací sadu obsahující přibližně 100 snímků pro každou barvu. Vektory příznaku byly vytvořené z 3D histogramů v barevném modelu HSV.

Počet sousedů/počet intervalů	1	3
Euklidovská vzdálenost		
4,4,4	74%	76%
8,4,4	77%	76%
8,8,8	77%	73%
16,8,8	77%	70%
16,16,16	71%	71%
Manhattanská vzdálenost		
4,4,4	74%	76%
8,4,4	76%	75%
8,8,8	77%	77%
16,8,8	76%	76%
16,16,16	76%	76%
Kosinová vzdálenost		
4,4,4	71%	76%
8,4,4	77%	75%
8,8,8	74%	72%
16,8,8	73%	68%
16,16,16	69%	68%

Tabulka 29: Znázornění přesnosti získané pomocí metody nejbližších sousedů (viz kap. 5.2) pro nesegmentovaná data z video záznamu. Výsledky jsou uvedeny pro různý počet hledaných sousedů a různé vzdálenosti.

Počet sousedů/počet intervalů	1	3
Euklidovská vzdálenost		
4,4,4	70%	70%
8,4,4	70%	70%
8,8,8	72%	71%
16,8,8	74%	71%
16,16,16	69%	70%
Manhattanská vzdálenost		
4,4,4	70%	71%
8,4,4	77%	77%
8,8,8	73%	77%
16,8,8	74%	76%
16,16,16	75%	76%
Kosinová vzdálenost		
4,4,4	67%	70%
8,4,4	70%	72%
8,8,8	70%	70%
16,8,8	74%	69%
16,16,16	70%	74%

Tabulka 30: Znázornění výsledků metody nejbližších sousedů (viz kap. 5.2) pro data extrahovaná z video záznamu a segmentovaná Bayesovou metodou (viz kap. 6.3.3). Výsledky jsou uvedeny pro různý počet hledaných sousedů a různé vzdálenosti.

Počet sousedů/počet intervalů	1	3
Euklidovská vzdálenost		
4,4,4	68%	69%
8,4,4	60%	63%
8,8,8	65%	59%
16,8,8	61%	60%
16,16,16	59%	59%
Manhattanská vzdálenost		
4,4,4	70%	67%
8,4,4	64%	65%
8,8,8	71%	67%
16,8,8	71%	67%
16,16,16	70%	67%
Kosinová vzdálenost		
4,4,4	68%	67%
8,4,4	64%	61%
8,8,8	64%	62%
16,8,8	63%	62%
16,16,16	65%	60%

Tabulka 31: Znázornění výsledků metody nejbližších sousedů (viz kap. 5.2) pro data extrahovaná z video záznamu a segmentovaná metodou detekce hran (viz kap. 6.3.2). Výsledky jsou uvedeny pro různý počet hledaných sousedů a různé vzdálenosti.

Počet sousedů/počet intervalů	1	3
Euklidovská vzdálenost		
4,4,4	52%	48%
8,4,4	48%	46%
8,8,8	48%	42%
16,8,8	46%	45%
16,16,16	45%	43%
Manhattanská vzdálenost		
4,4,4	60%	60%
8,4,4	63%	51%
8,8,8	58%	48%
16,8,8	56%	53%
16,16,16	59%	57%
Kosinová vzdálenost		
4,4,4	59%	55%
8,4,4	49%	52%
8,8,8	54%	45%
16,8,8	48%	50%
16,16,16	56%	50%

Tabulka 32: Znázornění výsledků metody nejbližších sousedů (viz kap. 5.2) pro data extrahovaná z video záznamu a segmentovaná metodou k-means (viz kap. 6.3.1) Výsledky jsou uvedeny pro různý počet hledaných sousedů a různé vzdálenosti.

B Seznam elektronických příloh

Adresářová struktura souboru `lysenvik_matlab.zip`:

- `/anotace/` - obsahuje anotační textové dokumenty
- `/segmentace/` - zdrojové kódy segmentačních metod - Bayesovy, detekce hran a k-means
- `/klasifikace/` - zdrojové kódy realizující trénování a testování klasifikačních metod
- `/klasifikace/barevne_definice/` - zdrojové kódy metody barevných definic
- `/klasifikace/metoda_podpurnych_vektoru/` - zdrojové kódy metody podpurných vektoru
- `/klasifikace/metoda_nejblizsich_sousedu/` - zdrojové kódy metody podpurných vektoru
- `/klasifikace/video/` - zdrojové kódy realizující klasifikaci barev ve video záznamu
- `/DEMO/` - obsahuje ukázkou klasifikace barev dopravních prostředků pomocí již natrénovaného modelu SVM na datech segmentovaných Bayesovou metodou. Součástí DEMO souboru je také ukázkou Bayesovy segmentační metody.

Kvůli velikosti natrénovaných modelů konvoluční neuronové sítě a datových souborů *kaggle* a *traffic*, nejsou součástí elektronické přílohy a jsou dostupné na:

https://drive.google.com/drive/folders/1PK_9itHL4uAsJ2Bdfmp6EUe2KC712T0E?usp=sharing [21.05.2020].

Použitá literatura

- [1] K. S. Aarathi a A. Abraham. “Vehicle color recognition using deep learning for hazy images”. In: *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*. 2017, s. 335–339.
- [2] *Analýza shluků CLU [online]*. Katedra analytické chemie, Univerzita Pardubice. Dostupné z: <https://meloun.upce.cz/docs/research/chemometrics/methodology/4gmetody.pdf>.
- [3] Johann Andersen a Steve Sutcliffe. “Intelligent Transport Systems (ITS)-An Overview”. In: *IFAC Proceedings Volumes* 33.18 (2000), s. 99–106.
- [4] António dos Anjos a Hamid Reza Shahbazkia. *BI-LEVEL IMAGE THRESHOLDING*.
- [5] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [6] Tuntai Boriboonthana. *Color-Classification-CNN*. <https://github.com/charlespwd/project-title>. 2017.
- [7] Yuri Y Boykov a M-P Jolly. “Interactive graph cuts for optimal boundary & region segmentation of objects in ND images”. In: *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*. Sv. 1. IEEE. 2001, s. 105–112.
- [8] *Carvana Image Masking Challenge*. <https://www.kaggle.com/c/carvana-image-masking-challenge>. Zář. 2017.
- [9] Corinna Cortes a Vladimir Vapnik. “Support-vector networks”. In: *Machine learning* 20.3 (1995), s. 273–297.
- [10] Thomas Cover a Peter Hart. “Nearest neighbor pattern classification”. In: *IEEE transactions on information theory* 13.1 (1967), s. 21–27.
- [11] Susan Crow. “Manhattan Distance”. In: *Encyclopedia of Machine Learning*. Ed. Claude Sammut a Geoffrey I. Webb. Boston, MA: Springer US, 2010, s. 639–639. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_506. URL: https://doi.org/10.1007/978-0-387-30164-8_506.
- [12] K. Deb a K. Jo. *HSI color based vehicle license plate detection*. 2008.
- [13] Adit Deshpande. *A Beginner’s Guide To Understanding Convolutional Neural Networks*. <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner’s-Guide-To-Understanding-Convolutional-Neural-Networks/>.
- [14] Y. Dong, M. Pei a X. Qin. “Vehicle Color Recognition Based on License Plate Color”. In: (lis. 2014), s. 264–267. ISSN: null. DOI: 10.1109/CIS.2014.63.
- [15] Kaibo Duan et al. “Multi-category classification by soft-max combination of binary classifiers”. In: *International Workshop on Multiple Classifier Systems*. Springer. 2003, s. 125–134.
- [16] Erida Dule, Muhittin Gökmen a Muhammet S. Beratoglu. “A convenient feature vector construction for vehicle color recognition”. In: 2010.
- [17] P. Sudharshan Duth a M. Deepa. “Color detection in RGB-modeled images using MAT LAB”. In: *International journal of engineering and technology* 7 (2018), s. 29.
- [18] Alexandros Eleftheriadis et al. *Methods and apparatus for performing digital image and video segmentation and compression using 3-D depth information*. US Patent 6,055,330. Dub. 2000.
- [19] Pedro F Felzenszwalb a Daniel P Huttenlocher. “Distance transforms of sampled functions”. In: *Theory of computing* 8.1 (2012), s. 415–428.
- [20] E. Fida et al. *Unsupervised image segmentation using lab color space*. 2017.

- [21] Yoav Freund a Robert E Schapire. “Large margin classification using the perceptron algorithm”. In: *Machine learning* 37.3 (1999), s. 277–296.
- [22] Martin Gardner. “Taxicab Geometry”. In: *The Last Recreations: Hydras, Eggs, and Other Mathematical Mystifications*. New York, NY: Springer New York, 1997, s. 159–175. ISBN: 978-0-387-30389-5. DOI: 10.1007/978-0-387-30389-5_10. URL: https://doi.org/10.1007/978-0-387-30389-5_10.
- [23] J-L Gauvain a Chin-Hui Lee. “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains”. In: *IEEE transactions on speech and audio processing* 2.2 (1994), s. 291–298.
- [24] Richard E Gonzalez Rafael C. - Woods. “Digital image processing [2nd ed.]. 2nd ed.” In: (2002).
- [25] Antonio Guillén et al. “Color Measurement and Analysis of Fruit with a Battery-Less NFC Sensor”. In: *Sensors* (dub. 2019), s. 1741. DOI: 10.3390/s19071741.
- [26] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, s. 770–778.
- [27] Kaiming He et al. “Mask R-CNN”. In: *CoRR* abs/1703.06870 (2017). arXiv: 1703.06870. URL: <http://arxiv.org/abs/1703.06870>.
- [28] Sepp Hochreiter a Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), s. 1735–1780.
- [29] Berthold KP Horn a Brian G Schunck. “Determining optical flow”. In: *Techniques and Applications of Image Understanding*. Sv. 281. International Society for Optics a Photonics. 1981, s. 319–331.
- [30] J. Hsieh et al. “Vehicle Color Classification Under Different Lighting Conditions Through Color Correction”. In: *IEEE Sensors Journal* 15.2 (ún. 2015), s. 971–983. ISSN: 2379-9153. DOI: 10.1109/JSEN.2014.2358079.
- [31] Chuanping Hu et al. “Vehicle color recognition with spatial pyramid deep learning”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.5 (2015), s. 2925–2934.
- [32] François Chollet et al. *Keras*. <https://keras.io>. 2015.
- [33] Noor Ibraheem et al. “Understanding Color Models: A Review”. In: *ARPN Journal of Science and Technology* (led. 2012).
- [34] Ing. Vítězslav Beran Ing. Michal Španěl. *Obrazové segmentační techniky [online]*. Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav počítačové grafiky a multimédií. Dostupné z: <http://www.fit.vutbr.cz/~spanel/segmentace/>.
- [35] Anil K Jain a Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- [36] Dannhoferová Jana. *Velká kniha barev: kompletní průvodce pro grafiky, fotografie a designéry*. Brno: Computer Press, ISBN 978-80-251-3785-7, 2012.
- [37] Ilona Janáková. *Přednáška: Detekce geometrických primitiv [online]*. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Dostupné z: <https://docplayer.cz/107403306-Detekce-geometrickych-primitiv.html>.
- [38] Angur Mahmud Jarman. *Hierarchical Cluster Analysis: Comparison of Single linkage, Complete linkage, Average linkage and Centroid Linkage Method*.
- [39] Y. Jeong, K. H. Park a D. Park. “Homogeneity patch search method for efficient vehicle color classification using front-of-vehicle image”. In: (říj. 2017), s. 1–5. ISSN: null. DOI: 10.1109/IST.2017.8261544.
- [40] Singaraju Jyothi a K.Bhargavi. “A Survey on Threshold Based Segmentation Technique in Image Processing”. In: *26. K. Bhargavi, S. Jyothi* 3 (lis. 2014).

- [41] Tejas K et al. *Efficient Licence Plate Detection By Unique Edge Detection Algorithm and Smarter Interpretation Through IoT*. 2017. arXiv: 1710.10418 [cs.NE].
- [42] Leslie Pack Kaelbling, Michael L Littman a Andrew W Moore. “Reinforcement learning: A survey”. In: *Journal of artificial intelligence research* 4 (1996), s. 237–285.
- [43] Annika Kangas, Sanna Laukkanen a Jyrki Kangas. “Social choice theory and its applications in sustainable forest management—a review”. In: *Forest Policy and economics* 9.1 (2006), s. 77–92.
- [44] Nick Kanopoulos, Nagesh Vasanthavada a Robert L Baker. “Design of an image edge detection filter using the Sobel operator”. In: *IEEE Journal of solid-state circuits* 23.2 (1988), s. 358–367.
- [45] K. Kim, S. Park a Y. Choi. “Deciding the Number of Color Histogram Bins for Vehicle Color Recognition”. In: (pros. 2008), s. 134–138. ISSN: null. DOI: 10.1109/APSCC.2008.207.
- [46] Kyungnam Kim et al. “Real-time foreground–background segmentation using codebook model”. In: *Real-time imaging* 11.3 (2005), s. 172–185.
- [47] Tae-Hyun Kim et al. *Multi-class Classifier-Based Adaboost Algorithm*. Říj. 2011. DOI: 10.1007/978-3-642-31919-8_16.
- [48] Yoon Kim et al. “Character-aware neural language models”. In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.
- [49] Alex Krizhevsky, Ilya Sutskever a Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, s. 1097–1105.
- [50] Sang Uk Lee, Seok Yoon Chung a Rae Hong Park. “A comparative performance study of several global thresholding techniques for segmentation”. In: *Computer Vision, Graphics, and Image Processing* 52.2 (1990), s. 171–190.
- [51] Christina Leslie, Eleazar Eskin a William Stafford Noble. “The spectrum kernel: A string kernel for SVM protein classification”. In: *Biocomputing 2002*. World Scientific, 2001, s. 564–575.
- [52] X. Li et al. *Vehicle Color Recognition Using Vector Matching of Template*. 2010.
- [53] Konstantinos G Liakos et al. “Machine learning in agriculture: A review”. In: *Sensors* 18.8 (2018), s. 2674.
- [54] Tony Lindeberg. “Scale-space for discrete signals”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 12 (dub. 1990), s. 234–254. DOI: 10.1109/34.49051.
- [55] D. Liu a J. Yu. “Otsu Method and K-means”. In: *2009 Ninth International Conference on Hybrid Intelligent Systems*. Sv. 1. 2009, s. 344–349.
- [56] Wei Liu et al. “Ssd: Single shot multibox detector”. In: *European conference on computer vision*. Springer. 2016, s. 21–37.
- [57] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Sv. 1. 14. Oakland, CA, USA. 1967, s. 281–297.
- [58] Raman Maini a Himanshu Aggarwal. “Study and comparison of various image edge detection techniques”. In: *International journal of image processing (IJIP)* 3.1 (2009), s. 1–11.
- [59] Christopher D Manning, Prabhakar Raghavan a Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.
- [60] Martin Engelin and Felix Machado de Silva. “Troll detection : A comparative study in detecting troll farms on Twitter using cluster analysis”. In: (2016).

- [61] Jiri Matas a Karel Zimmermann. “Unconstrained licence plate and text localization and recognition”. In: *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005*. IEEE. 2005, s. 225–230.
- [62] Sachin Mehta et al. “Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation”. In: *Proceedings of the european conference on computer vision (ECCV)*. 2018, s. 552–568.
- [63] *Metoda nejbližšího souseda*. <https://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--vicerozmerne-metody-pro-analyzu-dat--klasifikace--klasifikace-podle-minimalni-vzdalenosti--princip-klasifikace-podle-minimalni-vzdalenosti--metoda-nejblizsio-souseda>.
- [64] Milan Šonka a Václav Hlaváč. *Počítačové vidění*. Praha: Grada, 1992. 252 s. ISBN 80-85424-67-3.
- [65] Fausto Milletari, Nassir Navab a Seyed-Ahmad Ahmadi. “V-net: Fully convolutional neural networks for volumetric medical image segmentation”. In: *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE. 2016, s. 565–571.
- [66] Shervin Minaee et al. *Image Segmentation Using Deep Learning: A Survey*. 2020. arXiv: 2001.05566 [cs.CV].
- [67] Rodrigo Moraes, João Francisco Valiati a Wilson P Gavião Neto. “Document-level sentiment classification: An empirical comparison between SVM and ANN”. In: *Expert Systems with Applications* 40.2 (2013), s. 621–633.
- [68] Fionn Murtagh a Pedro Contreras. *Methods of Hierarchical Clustering*. 2011. arXiv: 1105.0121 [cs.IR].
- [69] Shmuel Nitzan a Ariel Rubinstein. “A further characterization of Borda ranking method”. In: *Public choice* (1981), s. 153–158.
- [70] Dan W Patterson. *Introduction to artificial intelligence and expert systems*. Prentice-hall of India, 1990.
- [71] Massimo Piccardi. “Background subtraction techniques: a review”. In: *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*. Sv. 4. IEEE. 2004, s. 3099–3104.
- [72] Rudra P. K. Poudel, Stephan Liwicki a Roberto Cipolla. “Fast-SCNN: Fast Semantic Segmentation Network”. In: *CoRR* abs/1902.04502 (2019). arXiv: 1902.04502. URL: <http://arxiv.org/abs/1902.04502>.
- [73] Reza Fuad Rachmadi a Purnama. “Vehicle color recognition using convolutional neural network”. In: *arXiv preprint arXiv:1510.07391* (2018).
- [74] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, s. 779–788.
- [75] E. Reinhard et al. “Color transfer between images”. In: *IEEE Computer Graphics and Applications* 21.5 (čvc 2001), s. 34–41. ISSN: 1558-1756. DOI: 10.1109/38.946629.
- [76] Ph.D. RNDr. PaedDr. Eva Volná. “Studijní materiály pro distanční kurz: Neuronové sítě 1”. In: (2008). https://www1.osu.cz/~volna/Neuronove_site_skripta.pdf.
- [77] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).
- [78] Sumit Saha. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [79] Ravi Srisha. “Operators Used In Edge Detection Computation: A Case Study”. In: 7 (led. 2012).

- [80] *Studijní materiály: (Algoritmy podpůrných vektorů [online]*. Masarykova univerzita v Plzni, Fakulta informatiky, Strojové učení. Dostupné z: https://is.muni.cz/el/fi/podzim2005/PA034/09_SVM.pdf.
- [81] Richard S Sutton, Andrew G Barto et al. *Introduction to reinforcement learning*. Sv. 135. MIT press Cambridge, 1998.
- [82] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, s. 1–9.
- [83] D. S. B. Tilakaratna et al. “Image analysis algorithms for vehicle color recognition”. In: (břez. 2017), s. 1–4. ISSN: null. DOI: 10.1109/IEECON.2017.8075881.
- [84] Santosh Kumar Uppada. “Centroid Based Clustering Algorithms-A Clarion Study”. In: 2014.
- [85] FAN Wei. “Color Image Segmentation Algorithm Based on Region Growth [J]”. In: *Computer Engineering* 13 (2010).
- [86] Bing Xu et al. “Empirical Evaluation of Rectified Activations in Convolutional Network”. In: *CoRR* abs/1505.00853 (2015). arXiv: 1505.00853. URL: <http://arxiv.org/abs/1505.00853>.
- [87] Maoke Yang et al. “Denseaspp for semantic segmentation in street scenes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, s. 3684–3692.
- [88] Chen Zhang et al. “Color Image Segmentation in RGB Color Space Based on Color Saliency”. In: led. 2014.
- [89] Liping Zheng, Guangyao Li a Yun Bao. “Improvement of grayscale image 2D maximum entropy threshold segmentation method”. In: *2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM)*. Sv. 1. IEEE. 2010, s. 324–328.