BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DETECTION AND CLASSIFICATION OF VEHICLES FOR EMBEDDED PLATFORMS

DETEKCE A KLASIFIKACE VOZIDEL PRO VESTAVĚNÉ PLATFORMY

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **PATRIK SKALOŠ**
AUTOR PRÁCE

SUPERVISOR **Ing. JAKUB ŠPAŇHEL**
VEDOUCÍ PRÁCE

BRNO 2023

Abstract
An abstract of the work in English will be written in this paragraph.

Abstrakt

Keywords
Here, individual keywords separated by commas will be written in English.

Klíčová slova

Reference
SKALOŠ, Patrik. *Detection and Classification of Vehicles for Embedded Platforms.* Brno, 2023. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Jakub Špaňhel

Detection and Classification of Vehicles for Embedded Platforms

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mr. X The supplementary information was provided by Mr. Y I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

....
Patrik Skaloš
January 29, 2023

Acknowledgements

Here it is possible to express thanks to the supervisor and to the people which provided professional help (external submitter, consultant, etc.).

Contents

List of Figures

Chapter 1

Introduction

Vehicle detection in real-time is crucial for enhancing traffic safety and flow. It can be used to manage traffic at an intersection using traffic lights, spot speeding or wrong-way drivers, enhance route planning to ease congestion, alert drivers to potentially dangerous situations, and offer insightful information about driving behavior.

There are various methods for detecting vehicles, each of which has its advantages and limitations. In this paper, we focus on detection using a camera-based traffic surveillance system. This approach was chosen because of its low price, versatility, and ease of installation.

There are two main techniques for vehicle detection using a camera: image processing and convolutional neural networks (CNNs). Both approaches are based on analyzing images to identify patterns and features to detect vehicles. Image processing techniques involve applying a series of pre-defined filters and transformations to an image to extract relevant information. CNNs, on the other hand, are a type of machine learning algorithm that learns to recognize patterns and features through training. CNNs are generally more effective than image processing techniques for vehicle detection, as they can learn to recognize complex patterns and features that may be difficult to extract using pre-defined filters and transformations, but they have traditionally required a significant amount of computing power, commonly provided by a graphical processing unit (GPU). However, recent developments suggest that it may now be possible to perform CNN-based tasks in real-time with reduced processing requirements.

In this paper, we attempt to train a CNN with a smaller architecture that is able to run on a single central processing unit (CPU) instead of a GPU. This would allow the system to process the camera feed on-site, rather than relying on its transmission over the internet and remote processing. This would reduce the cost of the system and the amount of network bandwidth required, as well as expand the range of purposes for which the system can be utilized.

Chapter 2

Common Approaches to Vehicle Detection

This chapter contains a summary of commonly used solutions for the problem of vehicle detection. First, we introduce systems that are not based on cameras, and then we compare CNNs with image processing techniques.

2.1 Vehicle Detection Without a Camera

Magnetic Sensors

One of the simplest solutions to detect vehicles is to use an induction loop.[1] The sensor, composed of a single wire, can be buried under concrete and detect the presence of metal objects passing by. With a controller, induction loops typically provide data about vehicle presence, but using more advanced algorithms, speed, approximate classification and much more can be determined from this simple sensor. Although the design is very simple, installation is not and induction loops can even get damaged over time, while repairs call for temporary shutdowns of roads. It is worth noting that there are many other types of magnetic sensors, which can provide more detailed and accurate data with simpler sensor installation, but the idea stays the same.

Ultrasonic Sensors

Another type of inexpensive and simple detector is an ultrasonic sensor.[4] These sensors can operate in a variety of conditions and are typically mounted on existing infrastructure. These distance measurement sensors are usually used to detect the presence and distance of nearby vehicles and their speed, but they can be utilized in many different ways to even provide a simple shape of a vehicle to classify it.

Radars and Lidars

Radars (Radio Detection and Ranging), which can also be mounted on existing infrastructure above ground, work similarly to ultrasonic sensors, but a single radar can oversee a much wider area of a road.[2] They are usually used to detect the presence, speed, heading and shape of a vehicle. The shape can then be used to predict a class of the vehicle. Lidars

(Light Detection and Ranging) can be used in a similar way, but are far more accurate, which makes them better at detecting shapes and locations of objects.

2.2 Vehicle Detection With a Camera

Camera-based systems, which are relatively inexpensive, can be mounted on existing infrastructure, do not emit energy and are highly versatile, while providing a large amount of data. In addition to detecting the presence, speed, heading and shape of a vehicle, these systems can also provide its color, license plate and countless other characteristics.

There are two main techniques for vehicle detection using a camera: image processing and convolutional neural networks (CNNs). Both approaches are based on analyzing images to identify patterns and features to detect vehicles. Image processing techniques involve applying a series of pre-defined filters and transformations to an image to extract relevant information. CNNs, on the other hand, are a type of machine learning algorithm that learns to recognize patterns and features through training. CNNs are generally more effective than image processing techniques for vehicle detection, as they can learn to recognize complex patterns and features that may be difficult to extract using pre-defined filters and transformations. However, image processing can still be extremely helpful for simpler tasks or as a component of a larger vehicle detection system. Additionally, CNNs typically require a lot more processing power.

Image Processing Techniques

A huge amount of research has been conducted on using image processing to solve the problem of vehicle detection.[5] Many of the techniques proposed can operate in real-time and are very reliable in typical environmental conditions. However, they can be sensitive to changes in illumination or have their performance affected by rain, snow, shadows, occlusions, or noise. While there are methods for addressing some of these challenges, such as shadow removal techniques[3], they add to the computational requirements of the system. Overall, our research suggests that the image processing approach is well-suited for simpler tasks or systems with lower accuracy requirements, but it is often difficult to implement and may lack versatility in more complex or demanding scenarios.

Convolutional Neural Networks

Chapter 3

úvod do problematiky

3.1 Convolutional Neural Networks

Chapter 4

Related Work

4.1 Object Detection?

4.2 Light-weight Detectors

4.3 Vehicle Detection

Chapter 5

Riešenie

5.1 Datasets

Big and high-quality datasets are very important when training a CNN-based detector. In this section, used datasets are listed and analyzed. First, the criteria for recognizing an appropriate dataset for our task are explained. Each chosen dataset is then analyzed individually. Finally, a summary of all used datasets is provided.

[[Napísať v akom formáte sú datasety a do akého formátu to processujem? A že python skriptom? Možno samostatný subsection?]]

5.1.1 Dataset Criteria

Here, we briefly explain the most important criteria for selecting datasets to be used in a project like this.

Camera Angle

Since we're building a detector for a camera mounted on infrastructure, it is recommended to use datasets containing surveillance-type images. Datasets containing only images taken from, for example, a car dashboard camera, were therefore disregarded.

Classes

If we don't want to re-label the dataset manually, its classes must be mappable to *our* classes. In this work, 8 object classes are considered:

- Pedestrian
- Bicycle
- Motorcycle (any two-wheeled motorized vehicle)
- Passenger Car
- Transporter (or a van, pick-up truck, etc.)
- Bus (including a minibus)
- Truck

- Unknown

Car trailers are ignored and considered a part of the vehicle they are attached to.

Diversity of Images

For the trained detector to generalize well, it's important for a dataset to contain images with different camera angles, lighting conditions, weather, etc.

Dataset Quality

We observed that many datasets contained incorrect annotations or classifications. It is important to check the dataset and either fix faulty annotations, ignore incorrectly annotated images or even disregard the whole dataset.

5.1.2 Individual Datasets

This subsection individually analyses all datasets used in this work. Table 5.1 compares these datasets on a higher level for an overview. Example frames of each dataset are shown in figure **[[ref]]**.

Miovision Traffic Camera Dataset

The MIO-TCD (Miovision traffic camera dataset) **[[cite]]** is the biggest and most important dataset for this work, consisting of two parts: *Classification dataset* and *Localization dataset*. Only the *Classification* part is used here, containing 648959 annotations in 137743 images, reduced to 110000 images by the fact that the *Test* subset is not annotated. Images are taken from many different points of view. Roughly 79% of all images are of resolution `720x480`, but the image quality seems to be lower. The rest is of resolution `342x228`. **[[ref na example fotku]]**

Although this dataset is big, it probably wouldn't be enough by itself. The images are of low quality, there aren't many pedestrians, bicycles and motorcycles, and more pictures with different weather and lighting conditions are needed.

AAU RainSnow Traffic Surveillance Dataset

Another important dataset is the *AAU RainSnow* dataset **[[cite]]** . The authors mounted two synchronized (one RGB and one thermal) cameras on street lamps at seven different Danish intersections to take 5 minute long videos at different lighting and weather conditions - night and day, rain and snow. They then extracted 2200 frames from the videos and annotated them on a pixel-level. Several different types of masks were also created and included in the dataset.

In our work, we only use the annotated frames from the RGB camera, containing 13297 annotations in 2200 frames of resolution `640x480`. **[[ref na example fotku]]**

The dataset uses these 6 classes:

- Pedestrian
- Bicycle
- Motorbike

- Car
- Bus
- Truck

This creates a problem - vehicles of our internal class `transporter` (small van) don't have their own class in the dataset, but are classified as trucks. **[[ako som to vyriešil?]]**

Several other minor problems were found when processing this dataset:

- Frames in groups `Egensevej-1`, `Egensevej-3` and `Egensevej-5` are hardly usable because of the low-quality camera and challenging weather and lighting conditions, so they were dismissed
- Some frames had bounding boxes over the whole frame - this is most certainly an annotation error. These frames were ignored as well
- Objects outside the area of interest were not annotated, so a mask had to be used to remove these objects from the frames
- The mask for `Hadsundvej` intersection didn't fully cover the area that should be ignored. This was fixed by simply editing the mask

Multi-View Traffic Intersection Dataset

For this dataset, the authors **[[cite]]** recorded one intersection from two points of view - one camera was mounted on existing infrastructure and one was attached to a hovering drone. The dataset contains roughly 65000 annotated objects in approximately 5500 frames (equal share of frames for both cameras). **[[ref na example fotku]]**

The frames are extracted from a continuous video (30FPS) and there's often no difference between two consecutive frames, so the effective amount is much lower. Therefore, to make training more effective, only every third **[[update?]]** frame is considered when processing this dataset.

All annotated objects fall into one of four classes:

- Bicycle
- Car
- Bus
- Lorry

This, at first, might not seem like enough, but a closer inspection of the annotated frames reveals that there are no pedestrians or motorcycles. However, although there are vans (`transporters`) in the frames, they are classified as trucks.

When processing this dataset, two other problems were encountered:

- Vehicles that are not on the road are not annotated, so they have to be masked out. This is not as easy for the drone video because the camera is moving, but it is still simple enough
- Many frames of the drone footage are not annotated and have to be ignored

Night and Day Instance Segmented Park Dataset

Another useful dataset is the *NDISPark* **[[cite]]** , which contains images of parked vehicles taken by a camera mounted on infrastructure. Although the dataset only contains 101 frames after processing, there are 1904 objects annotated in total. This still makes it a tiny dataset, but it provides images of vehicles from many different points of view and also contains many occluded vehicles. **[[ref example fotku]]** Additionally, all frames are 2400px in width and within 908px and 1808px in height.

This dataset does not contain any classifications, but luckily, it only contains pedestrians, cars, `transporters` and a few unattached car trailers. When processing, frames containing pedestrians are therefore ignored and all `transporters` are manually classified. Unattached car trailers had to be manually removed since we don't consider them as separate vehicles. **[[Pravda?]]**

VisDrone Dataset

The VisDrone dataset **[[cite]]** is very different from all the previous datasets since it doesn't just contain traffic surveillance images. It contains many objects per frame from many different points of view - 68726 objects in just 1610 frames taken by a camera mounted on a drone.

We think it might be a helpful addition to our datasets mainly because it contains many annotated people from different points of view (lacking by previous datasets).

However, a problem was encountered when checking the dataset: a person standing is classified as a pedestrian and a person sitting is classified as a person. That meant we had to ignore the `person` class, because otherwise, people sitting on a motorcycle would be annotated, too, and that is an unwanted behavior. This, however, means we will be teaching the detector to ignore sitting people.

Additionally, objects in this dataset are classified into 12 classes, which can be easily mapped to our 8 *internal* classes.

5.1.3 Summary of Datasets

In Table 5.1, we compare used datasets on a higher level and show the number of images and instances contained. We also display the number of classes used by the authors and some short comments. It is clear that the MIO-TCD **[[ref?]]** dataset amounts for most of our data, but we believe that the other datasets are crucial for training a good detector, since their images feature a greater variety of lighting and weather conditions, points of view and they contain more annotated pedestrians and two-wheeled vehicles.

In Table 5.2 we show the number of annotated object instances per class in all used datasets combined. It is clear that passenger cars are most frequent, as they should be, but there are too few bicycles and motorcycles which raises concerns. **[[Solution? Viac na bicykle som nenašiel (na motocykle som moc nehľadal, aspoň teda nepamätám). Postačí to ak iba poriešim class balance?]]**

Dataset tag	# images	# instances	# classes	Comments
MIO-TCD	110000	349199	11	Large Low-quality images
AAU	1899	11013	6	Small Weather variation
MTID	1800	21674	4	Small Continuous video
NDISPark	101	1904	1	Small Occlusions
VisDrone	1610	68726	12	Small Many pedestrians
Total	*115314	452516	-	

Table 5.1: * číslo je menšie ako sum lebo kvôli mmdetection middle formátu musím ignorovať snímky bez anotácii

Class	# instances
Pedestrian	28602
Bicycle	5784
Motorcycle	7700
Passenger car	285891
Transporter	58834
Bus	15330
Truck	23401
Unknown	26974
Total	452516

Table 5.2: Numbers of class instances of all datasets combined

Chapter 6

Results

Chapter 7

Future Work

Chapter 8

Conclusion

Bibliography

[1] Bugdol, M., Miodonska, Z., Krecichwost, M. and KASPEREK, P. Vehicle detection system using magnetic sensors. *Transport Problems*. march 2014, vol. 9.

[2] Fang, J., Meng, H., Zhang, H. and Wang, X. A Low-cost Vehicle Detection and Classification System based on Unmodulated Continuous-wave Radar. In: *2007 IEEE Intelligent Transportation Systems Conference*. 2007, p. 715–720. DOI: 10.1109/ITSC.2007.4357739.

[3] Murali, S. and V K, G. Shadow Detection and Removal from a Single Image Using LAB Color Space. *Cybernetics and Information Technologies*. march 2013, vol. 13. DOI: 10.2478/cait-2013-0009.

[4] Stiawan, R., Kusumadjati, A., Aminah, N. S., Djamal, M. and Viridi, S. An Ultrasonic Sensor System for Vehicle Detection Application. *Journal of Physics: Conference Series*. IOP Publishing. apr 2019, vol. 1204, no. 1, p. 012017. DOI: 10.1088/1742-6596/1204/1/012017. Available at: https://dx.doi.org/10.1088/1742-6596/1204/1/012017.

[5] Wu, K., Xu, T., Zhang, H. and Song, J. Overview of video-based vehicle detection technologies. *ICCSE 2011 - 6th International Conference on Computer Science and Education, Final Program and Proceedings*. august 2011. DOI: 10.1109/ICCSE.2011.6028764.