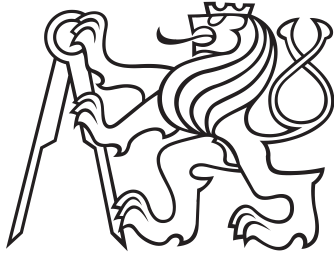


Diplomová práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra radioelektroniky

## Inteligentní kamera pro kontrolu parkoviště s časovým omezením

**David Roubal**

Vedoucí: doc. Ing. Stanislav Vítek, Ph.D.

Studijní program: Elektronika a komunikace

Květen 2023



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Roubal** Jméno: **David** Osobní číslo: **474249**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra radioelektroniky**  
Studijní program: **Elektronika a komunikace**  
Specializace: **Technologie internetu věcí**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Inteligentní kamera pro kontrolu parkoviště s časovým omezením**

Název diplomové práce anglicky:

**Smart Camera to Control Parking Slot with Time Limit**

Pokyny pro vypracování:

Navrhněte a implementujte inteligentní bezpečnostní kameru pro správu parkoviště s časovým omezením (parkoviště typu K+R). Při vypracování se řiďte následujícími pokyny:

1. Kamera disponuje uživatelským rozhraním pro konfiguraci parkovacího místa a času, po který může jedno vozidlo na parkovišti stát. Pro přístup k rozhraní kamera disponuje rozhraním Ethernet.
2. Veškeré zpracování obrazu probíhá přímo v kameře, výstupem z kamery jsou pouze telemetrická data.
3. Komunikace s centrálním serverem probíhá pomocí LPWAN sítě (např. LoRa).
4. Činnost kamery proveďte provozem v reálných podmínkách, diskutujte vliv světelných podmínek na přesnost detekce vozidla.

Seznam doporučené literatury:

- [1] VÍTEK, Stanislav; MELNIČUK, Petr. A distributed wireless camera system for the management of parking spaces. *Sensors*, 2018, 18.1: 69.  
[2] CHEN, Lun-Chi, et al. Video-based parking occupancy detection for smart control system. *Applied Sciences*, 2020, 10.3: 1079.

Jméno a pracoviště vedoucí(ho) diplomové práce:

**doc. Ing. Stanislav Vítek, Ph.D. katedra radioelektroniky FEL**

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

Datum zadání diplomové práce: **02.02.2022**

Termín odevzdání diplomové práce: \_\_\_\_\_

Platnost zadání diplomové práce: **30.09.2023**

\_\_\_\_\_  
doc. Ing. Stanislav Vítek, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
doc. Ing. Stanislav Vítek, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

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

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen 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 studenta



## Poděkování

Rád bych poděkoval doc. Ing. Stanislavu Vítkovi, Ph.D., vedoucímu mé diplomové práce, za odborné vedení, trpělivost, připomínky a čas, který mi v průběhu zpracování diplomové práce věnoval.

## Prohlášení

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

V Praze, 26. května 2023

## Abstrakt

Cílem diplomové práce je vyvinout prototyp kamerového modulu pro monitorování obsazenosti parkovišť v reálném čase. Kamerový modul zpracovává data z kamery pomocí mikropočítače Raspberry Pi Zero a odesílá je do webového rozhraní přes síť SigFox. Webové rozhraní umožňuje zobrazit a analyzovat shromážděná data o parkovacích místech. Pro klasifikaci parkovacích míst je použita metoda podpůrných vektorů (SVM) s příznaky založenými na histogramu orientovaných gradientů (HOG) a barevném histogramu. Prototyp dosahuje přesnosti okolo 80 % v závislosti na počasí a světelných podmínkách. Součástí práce je rozsáhlá datová sada, která byla vytvořena pomocí programu Godot pro simulaci různých podmínek a speciálních případů. Na konec je v práci poukázáno na možnosti využití pokročilého hlubokého učení v této oblasti, jak pro klasifikaci parkovacích míst, tak pro detekci poznávacích značek vozidel v obrázcích.

**Klíčová slova:** strojové učení, chytré parkování, analýza vlivu počasí, Raspberry Pi, Edge Computing

**Vedoucí:** doc. Ing. Stanislav Vítek, Ph.D.

## Abstract

The objective of this thesis is to construct a prototype camera module capable of real-time monitoring of parking occupancy. The camera module processes data captured by the camera using a Raspberry Pi Zero microcomputer and transmits it to a web interface via the SigFox network. The web interface enables the visualization and analysis of collected data on parking spaces. A support vector machine (SVM) with features derived from the histogram of oriented gradients (HOG) and color histogram is employed for parking space classification. The prototype achieves an accuracy of around 80% depending on weather and lighting conditions. The thesis includes an extensive dataset that was created using the Godot program to simulate various conditions and special cases. In the end, the work points out the possibilities of using advanced deep learning in this area, both for the classification of parking spaces and for the detection of vehicle license plates in images.

**Keywords:** machine learning, smart parking, weather impact analysis, Raspberry Pi, Edge Computing

# Obsah

|  |           |  |
|--|-----------|--|
| <b>Seznam zkratk</b>   | <b>1</b>  |  |
| <b>Úvod</b>  | <b>3</b>  |  |
| <b>1 Teoretická část</b>   | <b>5</b>  |  |
| 1.1 Využití kamerových systémů pro sledování volné kapacity parkovišť .      | 5         |  |
| 1.2 Zpracování obrazu pro sledování volné kapacity parkovišť . . . . .       | 7         |  |
| 1.2.1 Klasifikace, detekce a segmentace . . . . .                            | 7         |  |
| 1.2.2 Postupy založené na klasifikaci  | 10        |  |
| 1.2.3 Postupy založené na detekci .  | 12        |  |
| 1.2.4 Deskriptor HOG . . . . .   | 13        |  |
| 1.2.5 Kombinace HOG a HSV příznaků pro extrakci informací z obrazu . . . . . | 15        |  |
| 1.2.6 Vybraná metoda klasifikace .   | 15        |  |
| 1.2.7 Detekce, klasifikace a segmentace: YOLOv8 . . . . .                    | 16        |  |
| 1.3 Datové sady . . . . .  | 17        |  |
| 1.3.1 Úprava datové sady . . . . .   | 17        |  |
| 1.3.2 Veřejně dostupné snímky parkovišť . . . . .                            | 18        |  |
| 1.4 Komunikační síť pro přenos dat z kamery . . . . .                        | 19        |  |
| 1.5 Výpočetní zařízení pro zpracování obrazu . . . . .                       | 20        |  |
| 1.6 Zobrazení dat z parkovišť . . . . .                                      | 21        |  |
| 1.6.1 Existující řešení . . . . .  | 21        |  |
| <b>2 Praktická část</b>  | <b>23</b> |  |
| 2.1 Implementace kamery pro snímání parkoviště . . . . .                     | 23        |  |
| 2.2 Webové rozhraní pro zobrazení dat z parkovišť . . . . .                  | 24        |  |
| 2.2.1 Funkce, testování a výsledky z reálného provozu . . . . .              | 24        |  |
| 2.2.2 REST API pro komunikaci se serverem . . . . .                          | 26        |  |
| 2.3 Testování a srovnání metod klasifikace . . . . .                         | 27        |  |
| 2.3.1 Použití sloučených příznaků HOG a HSV histogramu . . . . .             | 27        |  |
| 2.3.2 Testování vlivu počasí a světelných podmínek CNRPark+EXT . . . . .     | 28        |  |
| 2.3.3 Testovací sada PKLot . . . . .   | 33        |  |
| 2.3.4 Použití transformace při předzpracování snímku . . . . .               | 35        |  |
| 2.3.5 Vliv velikosti datové sady na parametry modelu . . . . .               | 35        |  |
| 2.4 Kontrola délky parkování vozidel   | 35        |  |
| <b>Závěr</b>   | <b>39</b> |  |
| <b>Literatura</b>  | <b>43</b> |  |
| <b>A Výsledný testovaný kamerový modul</b>                                   | <b>47</b> |  |

## Obrázky

|  |    |   |    |
|--|----|---|----|
| 1 Definování úloh kamerového systému a datového úložiště . . . . .   | 4  | 2.5 Ukázka snímku z datové sady CNRPark+EXT . . . . .                                   | 35 |
| 1.1 Nevhodné umístění kamery pro sledování parkovišť . . . . .   | 7  | 2.6 Detekce SPZ pomocí obrysů . . . . .   | 36 |
| 1.2 Vlevo výsledek detekce, vpravo výsledek segmentace [1] . . . . .   | 8  | 2.7 Detekce SPZ pomocí HAAR příznaků (rohy) a kaskádového klasifikátoru . . . . .       | 37 |
| 1.3 Blokové schéma procesu trénování modelu pomocí příznaků . . . . .  | 9  | 2.8 Detekce SPZ pomocí algoritmu YOLOv8 naučeného na vlastních datech . . . . .         | 37 |
| 1.4 (a) Otočení parkovacího místa o specifický úhel, (b) změna perspektivy obrazu pomocí vhodné transformace . . . . . | 11 | A.1 Zapojení LoRa modulu (Ra-02) k Raspberry Pi Zero 2 (vysílač) . . . . .              | 47 |
| 1.5 Aplikace Sobelova filtru (b) a Cannyho hranového detektoru (c) na Originální obrázek (a) . . . . .                 | 14 | A.2 Propojení LoRa modulu (Ra-02) a ESP32 (přijímač a brána do sítě Internet) . . . . . | 47 |
| 1.6 Postup při vytváření HOG deskriptoru . . . . .   | 14 | A.3 Testovaný kamerový modul využívající Raspberry Pi Zero 2 a SigFox . . . . .         | 48 |
| 1.7 (a) Originální obrázek, (b) ekvalizovaný histogram (vylepšený kontrast). . . . .                                   | 18 |   |    |
| 2.1 Náhled statického obrázku parkoviště s vyznačenými parkovacími místy . . . . .                                     | 26 |   |    |
| 2.2 Ukázka pozitivních snímků z CNRPark+EXT (deštivo/18. hodina) . . . . .   | 31 |   |    |
| 2.3 Vylepšení kontrastu snímku pomocí ekvalizace histogramu (vlevo – originální obrázek, vpravo – upravený) . . . . .  | 31 |   |    |
| 2.4 Ukázka snímku z datové sady PKLot . . . . .  | 34 |   |    |



## Tabulky


|  |    |   |    |
|--|----|---|----|
| 1.1 Používané vlastnosti pro popis snímku parkovacího místa . . . . .  | 9  | 2.8 Výsledky klasifikace HOG – srovnání průměrných výsledků jednotlivých klasifikátorů (trénovací data – FELslot) . . . . . | 33 |
| 1.2 Počet anotovaných oblastí v datové sadě PKLot . . . . .  | 19 | 2.9 Výsledky klasifikace na testovacích datech PKLot . . . . .  | 34 |
| 1.3 Počet snímků datové sady CNRPark+EXT podle počasí a denní doby . . . . .   | 19 |   |    |
| 1.4 Přehled snímačů v modulech pro rozhraní CSI-2 . . . . .  | 22 |   |    |
| 2.1 Srovnání rychlosti knihovny OpenCV pro Python a C++ . . . . .  | 24 |   |    |
| 2.2 Výsledky klasifikace – porovnání HOG a HOG+HSV . . . . .   | 27 |   |    |
| 2.3 Výsledky klasifikace RF s HOG+HSV – různé světelné podmínky (trénovací data – CTUPARK3Db) . . . . .                            | 29 |   |    |
| 2.4 Výsledky klasifikace RF s HOG – různé světelné podmínky (trénovací data – CTUPARK3Db) . . . . .                                | 30 |   |    |
| 2.5 Výsledky klasifikace HOG+HSV – srovnání průměrných výsledků jednotlivých klasifikátorů (trénovací data – CTUPARK3Db) . . . . . | 32 |   |    |
| 2.6 Výsledky klasifikace HOG – srovnání průměrných výsledků jednotlivých klasifikátorů (trénovací data – CTUPARK3Db) . . . . .     | 32 |   |    |
| 2.7 Výsledky klasifikace HOG+HSV – srovnání průměrných výsledků jednotlivých klasifikátorů (trénovací data – FELslot) . . . . .    | 33 |   |    |





## Seznam zkratek

|        |  |
|--------|--|
| ADASYN | Adaptive Synthetic Sampling – adaptivní syntetické vzorkování                                |
| API    | Application Programming Interface – rozhraní pro programování aplikací                       |
| CNN    | Convolutional Neural Network – konvoluční neuronová síť                                      |
| CS-LBP | Center-Symmetric Local Binary Pattern – středově symetrický lokální binární vzor             |
| CSI    | Camera Serial Interface  |
| CSV    | Comma-separated values – hodnoty oddělené čárkami  |
| FF     | Fast Forest – rychlý les   |
| FLOPS  | Floating Point Operations Per Second – počet operací s plovoucí desetinnou čárkou za sekundu |
| FPS    | Frames Per Second – snímky za sekundu  |
| GAN    | Generative Adversarial Network – generativní soupeřící síť                                   |
| GLCM   | Gray Level Co-occurrence Matrix – matice společného výskytu                                  |
| HOG    | Histogram of Oriented Gradients – histogram orientovaných gradientů                          |
| LBP    | Local Binary Pattern – lokální binární vzor  |
| LPQ    | Local Phase Quantization   |
| LPWAN  | Low-Power Wide-Area Network – nízkoenergetická rozlehlá síť                                  |
| LR     | Logistic Regression – logistická regrese   |



|         |   |
|---------|---|
| LUV     | barevný prostor, který souvisí s barevným prostorem CIE XYZ   |
| MSE     | Mean Squared Error – střední kvadratická chyba  |
| OCR     | Optical Character Recognition – optické rozpoznávání znaků  |
| P+R     | Park and Ride – zaparkuj a jeď  |
| REST    | Representational State Transfer – reprezentativní přenos stavu                                      |
| RF      | Random Forest – náhodný les   |
| ROC     | Receiver Operating Characteristic   |
| SIFT    | Scale Invariant Feature Transform – transformace invariantní vůči změně měřítka                     |
| SMOTE   | Synthetic Minority Over-sampling Technique – syntetická technika převzorkování menšiny              |
| SURF    | Speeded Up Robust Features  |
| SVM     | Support Vector Machine – metoda podpůrných vektorů  |
| XCS-LBP | Extended Center-Symmetric Local Binary Pattern – rozšířený středově symetrický lokální binární vzor |
| YOLO    | You Only Look Once – pouze jedno zpracování   |



## Úvod

Každoročně se problematika parkování ve městských aglomeracích stává komplikovanější, protože narůstá počet motorových vozidel. V metropoli Praze se počet vozidel zvyšuje každým rokem o desetitisíce. Přetížení parkovacích kapacit má za následek dopravní komplikace, jako je nelegální parkování na nevyhrazených místech, blokování vozovky nebo dlouhé hledání volného parkovacího místa. Doba potřebná k nalezení parkovacího místa může přesáhnout 10 minut.

Existuje mnoho metod pro řešení problémů s parkováním. U velkých měst se staví nová parkoviště P+R, která jsou napojena na veřejnou dopravu. Zvyšuje se také počet parkovacích míst ve městech, ale v některých oblastech již není možné přidávat nová místa. Parkoviště v centrech měst mohou být rušivá.

Jednou z možností je regulace obsazenosti parkovišť úpravou cen za parkování, což by mohlo přesměrovat řidiče na méně vytížená parkoviště nebo do parkovacích domů. Cena za parkování se může dynamicky měnit podle ročního období, dne v týdnu a denní doby. Pro takovou regulaci je nutné analyzovat obsazenost parkovišť.

Sledování obsazenosti umožňuje sdělovat informace řidičům prostřednictvím informačních tabulí nebo aplikace. Při současném rozšíření autonomních vozidel jsou dopravní informace užitečné a pomáhají plně využít jejich funkce.

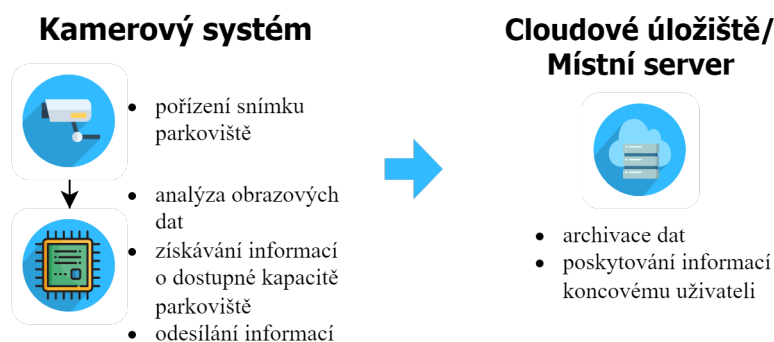
Tato práce se zabývá akvizicí a analýzou obrazových dat z parkovacích ploch. Získané informace o obsazenosti jsou přenášeny do uživatelského rozhraní prostřednictvím sítě LPWAN. Využití kamery jako senzoru pro detekci obsazenosti parkovacích míst přináší řadu výhod, jako je pokrytí velké plochy

a schopnost rozlišovat různé objekty. Je také možné ověřit oprávnění k parkování. Hlavním omezením je spolehlivost výsledku, která může být ovlivněna vnějšími faktory, jako je špatné počasí nebo nedostatek světla.

Klasifikace parkovacích míst a odesílání pouze nezbytných telemetrických dat prostřednictvím technologie LPWAN minimalizuje riziko přetížení sítě přenosem obrazových dat v reálném čase. Zjednodušený proces je znázorněn na obr. 1. Pro přenos komprimovaného obrazu z kamery lze použít připojení Ethernet nebo Wi-Fi.

Cílem této práce je zpracovávání a analýza obrazových dat. Získané informace jsou prezentovány uživateli prostřednictvím webového rozhraní. Rozhraní poskytuje základní informace o parkovišti. Tyto informace zahrnují geografickou polohu, provozní hodiny a pravidla parkování.

Inteligentní kamerový systém poskytuje další údaje. Tyto údaje zahrnují počet obsazených parkovacích míst, překročení doby parkování a kontrolu oprávnění k parkování na vyhrazených místech. V případě krádeže nebo jiné nezákonné činnosti je vytvářen záznam.



**Obrázek 1:** Definování úloh kamerového systému a datového úložiště

# Kapitola 1

## Teoretická část

Teoretická část práce se zabývá analýzou a výběrem optimálního řešení pro monitorování volné kapacity parkoviště s využitím metod strojového i hlubokého učení. V rámci analýzy jsou porovnány charakteristiky jednotlivých klasifikačních metod běžně používaných pro tento typ úloh. U strojového i hlubokého učení je popsán konkrétní algoritmus. Jsou definovány klíčové parametry datové sady a možnosti využití hotových datových sad. Jsou stanoveny požadavky vyplývající pro LPWAN, technologii pro bezdrátový přenos s nízkou spotřebou energie. Dále jsou zkoumány požadavky na výpočetní zařízení pro provádění klasifikace vozidel na parkovišti. Různé typy zařízení jsou porovnány z hlediska výkonu, spotřeby energie a kompatibility s metodami strojového i hlubokého učení.

### 1.1 Využití kamerových systémů pro sledování volné kapacity parkovišť

Příkladem používaných technologií pro monitorování volné kapacity parkovišť jsou následující:

- Řešení s jedním senzorem pro každé parkovací místo, jako jsou indukční smyčky, ultrazvukové senzory nebo optické závory.
- Automatické elektrické závory umístěné na vjezdu a výjezdu z parkoviště.
- Kamerové systémy, které umožňují pokrytí větších oblastí jedním senzorem.

Každé z řešení pro monitorování obsazenosti parkovacích míst má své výhody i nevýhody. Tato práce se zabývá použitím kamerového systému. Klíčovými parametry kamerového systému jsou přesnost správného určení obsazeného místa parkoviště a rychlost zjištění změny stavu obsazenosti parkovacího místa. Důležitá je také robustnost v různých podmínkách, kterými jsou například znečištění, překážky ve výhledu, zhoršené světelné podmínky nebo povětrnostní vlivy. Dalšími parametry jsou škálovatelnost, tedy jak dobře se dané řešení přizpůsobuje velikosti a složitosti parkoviště, a náklady na senzor, údržbu a další potřebnou infrastrukturu.

V poslední době došlo ke značnému zlepšení přesnosti a rychlosti kamerových systémů díky pokroku v oblasti hlubokého učení. Tyto systémy jsou nyní schopny dosáhnout téměř dokonalých výsledků. Nicméně, ohledně robustnosti kamerových systémů existují určité nevýhody. Tyto systémy jsou závislé na dostatečném osvětlení prostředí, což může být problém v noci nebo za špatného počasí. Dalším faktorem ovlivňujícím výsledek je přítomnost překážek ve výhledu kamery, jako jsou například rozrostlé stromy (viz obr. 1.1). Výhodou těchto systémů je jejich schopnost rozpoznat libovolné objekty, které brání v zaparkování.

Škálovatelnost kamerových systémů spočívá v optimalizaci počtu kamer pro pokrytí maximálního počtu parkovacích míst. Hlavním úkolem je vybrat umístění kamery tak, aby nedocházelo k příliš velkému zkreslení obrazu a případně se pokusit zkreslení eliminovat. Ideálně by měla být kamera umístěna co nejvýše nad parkovištěm, aby mohla zachytit největší plochu. Úhel mezi zenitem a bodem zájmu by měl být minimální, aby nedocházelo k překryvu vozidel a zakrytí parkovacích míst za nimi (viz obr. 1.1a). V opačném případě je nutné použít více kamer, což zvyšuje nároky na infrastrukturu a sběr dat. Jedna kamera s vhodným umístěním je schopna vyhodnocovat i stovky parkovacích míst.





**Obrázek 1.1:** Nevhodné umístění kamery pro sledování parkovišť

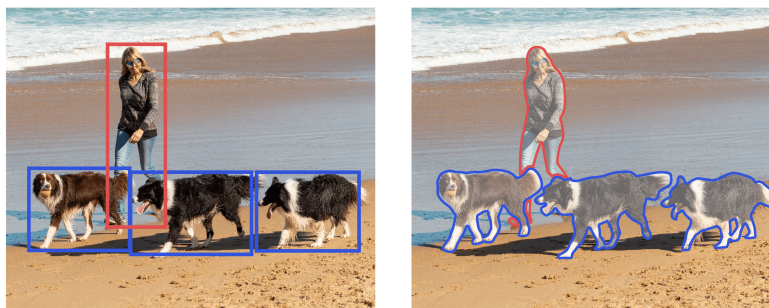
Náklady na kamerové systémy se skládají z nákladů na pořízení kamery, její instalaci a údržbu, a také z nákladů na přenos a uložení dat. Náklady na samotnou kameru se velmi liší podle požadovaných vlastností. Mezi důležité parametry se řadí ohnisková vzdálenost, formát snímače a s nimi související zorný úhel a zkreslení objektivu. Kamera může mít i další užitečné vlastnosti, jako je měření expozice, infračervené noční vidění nebo odolnost proti povětrnostním vlivům. Tyto vlastnosti mohou přispět ke zlepšení kvality obrazu a zvýšení spolehlivosti kamerového systému.

## 1.2 Zpracování obrazu pro sledování volné kapacity parkovišť

### 1.2.1 Klasifikace, detekce a segmentace

V oblasti zpracování obrazu se rozlišují tři hlavní úlohy: klasifikace, detekce a segmentace.

- **Klasifikace:** Spočívá v přiřazení obrazu do předem definovaných tříd na základě jeho vlastností, jako jsou barva, textura a tvar.
- **Detekce:** Zabývá se identifikací a lokalizací specifických objektů nebo vzorů v obraze a určuje jejich přítomnost a polohu pomocí ohraničujících rámečků.
- **Segmentace:** Slouží k rozdělení obrazu na jednotlivé regiony nebo pixely na základě barvy, textury a jasů a jejich sloučení do specifických kategorií.



**Obrázek 1.2:** Vlevo výsledek detekce, vpravo výsledek segmentace [1]

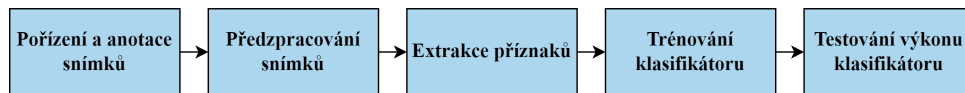
U klasifikace, například při zjišťování volné kapacity parkoviště, je potřeba vyznačit každé parkovací místo, které má být klasifikováno. Pro účel klasifikace se pak vyřízne ze snímku. Výhodou klasifikace může být nižší výpočetní náročnost, která se odvíjí i od typu trénovaného modelu. Pro trénování klasifikátoru se používají celé obrázky, zatímco u detekce nebo segmentace jsou potřeba anotovaná data pro jednotlivé segmenty v kontextu celého obrazu.

Na rozdíl od klasifikace má detekce výhodu v tom, že není nutné anotovat jednotlivá parkovací místa. Detekce také umožňuje lépe vyhodnotit situaci, pokud automobil není správně zaparkován mezi vymezenými čarami. Všechny automobily v obraze jsou detekovány, spočítány a porovnány s maximální kapacitou parkoviště. Detekce vyhodnocuje i vozidla v pohybu mimo parkovací místa. Pokud je součástí snímku jiná komunikace, je třeba tato data ze záběru odstranit. Výsledky zahrnují oblast, ve které se dané vozidlo nachází. Tuto oblast definují čtyři čísla a lze je odeslat pro určení přesného umístění obsazených parkovacích míst. Pokud jsou data anotována, lze tuto informaci vyhodnotit přímo v kamerovém systému a odeslat pouze binární číslo pro každé jedno parkovací místo.

Segmentace je podobná detekci, ale má navíc výhodu v tom, že dokáže určit i velikost oblasti parkovacích míst pro přesnější vyhodnocení obsazenosti. Umožňuje také sledovat jednotlivá vozidla a identifikovat je, určit zda odjíždí a zda jej na parkovišti nahradilo jiné vozidlo.

Prvním typem metod strojového učení pro klasifikaci jsou metody využívající extrahované vektory příznaků, známé jako deskriptory. Tyto deskriptory

obsahují informace o barvách, hranách a rozích v datově úspornější podobě, například v histogramu. Srovnání některých relevantních vlastností s příklady deskriptorů pro typ úlohy s zjišťování volné kapacity parkoviště je v tab. 1.1. Příznakový vektor poté slouží k trénování modelu. Zjednodušené schéma celého cyklu je na obr. 1.3. Pořízení a anotace dat, stejně jako předzpracování snímků jsou podrobněji popsány v kapitole 1.3.



**Obrázek 1.3:** Blokové schéma procesu trénování modelu pomocí příznaků

**Tabulka 1.1:** Používané vlastnosti pro popis snímku parkovacího místa

| Vlastnost | Výhody                          | Nevýhody                 | Příklady deskriptorů               |
|-----------|---------------------------------|--------------------------|------------------------------------|
| Barva     | snadné měření                   | vliv světelných podmínek | Barevný histogram, barevné momenty |
| Tvar      | snadná detekce                  | vliv perspektivy kamery  | HOG, SIFT                          |
| Textura   | lze rozlišit různé typy vozidel | obtížnější detekce       | LBP, LPQ, GLCM                     |

Klasifikátory, které využívají extrahované deskriptory, zahrnují metodu podpurných vektorů (SVM), náhodný les (RF) a SGD. SVM hledá nadrovinu pro optimální rozdělení tříd a dokáže klasifikovat velký počet tříd. RF vytváří rozhodovací stromy a kombinuje jejich výsledky. Jeho výhodou je snadná paralelizace, ale může být pomalejší než ostatní metody. SGD iterativně upravuje váhy modelu pomocí gradientu chybové funkce a umožňuje sekvenční učení po menších částech.

Další možností klasifikace je hluboké učení. Algoritmy hlubokého učení integrují procesy předzpracování obrazu, extrakce příznaků a trénování do jednoho bloku. Tento blok je tvořen umělou neuronovou sítí, která se skládá z jednotlivých vrstev, každá s vlastním účelem. Pro zpracování obrazu se

v hlubokém učení často používají konvoluční neuronové sítě (CNN). Tyto sítě se skládají z několika druhů vrstev. Vstupní vrstva přijímá obrázky a předává je dalším vrstvám. Konvoluční vrstva aplikuje filtr pro vytvoření map příznaků. Pooling vrstva redukuje rozměry map příznaků, snižuje počet parametrů a zvyšuje robustnost. Plně propojená vrstva spojuje mapy příznaků do jednoho vektoru a provádí klasifikaci. Výstupní vrstva vrací výsledky klasifikace na základě plně propojené vrstvy (třídy nebo skóre). Existuje mnoho modelů CNN používaných pro klasifikaci a detekci objektů v obrazu. Příklady modelů s nízkou výpočetní náročností, které se hodí pro zpracování na okraji sítě jsou AlexNet, GoogLeNet, MobileNet a EfficientNet.

### 1.2.2 Postupy založené na klasifikaci

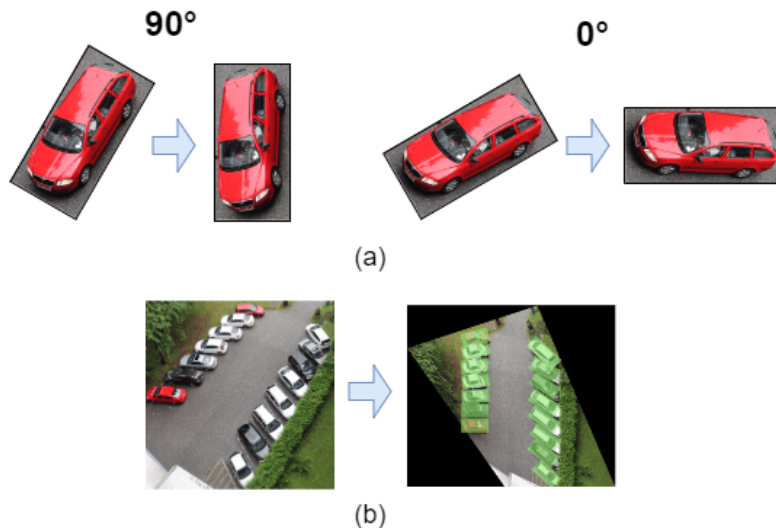
V přehledovém článku [2] jsou srovnány metody klasifikace parkovacích míst pomocí postupu s extrahovanými příznaky a hlubokého učení. Průzkum využívá veřejně dostupné datové sady a rozlišuje metody pro klasifikaci jednotlivých parkovacích míst, automatickou detekci parkovacích míst a detekci a počítání automobilů. Některé výsledky jsou vyhodnoceny jako nereprodukovatelné nebo obsahují postupy testování, které mohou vést ke zkresleným výsledkům.

Článek [3] popisuje reprodukovatelný postup klasifikace parkovacích míst. Postup kombinuje příznaky z barevného prostoru LUV, velikosti gradientu a kvantizovaných gradientů. Klasifikace se provádí metodami podpurných vektorů (SVM) a lineární regrese (LR), přičemž LR dosahuje lepších výsledků. Algoritmus funguje dobře i na datových sadách, kde není natrénován. Autor navrhuje tuto metodu pro situace, kdy nelze klasifikátor natrénovat pro každou kameru zvlášť.

V dalším článku s reprodukovatelným výsledkem se autoři zabývají detekováním automobilů pomocí metody odstraňování pozadí. V článku je použit algoritmus SURF natrénován pomocí SVM. Výhodou přístupu je, že není potřeba anotovat obrázky, ale parkovací místa jsou nalezena automaticky. Dále metoda dosahuje poměrně vysoké přesnosti na datovém setu, na kterém je i trénována. Nicméně na jiné datové sadě jsou výsledky metody horší. [4]

Některé postupy při předzpracování obrazu snižují vliv úhlu a polohy

kamery na přesnost klasifikace parkovacích míst. Například otočením obdélníkového výřezu parkovacího místa o určitý úhel [5, 6] nebo rozdělením do skupin podle úhlu natočení a klasifikace každé skupiny zvlášť [7]. Některé algoritmy pro extrakci příznaků jsou rotačně invariantní, jako LBP nebo SIFT. Jiné metody transformují perspektivu obrazu, aby se parkovací místa parkovací místa více přibližovala tvaru obdélníku [8].



**Obrázek 1.4:** (a) Otočení parkovacího místa o specifický úhel, (b) změna perspektivy obrazu pomocí vhodné transformace

Studie [9] ukazuje, že zprůměrování výsledků ze dvou kamer s překrývajícími se zornými poli vede k přibližně o 5 % vyšší přesnosti. Dosažená přesnost je 99,0 % při použití kombinace barevného rozlišení vozidla od asfaltového pozadí a Cannyho hranového detektoru. Tato metoda využívá pouze techniky zpracování obrazu a je proto výpočetně nenáročná pro analýzu snímků v reálném čase.

Článek [10] zkoumá vliv počasí na přesnost modelu se SIFT a SVM. Při testování na datové sadě CNRPark+EXT dosahuje model nejvyšší přesnosti okolo 90 % s minimálním rozdílem pro různé druhy počasí. Model je natrénovaný pomocí slunečných snímků. Zařazení snímků se zamračeným nebo deštivým počasím do trénovací sady přesnost modelu nezlepšuje. To ukazuje, že pro SIFT není informace o změně světla tak důležitá jako tvar, hrany nebo výrazné body.

Studie [11] porovnává příznaky z 400 obrázků s chodci ve stupních šedi. Kombinuje techniky strojového učení SVM s lineárním jádrem nebo polynomálním jádrem a Random Forest (RF) s různými algoritmy pro extrakci příznaků, HOG, SURF, CS-LBP a XCS-LBP. Deskriptory XCS-LBP a CS-LBP jsou kratší a vylepšené varianty LBP. Velmi dobrých výsledků dosahuje kombinace algoritmu HOG s XCS-LBP a SURF-128 s XCS-LBP. Obě metody dosahují přesnosti 99,2 %. Nejlepší je kombinace CS-LBP a XCS-LBP trénovaná algoritmem RF, která na testovací datové sadě dosahuje 100% přesnosti. Testy jsou provedeny na 30 % datové sady a 70 % je určeno pro trénování.

Dalším vylepšením zmíněného algoritmu založeného na LBP příznacích může být například algoritmus Fast Forest (FF). Dosahuje vyšší rychlosti než původní algoritmus RF při zachování stejné přesnosti. [12]

V přehledovém článku [2] jsou nejpoužívanějšími metodami pro klasifikaci parkovacích míst pomocí hlubokého učení ty, které využívají model AlexNet nebo jeho modifikace. AlexNet je nenáročná konvoluční neuronová síť (CNN) s 8 vrstvami. Většina metod založených na CNN dosahuje vysoké přesnosti i při použití dat z jiného parkoviště a některé architektury dosahují téměř 100% hranice přesnosti. Metody s příznakovými vektory vykazují nižší přesnost, ale není dostatek relevantních dat pro kvantitativní porovnání.

### 1.2.3 Postupy založené na detekci

Jednou z metod pro zjištění obsazenosti parkoviště je detekce a počítání automobilů pomocí algoritmu. Často se používají konvoluční neuronové sítě. Pro vyhodnocení metod se používá metrika střední absolutní chyby (MAE).

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (1.1)$$

kde  $n$  je počet pozorování,  $y_i$  je skutečný počet detekovaných objektů (např. automobilů na parkovišti) a  $\hat{y}_i$  je skutečný počet objektů (např. skutečný počet automobilů na parkovišti). Nižší hodnota MAE znamená lepší přesnost detekce.

Nejlepší sítí pro detekci obsazených míst podle srovnání v článku [2] je LC-FCN8. Autoři architektury v [13] používají novou ztrátovou funkci, která umožňuje trénování na bodových anotacích. Ty jsou snazší k vytváření než

plné anotace. LC-FCN8 je testována na různých datových sadách, včetně PKLot. Na této sadě dosahuje střední absolutní chyby  $MAE = 0,2$ . Druhý nejlepší výsledek, s hodnotou  $MAE$  rovnou  $1,0$ , je dosažen pomocí sítě Mask R-CNN na datové sadě CNRPark+EXT [2].

#### 1.2.4 Deskriptor HOG

HOG je algoritmus pro extrakci příznaků z obrazu, kterým jsou získány informace o hranách a jejich orientaci. Hrana je definována jako místo v obraze, kde dochází k výrazné změně intenzity pixelů. Tuto změnu intenzity lze detekovat pomocí derivace, ale pro zjednodušení výpočtu se často používá konvoluce s vhodným jádrem.

Při extrakci HOG deskriptoru se nejdříve provádí filtrace obrazu pomocí Sobelova operátoru. Pro každý obrazový prvek je vypočtena konvoluce se dvěma různými jádry. První jádro  $\mathbf{G}_x$  aproximuje gradient ve vodorovném směru a druhé jádro  $\mathbf{G}_y$  aproximuje gradient ve svislém směru.

$$\mathbf{G}_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad \mathbf{G}_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Velikost gradientu v každém bodě obrazu je vypočtena jako euklidovská vzdálenost z obou složek gradientu:

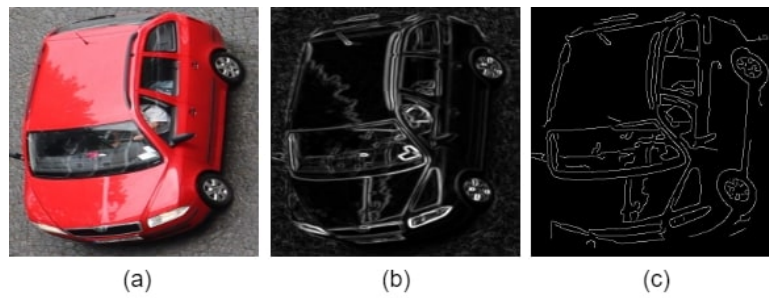
$$b = \sqrt{b_x^2 + b_y^2}. \quad (1.2)$$

Jednou z výhod použití Sobelova operátoru je možnost získání informací o směru gradientu. Směr lze vypočítat z výsledku konvoluce:

$$b = \arctan \frac{b_y}{b_x}. \quad (1.3)$$

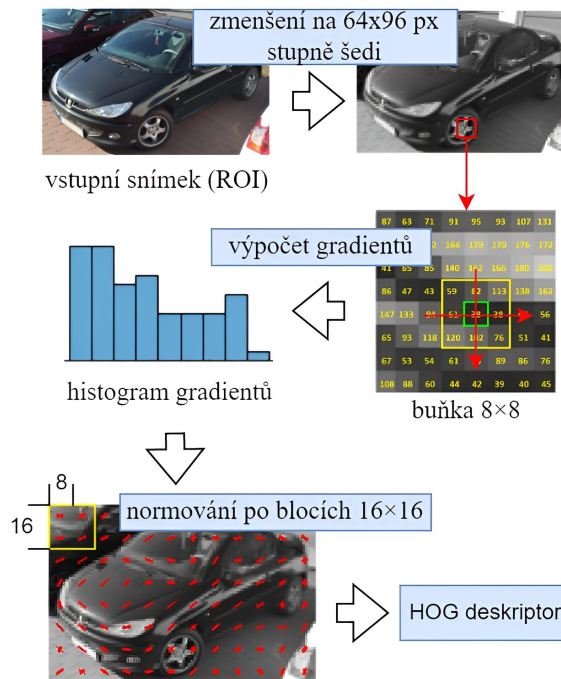
Hlavní nevýhodou je vyšší počet detekovaných falešných hran a citlivost na šum. Použitím vhodného filtru lze snížit počet hran v obraze a zachovat pouze ty nejvýraznější. Na obr. 1.5 je zobrazen příklad použití Sobelova filtru pro detekci hran a Cannyho hranového detektoru, který navíc využívá nastavování prahu pro zvýraznění hran. [14]

Po výpočtu velikosti a směru gradientu pomocí Sobelova operátoru se obraz rozdělí na malé oblasti (buňky). Pro každou buňku se vytvoří histogram



**Obrázek 1.5:** Aplikace Sobelova filtru (b) a Cannyho hranového detektoru (c) na Originální obrázek (a)

orientací gradientů, kde každý obrazový prvek přispívá do příslušného sloupce podle svého směru gradientu a váhy odpovídající jeho velikosti gradientu. Histogramy jednotlivých buněk se poté normalizují a spojí do jednoho vektoru, který tvoří HOG deskriptor. Příklad postupu vytváření HOG deskriptoru je znázorněn na obr. 1.6.



**Obrázek 1.6:** Postup při vytváření HOG deskriptoru



### 1.2.5 Kombinace HOG a HSV příznaků pro extrakci informací z obrazu

Pro popis barev obrázku lze použít barevný model HSV, který se skládá ze tří složek: odstín (Hue), sytost (Saturation) a jas (Value). Tento model umožňuje oddělit složku jasu od složek odstínu a sytosti. Jas může být ovlivněn faktory jako je osvětlení scény nebo expozice snímku, zatímco odstín a sytost poskytují lepší popis samotné barvy objektů na obrázku [15]. Pro oddělení jasové složky lze použít jen histogram odstínu a sytosti. Kombinací deskriptorů HSV a HOG je možné získat výsledný deskriptor obsahující navíc informaci o barvě.

Když spojujeme deskriptory, je důležité zajistit, aby oba vektory byly normalizované a měly stejnou váhu. Pro dosažení optimálního poměru lze váhu příznaků upravit vynásobením vektoru konstantou. Možnost, jak normalizovat vektory příznaků  $\mathbf{x}$  před spojením je vydělit každý prvek jeho L2-normou. L2-norma vektoru je definována jako:

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}, \quad (1.4)$$

kde  $n$  je počet prvků vektoru.

Normalizovaný vektor příznaků  $\mathbf{x}'$  je získán dělením každého prvku  $x$  L2-normou:

$$x'_i = \frac{x_i}{\|\mathbf{x}\|_2}. \quad (1.5)$$

### 1.2.6 Vybraná metoda klasifikace

Výběr klasifikátoru vyžaduje experimentování. Závisí na vstupních datech a požadavcích, jako jsou přesnost, rychlost nebo adaptabilita na nová data.

SVM klasifikátor se často používá pro binární klasifikaci. Jeho úkolem je najít nadrovinu, která nejlépe odděluje data do různých tříd. Volba jádra ovlivňuje tvar nadroviny. Jádro může být nelineární, což je výhodné u datových sad, které nejsou lineárně separabilní.

Přesnost SVM klasifikátoru závisí i na předzpracování vstupních dat. Vstupem do SVM klasifikátoru musí být vždy stejně dlouhé vektory příznaků. Výstupem klasifikátoru je predikce třídy pro každý vstupní vzorek. SVM

určuje vzdálenost vzorku od rozhodovací hranice. Pokud je vzdálenost kladná, je vzorek klasifikován do jedné třídy. Pokud je záporná, je klasifikován do druhé třídy.

Výsledná přesnost klasifikátoru se ověřuje na datové sadě pro validaci. Vhodná datová sada by měla být reprezentativní pro data, na kterých bude klasifikátor používán v praxi. Měla by být dostatečně velká, aby umožnila spolehlivé odhadnutí přesnosti klasifikátoru. A měla by být nezávislá na trénovacích datech, aby nedocházelo ke zkreslení výsledků.

Pro určení kvality natrénovaného modelu se v této práci používají metriky ROC křivka a přesnost. ROC křivka je grafické znázornění závislosti mezi počtem pravdivě určených pozitivních snímků a počtem falešně určených pozitivních snímků. Přesnost klasifikátoru se vypočítá podle vztahu:

$$\text{Přesnost} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1.6)$$

kde  $TP$  jsou pravdivá pozitiva,  $FP$  jsou falešná pozitiva,  $TN$  jsou pravdivá negativa a  $FN$  jsou falešná negativa.

### 1.2.7 Detekce, klasifikace a segmentace: YOLOv8

Algoritmus YOLO (You Only Look Once) používá jednu konvoluční neuronovou síť. Při detekci zpracuje obrázek a rozdělí ho na podoblasti. Pro každou podoblast poté předpoví ohraničující boxy a pravděpodobnosti pro jednotlivé třídy. Při segmentaci navíc vymezení objekt na úrovni obrazových prvků a vrátí sadu masek pro nalezené objekty.

YOLO je vhodný pro zpracování obrazu v reálném čase. Obsahuje i malé předem natrénované modely, které jsou vhodné pro použití na mikropočítače typu Raspberry Pi. Většina verzí používá jako aplikační rámec Python knihovnu PyTorch.

První verze YOLO byla vydána v roce 2015. Nejnovější verze YOLOv8 byla spuštěna na začátku roku 2023 společností Ultralytics. Nová verze zahrnuje klasifikaci, segmentaci a detekci, ale také modely pro sledování objektů a rozlišení lidského postoje. Vývoj se zaměřuje na neustálé posouvání rychlosti i přesnosti detekce. [16]

## 1.3 Datové sady

### 1.3.1 Úprava datové sady

Vhodnou selekcí trénovacích dat zvyšujeme přesnost modelu. Nedostatečně rozmanitá data mohou vést k přetrénování modelu, který pak špatně funguje na nových datech. Pro zlepšení vlastností datové sady odstraňujeme nadbytečná data. Snížení počtu vzorků může výrazně ovlivnit dobu trénování. Například u metody SVM, která má časovou náročnost  $O(n^2)$ , může být vysoký počet vzorků problematický [17].

Chybějící snímky do datové sady je možné získat augmentací, tedy vytvářením různých variant z dostupných dat pomocí transformací jako rotace, změna velikosti, oříznutí, přidání šumu atd. Tím zlepšujeme zobecňování klasifikátoru a zvyšujeme šanci, že bude fungovat na nových datech. Pro přidání dat do datových sad existují metody jako SMOTE (syntetická menšinová převzorkovací technika), ADASYN (adaptivní syntetické vzorkování) nebo GAN (generativní soupeřící sítě). Tyto metody se hodí pro nevyvážené datové sady.

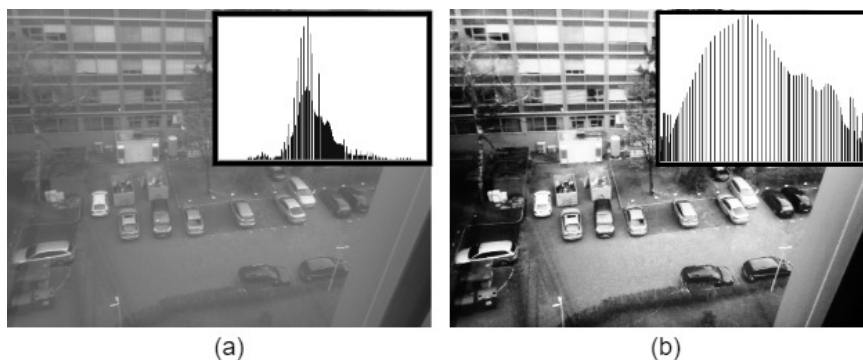
SMOTE je metoda pro vytváření umělých vzorků. Tato metoda může vyvážit velikost tříd, tím že vytváří syntetické vzorky pro třídu s menším zastoupením [18]. ADASYN je rozšíření SMOTE, které generuje množství umělých vzorků pro méně zastoupenou třídu podle obtížnosti klasifikace vzorků [19].

GAN používá dvě neuronové sítě (generátor a diskriminátor) k vytváření realistických vzorků. Generátor se snaží oklamat diskriminátor tím, že vytváří vzorky podobné skutečným datům, zatímco diskriminátor se snaží rozlišit skutečné a falešné vzorky. Obě sítě soutěží, dokud nedosáhnou rovnováhy, kdy generátor produkuje vysoce kvalitní vzorky. [20]

Jednou z možností vylepšení trénovacích dat je normalizace. Například standardizace pomocí z-skóre nebo směrodatné odchylky. Tyto metody mohou zvýšit stabilitu klasifikátoru. Normalizace se aplikuje jak na trénovací sadu, tak na testované objekty.

Přesnost klasifikace může být zlepšena pomocí jednoduchých technik předzpra-

cování obrazu. Například použitím vhodného filtru pro odstranění šumu. Ekvalizace histogramu může vylepšit kontrast a eliminovat proměnlivost v osvětlení (viz obr. 1.7). Další možností je úprava geometrie, například rotace nebo korekce zkreslení perspektivy [7].



**Obrázek 1.7:** (a) Originální obrázek, (b) ekvalizovaný histogram (vylepšený kontrast).

### 1.3.2 Veřejně dostupné snímky parkovišť

V této práci jsou použity dvě veřejné datové sady. První je PKLot se snímky dvou parkovišť v Brazílii [21]. Druhá je CNRPark-EXT se snímky jednotlivých parkovacích míst z Itálie [22]. CNRPark-EXT je použita k ověření algoritmů při změně počasí a různých světelných podmínkách. Tab. 1.3 ukazuje počet parkovacích míst rozdělených podle počasí a času během dne. Data z PKLot slouží v této práci ke srovnání dosažené přesnosti a obsahují záběry za různého počasí ze dvou různých parkovišť.

Datová sada CNRPark obsahuje obrázky parkovacích míst, ale nezahrnuje širší záběr parkoviště. Proto není vhodná pro trénování modelu detekce, který vyžaduje anotovaná data z obrázku s dalšími objekty. Datová sada PKLot je vhodnější volbou, protože obsahuje záběry celého parkoviště.

**Tabulka 1.2:** Počet anotovaných oblastí v datové sadě PKLot

|                | volné  | obsazené |
|----------------|--------|----------|
| testovací data | 36584  | 34100    |
| trénovací data | 265908 | 231948   |
| validační data | 73629  | 69687    |
| součet         | 376121 | 335735   |

**Tabulka 1.3:** Počet snímků datové sady CNRPark+EXT podle počasí a denní doby

| denní doba [h] | slunečno |       | deštivo  |       | zataženo |       |
|----------------|----------|-------|----------|-------|----------|-------|
|                | obsazené | volné | obsazené | volné | obsazené | volné |
| 7              | 144      | 2157  | 133      | 1859  | 265      | 3687  |
| 8              | 311      | 2190  | 454      | 3386  | 568      | 3912  |
| 9              | 1096     | 1354  | 1353     | 2480  | 1783     | 2692  |
| 10             | 1792     | 707   | 2336     | 1439  | 3034     | 1422  |
| 11             | 1832     | 650   | 2506     | 1293  | 3179     | 1255  |
| 12             | 1792     | 693   | 2453     | 1366  | 3074     | 1326  |
| 13             | 1559     | 885   | 2127     | 1697  | 2807     | 1658  |
| 14             | 1554     | 855   | 2134     | 1697  | 2863     | 1601  |
| 15             | 1602     | 787   | 2293     | 1535  | 2899     | 1565  |
| 16             | 1410     | 978   | 2007     | 1814  | 2389     | 1793  |
| 17             | 160      | 179   | 646      | 371   | 143      | 171   |
| 18             | —        | —     | 15       | 13    | —        | —     |

## 1.4 Komunikační síť pro přenos dat z kamery

V České republice jsou dostupné tři významné druhy LPWAN (nízkoenergetických rozlehlých sítí) s dostatečným pokrytím: LoraWAN, SigFox a NB-IoT (Narrowband IoT). Existuje také možnost vytvořit si vlastní síť



Nejdostupnější vývojové desky, jako jsou Arduino nebo ESP32-CAM, nejsou vhodné pro kamerový systém, protože postrádají knihovny pro strojové učení a mohou data z kamery pouze odesílat na server.

Vývojová deska STM32 Nucleo typu STM32F7XX umožňuje použití knihovny OpenCV a připojení kamery. Tato verze má ARM Cortex-M7 jádro s rychlostí až 216 MHz a podporuje operace s plovoucí desetinou čárkou (FPU). Je možné zprovoznit základní počítačové vidění, například SVM klasifikaci nebo HAAR detekci s nízkým výkonem. [23]

Dostatečný výkon s velkou uživatelskou podporou mají mikropočítače Raspberry Pi a Nvidia Jetson. Obě mají vlastní distribuci operačního systému Linux a jsou dostupné oblíbené knihovny pro počítačové vidění, jako jsou TensorFlow, PyTorch nebo Keras. Desky mají dostatečné množství paměti a výkonu a disponují Ethernet rozhraním a vstupně-výstupními piny pro komunikaci s dalšími moduly. Mají také CSI-2 konektor pro připojení kamery. Nevýhodou Nvidia Jetson oproti Raspberry Pi je vyšší cena, ale díky grafickému procesoru dosahuje v úlohách strojového učení velmi vysokého výkonu. Například Jetson TX1 má výkon až 1 TFLOPS [24], zatímco Raspberry Pi 4 dosahuje teoreticky okolo 13,5–32 GFLOPS [25].

V tab. 1.4 je seznam některých snímačů, které používají moduly s rozhraním CSI-2 kompatibilním s Raspberry Pi a Nvidia Jetson. Zorný úhel kamery se liší podle použité čočky, kterou lze změnit připevněním objektivu. Moduly podporují čočky s montáží CS nebo M12. Horizontální zorné pole kamery je obvykle okolo 40–60 °, ale s vhodnou čočkou lze zorné pole zvýšit například na 170 °. To však vede ke vzniku vyššího zkreslení a efektu tzv. „rybího oka“.

## 1.6 Zobrazení dat z parkovišť

### 1.6.1 Existující řešení

Již existuje mnoho systémů, které sledují obsazenost parkovišť a informují o ní prostřednictvím aplikace. Mezi globální řešení patří například aplikace ParkMe<sup>1</sup>, která umožňuje rezervovat parkovací místa a sledovat obsazenost.

<sup>1</sup>Dostupné z: <https://www.parkme.com/>.

**Tabulka 1.4:** Přehled snímačů v modulech pro rozhraní CSI-2

| Snímač         | Odhadovaná<br>cena [Kč] | Rozlišení<br>videa [px]<br>(FPS)  | Formát<br>snímače<br>(palce) |
|----------------|-------------------------|-----------------------------------|------------------------------|
| OV5640, OV5642 | 700                     | 1920x1080 (30)                    | 1/4                          |
| IMX477         | 2000                    | 1920x1080 (30),<br>1280x720 (60)  | 1/2,3                        |
| IMX219         | 700                     | 1920x1080 (60),<br>1280x720 (180) | 1/4                          |
| OV5647         | 300                     | 1920x1080 (30),<br>1280x720 (60)  | 1/4                          |

Dalším příkladem je Parkopedia<sup>2</sup>. Ta obsahuje databázi s více než 70 miliony parkovacích míst ve více než 15 000 městech po celém světě a nabízí možnost pokročilého vyhledávání.

V České republice existuje několik realizovaných řešení pro sledování obsazenosti parkovišť a rezervaci parkovacích míst. Mezi ně patří například Parkovanivbrne<sup>3</sup> a Zaparkuju<sup>4</sup>. Tyto aplikace umožňují rezervovat místo k parkování, platit parkovací poplatek přes webové rozhraní a poskytují spoustu informací k parkování ve městě, jako jsou informace o zónách, cenách, časových změnách a obsazenosti parkovišť.

Informace o parkování jsou dostupné z veřejně dostupného REST API na portálu Golemio nebo MojePrahaAPI. API pro parkování poskytuje informace o poloze, názvu, adrese a dostupnosti parkovacích míst. Detailní API poskytují další informace, jako jsou možnosti platby, celkový počet míst, informace o tarifech, typ parkování a typ zóny. [26, 27]

Evropská unie využívá standard DATEX II pro výměnu dopravních dat. Tento standard obsahuje modely pro odesílání dat z parkovišť a slouží pro dopravní podniky a mediální partnery.

<sup>2</sup>Dostupné z: <https://www.parkopedia.com/>.

<sup>3</sup>Dostupné z: <https://www.parkovanivbrne.cz/>.

<sup>4</sup>Dostupné z: <https://zaparkuju.cz/>.



## Kapitola 2

### Praktická část

Praktická část práce se zabývala realizací kamerového modulu komunikujícího přes LPWAN (SigFox a LoraWAN). Bylo provedeno srovnání rychlostí zpracování obrazu na Raspberry Pi verze 3, 4 a Zero 2,. Následně bylo vyvíjeno webové rozhraní pro zobrazení dat z parkoviště a webové rozhraní pro zpracování obrazu. Dále byly shrnuty možnosti komunikace kamerového modulu se serverem. Poté byly provedeny testy klasifikátorů a příznakových vektorů zkombinovaných z HOG a HSV histogramu. Pro testování byla využita vlastní modelovaná datová sada v programu pro tvorbu 3D grafiky. V poslední části byly diskutovány možnosti vyhodnocení délky stání na parkovišti.

### 2.1 Implementace kamery pro snímání parkoviště

Pro vytvoření prototypu (viz obr. v příloze A.3) byla zvolena vývojová deska Raspberry Pi. Zařízení bylo implementováno s Raspberry Pi Zero 2 a testování algoritmů proběhlo i na Raspberry Pi verze 3 a 4. Byl vybrán kamerový modul „Raspberry Pi kamera 5MP v1.3“, který má 5 MPx snímač OmniVision OV5647 (viz kap. 1.5) s úhlem záběru 65 °. Modul nemá IR filtr, takže není vhodný pro noční vidění s IR osvětlením.

Byly testovány dvě technologie pro odesílání dat. Dvoubodová komunikace byla dosažena pomocí modulů využívajících technologii LoRa. Data byla přijímána na vývojové desce ESP32 a zobrazena v konzoli na osobním počítači. Obr. v příloze (A.1, A.2) ukázaly zapojení LoRa modulů pro dosažení základní komunikace. Pro komunikaci pomocí sítě LoRa lze použít různé knihovny, například pro rodinu Raspberry je k dispozici knihovna PyLoRa, pro ESP32 je

k dispozici LoRa.h v aplikaci Arduino IDE. Nakonec byla úspěšně otestována Komunikace na vzdálenost 100 metrů v zastavěné oblasti.

Vybraná technologie pro odesílání dat kamerového modulu byla SigFox. Nabídla roční licenci na používání sítě a měla dostatečné pokrytí pro testování na parkovišti v Praze. Zařízení bylo umístěno do vodotěsného průhledného pouzdra s USB konektorem pro napájení z externí baterie nebo adaptérem. S externí baterií o kapacitě 10 000 mA vydržela kamera v provozu okolo 53 hodin při spuštěné Wi-Fi.

Kamera byla umístěna do vyšších pater budovy FEL ČVUT s výhledem na parkoviště a připojena k místní Wi-Fi. V intervalu 5 minut odesílala telemetrická data přes SigFox a zmenšený obrázek po internetu. Kamerový modul byl v provozu několik týdnů. Klasifikace probíhala pomocí kombinace HOG a SVM s modelem trénovaným na vlastní datové sadě. Klasifikátor v denních hodinách dosahoval přesnosti 70-80 %, ale za šera klesaly výsledky na 60-70 % a přibýval šum.

Tab. 2.1 ukázala výsledky měření rychlosti algoritmu HOG s délkou 1440. Klasifikace pomocí modelu SVM trvala desítky mikrosekund a byla zanedbatelná. Výsledky ukázaly, že algoritmus HOG v knihovně OpenCV není dostatečně optimalizovaný pro Python. Pro rychlou klasifikaci na zařízení s menším výpočetním výkonem mohlo být výhodnější použít jazyk C++.

**Tabulka 2.1:** Srovnání rychlosti knihovny OpenCV pro Python a C++

| Zařízení     | načtení obrázku [ms] |        | HOG extrakce [ms] |        |
|--------------|----------------------|--------|-------------------|--------|
|              | C++                  | Python | C++               | Python |
| RPi Zero 2 W | —                    | 10,45  | —                 | 186,72 |
| RPi 4B 8GB   | 12,60                | 9,28   | 34,50             | 146,51 |
| RPi 3 1GB    | 5,13                 | 5,14   | 48,88             | 152,60 |

## 2.2 Webové rozhraní pro zobrazení dat z parkovišť

### 2.2.1 Funkce, testování a výsledky z reálného provozu

Bylo vytvořeno webové rozhraní, které nabízelo tyto funkce:

- Náhled parkovišť s podrobnými informacemi a možností kategorizace do skupin.
- Vytváření soukromých skupin s přidělováním přístupu.
- Dávkové zobrazení nejčastěji otevíraných parkovišť na hlavní stránce s informacemi o obsazenosti.
- Vyhledávání parkovišť na mapě pomocí API Mapy.cz.
- Vytváření konfiguračních souborů pro kameru přímo v správě parkoviště.
- Nastavení přístupových klíčů a identifikátorů pro akvizici dat z kamery.
- Formuláře pro obnovu hesla, registraci a přihlášení s ověřením e-mailové adresy.

Rozhraní bylo vytvořeno v programovacím jazyce Node.js s použitím aplikačního rámce Express pro práci s REST API. Databáze MongoDB byla použita pro ukládání uživatelských dat i dat z kamery. Data z kamery se automaticky odstraňovala po 3 měsících kvůli úspoře zdrojů, ale bylo možné je stáhnout ve správě parkovišť.

Byl spuštěn omezovač počtu odeslaných požadavků v přihlašovacích a registračních formulářích. Při opakovaném zadání hesla se postupně oddalovala možnost opakovat zadání. Při odesílání přístupových formulářů z jedné IP adresy se po několika pokusech zablokoval přístup na 1 den.

Byl implementován WebSocket pomocí knihovny socket.io, který průběžně aktualizoval informace na stránkách uživatele bez dotazování na server. Díky tomu se vyznačená parkovací místa na obr. 2.1 měnila v reálném čase.

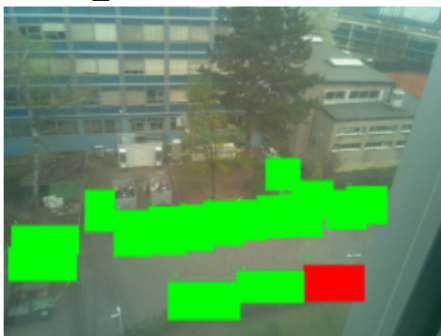
Rozhraní umožňovalo dva způsoby zobrazení parkoviště. Buď ukazovalo statický obrázek parkoviště a zvýrazňovalo jednotlivá parkovací místa podle příchozího hexadecimálního řetězce (červeně – obsazená, zeleně – volná) v případě odeslání dat přes LPWAN. Nebo rozhraní zobrazovalo přímo snímky z parkoviště.

Pro oddělení dokumentace a nástrojů od rozhraní byly vytvořeny druhé webové stránky v programovacím jazyce Python s použitím aplikačního rámce Django. Obsahovaly návody na instalaci serverů, nastavení odesílání dat a práci s rozhraním. Byly zde vytvořeny nástroje pro anotaci datových sad s různými typy výběrů a automaticky dopočítávaným úhlem natočení

parkovacího místa. Python byl vybrán kvůli podpoře moderních knihoven pro hluboké učení a možnosti vytvořit API pro testování klasifikačních metod a nahrávání vlastních modelů.

← Zpět

## CVUT\_1



Adresa: **Praha 6 - Dejvická**  
 Popis: Kamera na dvoře ČVUT FEL.  
 Souřadnice: 50.1021945N, 14.3926576E  
 Viditelnost: Veřejné  
 Obsazené místo: 1/17

**Obrázek 2.1:** Náhled statického obrázku parkoviště s vyznačenými parkovacími místy

### 2.2.2 REST API pro komunikaci se serverem

- **POST /parkoviste/kamera** – pošle data z kamery na server. Parametry v těle odesílaného objektu jsou:
  - „token“ – se generuje v rozhraní při tvorbě parkoviště (identifikuje kameru),
  - „password“ – se vytváří při tvorbě parkoviště (ověřuje oprávnění uživatele ukládat data do parkoviště),
  - „time“ – je čas odeslání dat na server,
  - „data“ – je hexadecimální řetězec s informací o obsazených parkovacích místech,
  - „imgSrc“ – je volitelný argument (obrázek ve formátu Base64).
- **POST /parkoviste/id** – vyhledá parkoviště podle „id“ skupiny.
- **POST /search** – vyhledá řetězec „data“ v názvu parkoviště.
- **POST /mapsearch** – vyhledá nejbližší parkoviště podle zadaných souřadnic. Parametry zahrnují vzdálenost v metrech „dist“ a řetězec formátu WGS84 se souřadnicemi „coords“.

- GET /csv\_pozice/id? – uloží CSV soubor pro parkoviště s daným „id“.

## 2.3 Testování a srovnání metod klasifikace

V této kapitole byly použity datové sady PKLot [21], CNRPark+EXT [22], FELslot [28] a vlastní navržená sada CTUPARK3D pro trénink a testování klasifikátorů. CTUPARK3D byla použita ve dvou variantách: originální varianta exportovaná z programu Godot (CTUPARK3Da) a vylepšená varianta s upraveným histogramem snímků (CTUPARK3Db). Pro strojové učení byly použity Python knihovny SciPy, OpenCV a Ultralytics.

### 2.3.1 Použití sloučených příznaků HOG a HSV histogramu

V této kapitole byla srovnána kombinace příznaků HOG a HSV histogramu se samostatným HOG deskriptorem. Pro trénink klasifikátorů bylo použito prvních 1000 kladných a 1000 záporných snímků z CTUPARK3Da. Testování proběhlo na testovací části datové sady PKLot.

Vstupní data byla škálována na  $72 \times 48$  px. HOG deskriptor měl délku 1440 a histogramy odstínu a sytosti měly každý 8 sloupců, což znamená dohromady délku 16. Oba výsledné vektory deskriptorů byly normovány tak, aby měly stejnou váhu.

**Tabulka 2.2:** Výsledky klasifikace – porovnání HOG a HOG+HSV

| klasifikátor | přesnost [%] |         |
|--------------|--------------|---------|
|              | HOG          | HOG+HSV |
| <b>SVM</b>   | 73,61        | 86,81   |
| <b>SGD</b>   | 73,60        | 86,35   |
| <b>RF</b>    | 73,51        | 86,82   |
| <b>LR</b>    | 74,22        | 86,22   |

Výsledky v tab. 2.2 ukázaly, že HSV histogram vylepšil výkon trénovací datové sady.

### ■ 2.3.2 Testování vlivu počasí a světelných podmínek CNRPark+EXT

Vliv počasí na klasifikaci byl zkoumán pro klasifikátory RF, LR, SVM a SGD. Pro natrénování byly použity dva různé modely: CTUPARK3Db a FELslot. Výsledky v tabulkách byly získány testováním na datové sadě CNRPark+EXT, která byla pro tento účel rozdělena do menších podsložek podle denní doby a počasí.

Z tabulek 2.3 a 2.4 byla zjištěna přesnost pro klasifikátor RF, který si ve všech testech vedl nejlépe. Průměrné hodnoty byly vypočteny bez zahrnutí 18. hodiny, protože snímky v tomto čase byly pořízeny jen za deštivého počasí a hodnota by ovlivňovala celkový výsledek. Vyšších přesností v řádu jednotek % algoritmus dosahoval při použití kombinace HOG+HSV deskriptoru. Nejlepších výsledků bylo dosaženo v 7 hodin ráno, kdy snímky nebyly tolik přeexponované. U snímků, na kterých bylo slunečno se během odpoledne výrazně snížila přesnost.

**Tabulka 2.3:** Výsledky klasifikace RF s HOG+HSV – různé světelné podmínky  
(trénovací data – CTUPARK3Db)

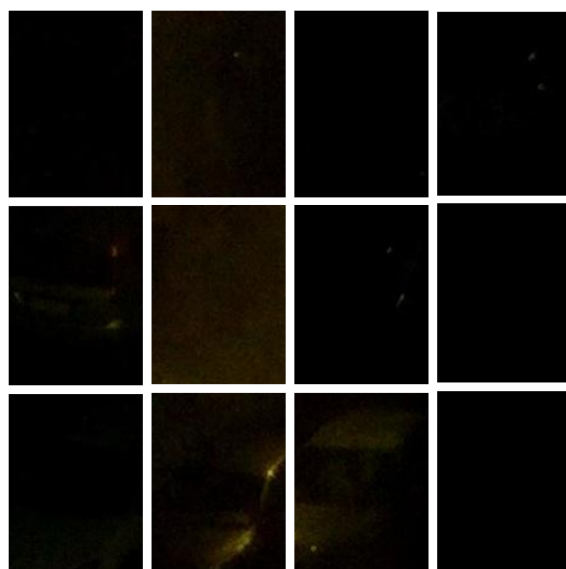
| CTUPARK3Db | přesnost [%]   |          |         |          |
|------------|----------------|----------|---------|----------|
|            | denní doba [h] | zataženo | deštivo | slunečno |
| 7          | 93,24          | 86,30    | 96,87   | 92,14    |
| 8          | 94,42          | 94,64    | 89,32   | 92,79    |
| 9          | 92,63          | 92,51    | 88,16   | 91,10    |
| 10         | 90,39          | 90,73    | 89,08   | 90,07    |
| 11         | 91,29          | 90,92    | 84,49   | 88,90    |
| 12         | 92,89          | 92,28    | 87,77   | 90,97    |
| 13         | 91,44          | 92,99    | 85,84   | 90,09    |
| 14         | 92,62          | 91,59    | 85,97   | 90,06    |
| 15         | 91,35          | 92,01    | 85,73   | 89,70    |
| 16         | 93,23          | 85,92    | 91,75   | 90,30    |
| 17         | 66,88          | 79,55    | 84,66   | 77,03    |
| 18         | —              | 57,14    | —       | —        |
| průměr     | 90,04          | 89,94    | 88,15   | —        |

**Tabulka 2.4:** Výsledky klasifikace RF s HOG – různé světelné podmínky (trénovací data – CTUPARK3Db)

| CTUPARK3Db | přesnost [%] |         |          |        |
|------------|--------------|---------|----------|--------|
|            | zataženo     | deštivo | slunečno | průměr |
| 7          | 94,69        | 92,12   | 96,78    | 94,53  |
| 8          | 93,46        | 93,41   | 92,72    | 93,20  |
| 9          | 91,02        | 89,20   | 86,08    | 88,77  |
| 10         | 87,72        | 88,34   | 84,19    | 86,75  |
| 11         | 87,39        | 86,79   | 84,89    | 86,36  |
| 12         | 88,59        | 88,03   | 83,78    | 86,80  |
| 13         | 88,64        | 88,65   | 82,41    | 86,57  |
| 14         | 88,87        | 88,65   | 84,56    | 87,36  |
| 15         | 88,17        | 87,30   | 83,55    | 86,34  |
| 16         | 87,90        | 86,36   | 87,65    | 87,30  |
| 17         | 65,92        | 80,43   | 82,60    | 76,32  |
| 18         | —            | 57,14   | —        | —      |
| průměr     | 87,49        | 88,11   | 86,29    | —      |

Snímků při zataženém počasí v 18. hodině bylo málo a byly velmi tmavé (viz obr. 2.2). Algoritmus HOG v těchto případech nedokázal detekovat správné hrany a vedlo to k tak výraznému snížení přesnosti. Významné zlepšení v počtu nalezených relevantních hran umožnila ekvalizace histogramu u těchto snímků (viz obr. 2.3).





**Obrázek 2.2:** Ukázka pozitivních snímků z CNRPark+EXT (deštivo/18. hodina)



**Obrázek 2.3:** Vylepšení kontrastu snímku pomocí ekvalizace histogramu (vlevo – originální obrázek, vpravo – upravený)

Výsledky v tabulkách 2.5 a 2.6 ukázaly srovnání různých typů klasifikátorů. Bylo zde patrné, že kromě lineární regrese rozdíly nejsou příliš výrazné. Nejlépe si vedl algoritmus RF a dosahoval průměrné přesnosti okolo 90 % v závislosti na počasí.

**Tabulka 2.5:** Výsledky klasifikace HOG+HSV – srovnání průměrných výsledků jednotlivých klasifikátorů (trénovací data – CTUPARK3Db)

| CTUPARK3Db   | přesnost [%] |              |              |
|--------------|--------------|--------------|--------------|
| klasifikátor | zataženo     | deštivo      | slunečno     |
| RF           | 90,04        | <b>87,21</b> | 88,15        |
| LR           | 85,50        | 84,00        | 85,86        |
| SGD          | <b>90,41</b> | 83,96        | 89,00        |
| SVM          | 88,26        | 85,85        | <b>89,37</b> |

**Tabulka 2.6:** Výsledky klasifikace HOG – srovnání průměrných výsledků jednotlivých klasifikátorů (trénovací data – CTUPARK3Db)

| CTUPARK3Db   | přesnost [%] |              |              |
|--------------|--------------|--------------|--------------|
| klasifikátor | zataženo     | deštivo      | slunečno     |
| RF           | <b>89,43</b> | <b>86,99</b> | <b>86,87</b> |
| LR           | 82,79        | 82,12        | 82,67        |
| SGD          | 87,12        | 85,26        | 85,31        |
| SVM          | 87,21        | 84,70        | 85,98        |

V tabulkách 2.7 a 2.8 byly pro porovnání výsledky testování při natrénovaném modelu na datové sadě FELslot, která byla vytvořena na skutečných parkovištích. K natrénování bylo použito stejné množství dat (2000 snímků). Výsledky tohoto modelu byly o několik jednotek % bodů nižší a naopak zde dosahovala lepších výsledků lineární regrese. Tento výsledek ukázal, že modelování datových sad může mít výhodu při trénování modelů. Kromě toho model obsahuje i snímky aut s nápisy a logy. Stejně tak by bylo možné ho připravit i na různé speciální případy, které se na silnicích běžně nevyskytují, ale mohly by.

**Tabulka 2.7:** Výsledky klasifikace HOG+HSV – srovnání průměrných výsledků jednotlivých klasifikátorů (trénovací data – FELslot)

| FELslot      | přesnost [%] |              |              |
|--------------|--------------|--------------|--------------|
| klasifikátor | zataženo     | deštivo      | slunečno     |
| RF           | <b>85,14</b> | <b>83,15</b> | <b>83,98</b> |
| LR           | 84,82        | 79,77        | 83,43        |
| SVM          | 82,84        | 79,54        | 83,50        |
| SGD          | 83,20        | 80,13        | 83,24        |

**Tabulka 2.8:** Výsledky klasifikace HOG – srovnání průměrných výsledků jednotlivých klasifikátorů (trénovací data – FELslot)

| FELslot      | přesnost [%] |              |              |
|--------------|--------------|--------------|--------------|
| klasifikátor | zataženo     | deštivo      | slunečno     |
| RF           | 83,20        | 80,13        | <b>83,24</b> |
| LR           | <b>85,99</b> | <b>83,13</b> | 81,65        |
| SVM          | 78,01        | 76,43        | 76,36        |
| SGD          | 80,89        | 78,18        | 77,59        |

### 2.3.3 Testovací sada PKLot

V tomto oddíle byla ověřena funkčnost modelu na testovací sadě PKLot a zobrazeny výsledky v tab. 2.9. V záhlaví tabulky jsou uvedeny různé datové sady, které byly použity pro trénování klasifikátoru. Byl otestován klasifikátor SVM s příznaky HOG+HSV a YOLOv8 s předtrénovaným modelem YOLOv8n na datech CTUAPARK3Db.

Z tabulky 2.9 ukázala, že CTUPARK3Db měla horší výsledky než CNR-Park+EXT při testování na PKLot. To by mohlo naznačovat, že způsob získávání dat měl velký vliv na přesnost. Na obrázku z PKLot 2.4 bylo vidět, že jedno parkoviště bylo fotografované skoro shora. Naopak data z CNR-Park+EXT byla fotografovaná spíše ze strany (viz obr. 2.5).

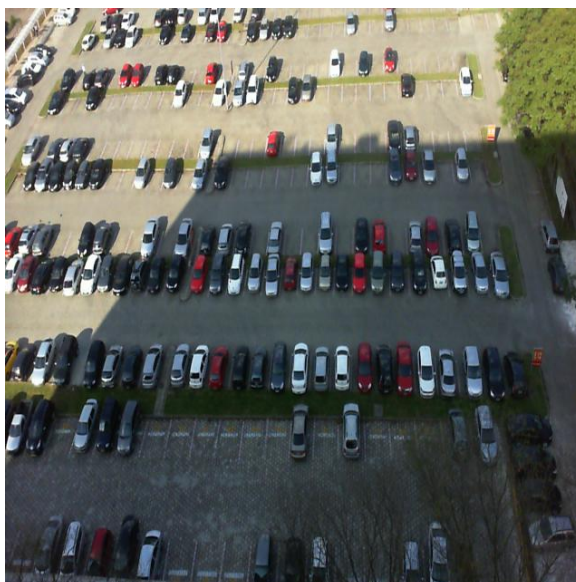
Data CTUPARK3Db byla upravena tak, aby měla stejný histogram jako dva vybrané obrázky z PKLot (jeden s volným a jeden s obsazeným místem). Tím byla vytvořena datová sada CTUPARK3Dc, která zvýšila přesnost

trénovaného modelu na 86,36 %.

S použitím dat CTUPARK3Db byla porovnána metoda YOLOv8 pro klasifikaci využívající hluboké učení s metodami využívající ke klasifikaci extrakci příznaků. Po trénování těmito daty se přesnost modelu zlepšila o 14,8 %. Náš model byl velmi malý a rychlý, klasifikoval jedno parkovací místo za průměrně jen 10,1 ms.

**Tabulka 2.9:** Výsledky klasifikace na testovacích datech PKLot

| FELslot | přesnost [%] |            |                         |
|---------|--------------|------------|-------------------------|
|         | CTUPARK3Db   | CTUPARK3Dc | CTUPARK3Db<br>(YOLOv8n) |
| 80,23   | 77,15        | 86,36      | 91,95                   |



**Obrázek 2.4:** Ukázka snímku z datové sady PKLot



**Obrázek 2.5:** Ukázka snímku z datové sady CNRPark+EXT

### ■ 2.3.4 Použití transformace při předzpracování snímku

V této práci byl zkoumán vliv úpravy perspektivy na přesnost klasifikace. Tento přístup vyžadoval speciální anotaci dat, která nevybírání obdélníkové tvary, ale čtyřúhelníky, kterými vozidlo „obkreslí“. Byl vytvořen nástroj na webovém rozhraní, který umožnil anotovat data tímto způsobem. Takto bylo anotováno 100 míst na jednom z parkovišť PKLot a tyto čtyřúhelníky byly poté transformovány na obdélníky v rámci předzpracování. Stejný postup byl proveden i s trénovací sadou. Výsledky dosahovaly 100% přesnosti u SGD a RF algoritmu pro HOG+HSV deskriptor.

### ■ 2.3.5 Vliv velikosti datové sady na parametry modelu

V rámci práce bylo provedeno rozsáhlé testování přesnosti klasifikace v závislosti na délce datové sady určené pro trénink. Z výsledků vyplývalo, že přesnost závisela spíše na výběru dat a kvalitě použitých dat a extrahovaných příznaků než na velikosti datové sady. Někdy byl model natrénovaný s 50 snímky lepší než model s 1000 snímky a jindy naopak, pokud byly snímky vybírány náhodně. Nelze tedy s jistotou říci, že velikost datové sady je klíčovým parametrem pro testování SVM klasifikátoru.

## ■ 2.4 Kontrola délky parkování vozidel

Jedním ze způsobů, jak zjistit, zda auto stojí na parkovacím místě příliš dlouho, bylo použití segmentace a sledování objektu. Tato metoda by vyžado-

vala detekci každého vozidla. Po detekci vozidla by bylo možné sledovat jeho pohyb a sbírat časové údaje o jeho poloze. Tyto údaje by se poté daly použít k výpočtu odhadu doby stání na parkovišti. Bylo by důležité, aby tato metoda byla velmi přesná pro správné vyhodnocení všech stojících automobilů. Navíc tato metoda nezjišťuje, zda má automobil platný parkovací lístek. To dnes řeší parkovací roboti, kteří jezdí po parkovišti a detekují poznávací značky automobilů a čtou jejich obsah pomocí OCR (optického rozpoznávání znaků).

Obrázky 2.6, 2.7 a 2.8 ukazují srovnání méně pokročilých metod detekce s detekčním algoritmem YOLOv8. Na prvním obrázku byla metoda zaměřena na detekci obrysů v obraze, která následně vyhodnocovala jejich tvar, velikost a poměr. Na obrázku bylo vidět dobře vyhodnocený obrys. Problém byl v tom, že mohlo být vyhodnoceno více podobných obrysů jiné části vozidla a je třeba správně vyhodnotit ten správný. To dělá tuto metodu nespolehlivou.



**Obrázek 2.6:** Detekce SPZ pomocí obrysů

Pokročilejší možností bylo použití HAAR deskriptoru pro trénování kaskádového klasifikátoru. HAAR deskriptor popisoval rohy v obraze a dokázal detekovat různé tvary. Na obrázku bylo vidět, že ohraničující obdélník vyhodnocený kaskádovým klasifikátorem byl příliš velký, a to mohlo způsobit větší chybovost při následném použití OCR. K testování byl využit předem natrénovaný model, který je součástí knihovny OpenCV.



**Obrázek 2.7:** Detekce SPZ pomocí HAAR příznaků (rohý) a kaskádového klasifikátoru

Poslední metoda použila dotrénovaný model YOLOv8 s daty SPZ staženými z Google obrázků a anotovanými v programu CAVT. Byl použit nejmenší předtrénovaný model Nano, který lze provozovat i na mikropočítačích jako je Raspberry. Na obrázku bylo vidět, že detekce ohraničujícího obdélníku byla velmi přesná. Model rozpoznával velmi dobře zahraniční i české typy poznávacích značek i ve větší vzdálenosti.



**Obrázek 2.8:** Detekce SPZ pomocí algoritmu YOLOv8 naučeného na vlastních datech





## Závěr

Cílem této práce bylo vytvořit prototyp inteligentního kamerového systému pro sledování parkovacích míst v reálném čase. Systém se skládal z kamery, mikropočítače Raspberry Pi Zero 2 a webového rozhraní. Kamerový systém používal algoritmus SVM a příznaky založené na HOG pro klasifikaci obrazových dat. Klasifikovaná data byla přenášena pomocí sítě SigFox. Webové rozhraní zobrazovalo tato data a další základní statické informace o parkovišti. Prototyp dosahoval maximální přesnosti pouhých 83 % v reálných podmínkách během několika týdnů testování. Tento výsledek byl ovlivněn tím, že model nebyl natrénován na datech z testovaného parkoviště, ale na trénovací sadě FELslot. Doba klasifikace celého menšího parkoviště s 17 místy se pohybovala kolem jedné sekundy. Bylo zjištěno, že použití jazyka C++ místo Pythonu pro extrakci deskriptoru HOG vedlo k přibližně pětinasobnému zvýšení rychlosti.

V práci byla také provedena komparace různých klasifikačních metod. Byly použity dvě veřejně dostupné testovací datové sady: PKLot a CNRPark+EXT. Datová sada CNRPark+EXT sloužila k otestování vlivu osvětlení na různé klasifikátory, jako jsou: RF, LR, SVM a SGD. Pro natrénování modelů byly použity dvě vlastní datové sady: CTUPARK3D, která byla vytvořena pomocí programu pro 3D grafiku, a FELslot, která byla vytvořena během diplomové práce Ing. Petrem Mělničukem na fakultě elektrotechnické ČVUT. Cílem bylo ověřit, zda modely natrénované na těchto syntetických datech mohou dosahovat srovnatelné přesnosti.

Výsledky prokázaly, že přesnost klasifikace byla zlepšena přidáním histogramu HSV (odstínu a sytosti) do deskriptoru pro trénování většinou v řádu jednotek %. Přesnost byla vyšší i na obrázcích, které byly získány za špatných

světelných podmínek. Dalším zjištěním bylo, že použití algoritmu HOG pro popis trénovací sady s obrázky s nízkým kontrastem (špatnými světelnými podmínkami) snižovalo přesnost natrénovaného modelu. Začlenění obrázků se špatnými světelnými podmínkami (příliš velká expozice nebo nedostatek světla) do trénovací sady, nevedlo ke zlepšení přesnosti při klasifikaci obrázků se špatnými světelnými podmínkami. Algoritmus HOG nedokázal detekovat na tmavších nebo přeexponovaných obrázcích dostatek hran. Tento fakt je potvrzen i nižší přesností u obrázků na kterých bylo slunečno a ve tmě, které vykazovaly tyto extrémy. Bylo zjištěno, že při použití HOG deskriptoru je vhodné upravit obraz tak, aby zvýšil kontrast snímku např. ekvalizací histogramu pro nalezení více relevantních hran.

Významné zlepšení přesnosti bylo dosaženo také změnou perspektivy jednotlivých výřezů a odstraněním vlivu jejich rotace a zkreslení. Nevýhodou tohoto přístupu byla potřeba anotace dat, proto byl počet testovaných dat pouze 200 obrázků z datové sady PKLot. Na natrénování bylo anotováno 1000 obrázků ze sady CTUPARK3D. Některé klasifikátory pomocí tohoto postupu dosahovaly přesnosti 100 %. Nicméně pro spolehlivé zhodnocení výsledku tohoto přístupu by bylo nutné otestovat algoritmus na více obrázcích, tedy anotovat větší množství dat, z více datových sad.

V práci byl také otestován model založený na hlubokém učení pro klasifikaci a detekci YOLOv8. Bylo prokázáno, že model dosahoval přesnosti o 14,8 % vyšší na testovací sadě PKLot, při použití dat CTUPARK3Db. Doba klasifikace byla průměrně 10,1 ms na jedno parkovací místo (testy provedeny na procesoru Intel i5-8250u). Dále byly ukázány výsledky detekce tímto algoritmem pro určení ohraničujícího boxu kolem poznávacích značek vozidel sloužící pro jejich identifikaci.

Budoucí práce by se měla věnovat vlivu kvality datové sady na výsledky klasifikace a detekce. Dále také na použití pokročilých metod hlubokého učení, jako jsou konvoluční neuronové sítě např. v práci porovnávaný algoritmus YOLO, který umožňuje dnes detekovat objekty rychlostí jednotek FPS i na mikropočítačích typu Raspberry, což je činí velmi dobrými konkurenty metodám založených na extrakci příznaků. Tyto algoritmy dosahují vysoké přesnosti a robustnosti klasifikace i v obtížných podmínkách. Zároveň umožňují i sledo-

vání objektů, které by mohlo v kombinaci s vysokou přesností detekce vozidel, kterou tyto sítě mají, kontrolovat dobu parkování jednotlivých vozidel na parkovišti pro omezenou dobu stání typu K+R.





## Literatura

- [1] Qualcomm Developer Network. In: *Classification, Object Detection and Image Segmentation* [online]. [cit. 2023-05-24]. Dostupné z: <https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk/learning-resources/image-segmentation-deeplab-neural-processing-sdk/classification-object-detection-segmentation>
- [2] ALMEIDA, Paulo, Jeovane H. ALVES, Rafael S. PARPINELLI a Jean P. BARDDAL. A systematic review on computer vision-based parking lot management applied on public datasets. *Expert Systems with Applications*. 2022, **2022**(198). ISSN 0957-4174. Dostupné z: doi:10.1016/j.eswa.2022.116731
- [3] AHRNBOM, Martin, Kalle ASTROM a Mikael NILSSON. Fast Classification of Empty and Occupied Parking Spaces Using Integral Channel Features. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2016, 2016, 1609-1615. ISBN 978-1-5090-1437-8. Dostupné z: doi:10.1109/CVPRW.2016.200
- [4] VARGHESE, Arun a G. SREELEKHA. An Efficient Algorithm for Detection of Vacant Spaces in Delimited and Non-Delimited Parking Lots. *IEEE Transactions on Intelligent Transportation Systems*. 2020, **21**(10), 4052-4062. ISSN 1524-9050. Dostupné z: doi:10.1109/TITS.2019.2934574
- [5] ALMEIDA, Paulo, Luiz S. OLIVEIRA, Eunelson SILVA, Alceu BRITTO a Alessandro KOERICH. Parking Space Detection Using Textural Descriptors. *2013 IEEE International Conference on Systems, Man, and*

- Cybernetics*. IEEE, 2013, 2013, 3603-3608. ISBN 978-1-4799-0652-9. Dostupné z: doi:10.1109/SMC.2013.614
- [6] DE ALMEIDA, Paulo R.L., Luiz S. OLIVEIRA, Alceu S. BRITTO, Eunelson J. SILVA a Alessandro L. KOERICH. PKLot – A robust dataset for parking lot classification. *Expert Systems with Applications*. 2015, **42**(11), 4937-4949. ISSN 09574174. Dostupné z: doi:10.1016/j.eswa.2015.02.009
- [7] VÍTEK, Stanislav a Petr MELNIČUK. A Distributed Wireless Camera System for the Management of Parking Spaces. *Sensors*. 2018, **18**(1). ISSN 1424-8220. Dostupné z: doi:10.3390/s18010069
- [8] BOHUSH, Rykhard, Sergey ABLAMEYKO, Tatiana KALGANOVA a Pavel YARASHEVICH. Extraction of image parking spaces in intelligent video surveillance systems. In: *Machine Graphics and Vision*. 2018. ISSN 2720-250X. Dostupné z: doi:10.22630/MGV.2018.27.1.3
- [9] AMATO, Giuseppe, Paolo BOLETTIERI, Davide MORONI, et al. A Wireless Smart Camera Network for Parking Monitoring. *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, 2018, 1-6. ISBN 978-1-5386-4920-6. Dostupné z: doi:10.1109/GLOCOMW.2018.8644226
- [10] MORA, Jhon E.G., Juan C.L. LOPERA a Diego A.P. PATIÑO CORTES. Automatic Visual Classification of Parking Lot Spaces: A Comparison Between BoF and CNN Approaches. In: *Applied Computer Sciences in Engineering*. Cham: Springer International Publishing, 2018, 2018-09-13, s. 160-170. Communications in Computer and Information Science. ISBN 978-3-030-00349-4. Dostupné z: doi:10.1007/978-3-030-00350-0\_14
- [11] HIRANMAI, M, Niranjana KRUPA B a H K NAGARAJ. Comparative Study of Various Feature Extraction Techniques for Pedestrian Detection. *Procedia Computer Science*. 2019, 154, 622-628. ISSN 18770509. Dostupné z: doi:10.1016/j.procs.2019.06.098
- [12] YATES, Darren a Md Zahidul ISLAM. FastForest: Increasing random forest processing speed while maintaining accuracy. *Information Sciences*. 2021, **557**, 130-152. ISSN 00200255. Dostupné z: doi:10.1016/j.ins.2020.12.067

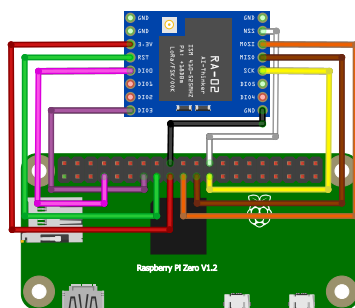
- [13] LARADJI, Issam H., Negar ROSTAMZADEH, Pedro O. PINHEIRO, David VAZQUEZ a Mark SCHMIDT. Where Are the Blobs: Counting by Localization with Point Supervision. *Computer Vision – ECCV 2018*. Cham: Springer International Publishing, 2018, 2018-10-09, 560-576. Lecture Notes in Computer Science. ISBN 978-3-030-01215-1. Dostupné z: doi:10.1007/978-3-030-01216-8\_34
- [14] KIM, Daniel. *Sobel Operator and Canny Edge Detector ECE 480 Fall 2013*. 2013, 1-10.
- [15] HDIOUD, Boutaina, Mohammed EL HAJ TIRARI, Rachid OULAD HAJ THAMI a Rdouan FAIZI. Detecting and Shadows in the HSV Color Space Using Dynamic Thresholds. In: *Bulletin of Electrical Engineering and Informatics*. 2018, s. 70-79. ISSN 2302-9285. Dostupné z: doi:10.11591/eei.v7i1.893
- [16] TERVEN, Juan a Diana CORDOVA-ESPARZA. *A Comprehensive Review of YOLO: From YOLOv1 and Beyond*. **2023**, 20-25. Dostupné z: doi:arXiv:2304.00501v2
- [17] WANG, Senzhang, Zhoujun LI, Chunyang LIU, Xiaoming ZHANG a Haijun ZHANG. Training data reduction to speed up SVM training. *Applied Intelligence*. 2014, **41**(2), 405-420. ISSN 0924-669X. Dostupné z: doi:10.1007/s10489-014-0524-2
- [18] BOWYER, Kevin W., Nitesh V. CHAWLA, Lawrence O. HALL a W. P. KEGELMEYER. SMOTE: Synthetic Minority Over-sampling Technique. *CoRR*. 2011, (abs/1106.1813). Dostupné z: doi:arXiv:1106.1813v1
- [19] HAIBO HE, YANG BAI, Edwardo A. GARCIA a SHUTAO LI. ADA-SYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, 2008, 1322-1328. ISBN 978-1-4244-1820-6. Dostupné z: doi:10.1109/IJCNN.2008.4633969
- [20] TANAKA, Fabio H. K. d. S. a Claus ARANHA. Data Augmentation Using GANs. *CoRR*. **2019**(abs/1904.09135). Dostupné z: doi:arXiv:1904.09135

- [21] DE ALMEIDA, Paulo R.L., Luiz S. OLIVEIRA, Alceu S. BRITTO, Euelson J. SILVA a Alessandro L. KOERICH. PKLot – A robust dataset for parking lot classification. *Expert Systems with Applications*. 2015, **42**(11), 4937-4949. ISSN 09574174. Dostupné z: doi:10.1016/j.eswa.2015.02.009
- [22] A Dataset for Visual Occupancy Detection of Parking Lots. In: *CNR-Park+EXT* [online]. [cit. 2023-05-26]. Dostupné z: <http://cnrpark.it/>
- [23] BONDAREV, Anton a Alexander KALMUK. Benchmarking OpenCV on STM32 MCUs. *Embedded* [online]. [cit. 2023-05-25]. Dostupné z: <https://www.embedded.com/benchmarking-opencv-on-stm32-mcus/>
- [24] Jetson TX1 Module. *NVIDIA Developer* [online]. [cit. 2023-05-25]. Dostupné z: <https://developer.nvidia.com/embedded/jetson-tx1>
- [25] VAN DE PAAR, ROEL. Raspberry Pi 4: Good Enough For Gaming?. *How-To Geek* [online]. [cit. 2023-05-25]. Dostupné z: <https://www.howtogeek.com/devops/raspberry-pi-4-good-enough-for-gaming/>
- [26] Obecná dokumentace Datové platformy. *Golemio* [online]. [cit. 2023-05-26]. Dostupné z: <https://operator-ict.gitlab.io/golemio/documentation/>
- [27] MojePrahaAPI. *Moje Praha* [online]. [cit. 2023-05-26]. Dostupné z: <https://mojepraha.eu/apispecs>
- [28] MELNIČUK, Petr. Distribuovaný kamerový systém pro správu parkovacích míst. ČVUT, 2017

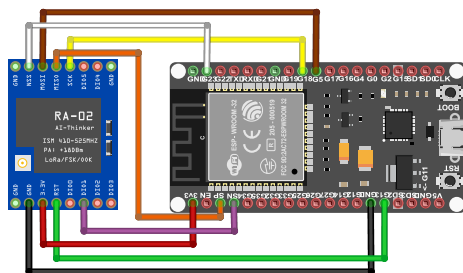


## Příloha A

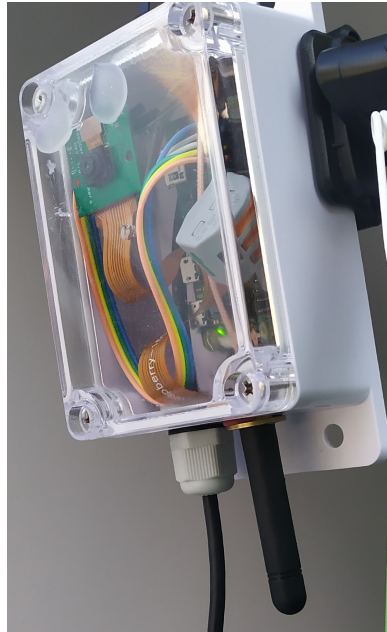
### Výsledný testovaný kamerový modul



Obrázek A.1: Zapojení LoRa modulu (Ra-02) k Raspberry Pi Zero 2 (vysílač)



Obrázek A.2: Propojení LoRa modulu (Ra-02) a ESP32 (přijímač a brána do sítě Internet)



**Obrázek A.3:** Testovaný kamerový modul využívající Raspberry Pi Zero 2 a SigFox