ALEXANDRU IOAN CUZA
UNIVERSITY of IAŞI

FACULTY OF
COMPUTER SCIENCE

Distributed Multi-Camera Car Tracking using Deep Learning

by

Marian-Sergiu Nistor
Andrei Ghiran

Scientific coordinators:
Prof. Dr. Adrian Iftene
Prof. Dr. Anca Ignat

University Al. I. Cuza Iasi
Faculty of Computer Science

Abstract

This thesis covers a Proof-of-Concept implementation of a distributed multi-camera car tracking system using deep learning techniques, including depth estimation [13] and object detection [2]. Based on the output of the neural networks, the application reconstructs the 3-dimensional vector space, mapping the geographical positions of the detected cars and obstacles.

The application is meant to be used as a complementary unit to self-driving road vehicles, ensuring that the vehicles have an accurate overview of their surrounding obstacles and other traffic participants that are located in their vicinity.

The motivation for developing such system originates from the issues with the current state of self-driving vehicles. Given the present condition of technology, automated driving systems experience difficulties in providing appropriate safety conditions [19], such as considering the concept of object permanence [8] when making decisions.

Furthermore, considering the social aspect, the general public is susceptible to transitioning towards artificially intelligent systems. Such autonomous vehicles would not be subject to the same standards as human drivers, but would instead have to provide much higher safety conditions [11].

The proposed application augments the vehicle-centered approach, by providing an external view on the vector space, ensuring that the automobiles make well informed decisions, eliminating the possibility of the car's camera view being obstructed and missing out elements from its surroundings.

Keywords: distributed systems, multi-camera car tracking, artificial neural networks, deep learning, depth estimation, object detection, 3-dimensional vector space reconstruction, self-driving systems enhancement, artificial intelligence, road safety

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Given the current state of technology, there are multiple issues that occur when operating an autonomous driving module. Such systems make use of feedback-based control mechanisms (e.g. speed controller, direction controller), which trigger actuators (e.g. steering wheel, breaks, accelerator), relying on the input received from the vehicle sensors (e.g. camera, infra-red distance sensors, accelerometer, gyroscope). The most popular controller model is the PID controller (proportional–integral–derivative controller) [15] [3].

In order for the vehicle to make the appropriate decision based on the environment mapping, the control system must comply to a set of standards with regard to several metrics (e.g. latency, frequency of response and jitter), to ensure the safety of the automobile itself and the other traffic participants [19].

Even though the internal processor of the self-driving module can be designed in a way that fits such standards, this condition is not sufficient, since the controller inputs data from sensors and outputs data to actuators. Therefore, the overall performance of the system is dictated by its weakest performing element. Considering this, optimizing for the latency and the frequency of the underlying processor could potentially compromise jitter, and vice-versa.

With that in mind, taking some load off the car sensing mechanisms using an external actor could imaginably result in a better general performance of the self-driving vehicle.

Furthermore, current autonomous driving systems cannot fully comprehend the concept of object permanence [8]. An external camera positioned above of the considered space would be able to map the 3-dimensional vector space more accurately, without needing to implement additional instruments for taking object permanence into account.

Even if such problems would be tackled successfully in the future, there is a tendency of the general public to not adhere to unfamiliar artificial intelligence approaches. Therefore, these vehicles are subject to much higher standards of accuracy and safety [11]. Thus, a complementary spatial interpretation system would then provide an additional validation metric for autonomous driving modules, therefore enhancing road safety conditions.

1.2 State of the Art

Given the current state of technology, the autonomous driving sensing systems work operate predominantly on client level. The components that map the vector space use data received from sensors placed on the vehicle itself.

As of 2021, the most popular self-driving system is the Tesla Autopilot, even though there are other implementations that permit higher levels of vehicle autonomy, as per SAE's J3016 standard [16]. Tesla's autonomous driving module is currently classified as Level 2 (Partial Driving Automation), meaning that the vehicle is able to perform steering and acceleration, but requires the driver to monitor the tasks, having the option to take control at any given point in time.

During the vector space mapping stage, the Tesla Autopilot performs a series of tasks [7]:

- Object recognition
- Vehicle tracking
- 3D mapping for the surrounding objects
- Lane detection
- Curve detection
- Sign detection
- Path planning, using GPS information.

In order for the autonomous driving module to achieve these tasks, Tesla models have a series of sensors mounted [7]:

- Radar, for detecting moving objects
- Video camera for detecting all surrounding objects (e.g. cars, pedestrians). The camera chip is custom made, in order to optimize for feature extraction. While normal industry cameras use automatic post-processing for aesthetic refinement of the frames, deep neural networks achieve potentially better results using raw video data, which is richer in information
- Ultrasonic distance sensors, for tracking surrounding vehicles, using particle filtering [5], Bayesian filtering [4] and Monte Carlo sampling techniques [17].

1.3 Proposed Solution

The proposed solution shifts some of the environment sensing and mapping responsibility to external actors, represented by external stationary video cameras, intended to be positioned above roadways, therefore being able to detect objects from a larger area.

Furthermore, this approach reduces the possibility of omitting entities when during the detection and vector space mapping phase. Since the system introduces an additional perspective of the environment, the autonomous vehicles do not have to rely as much on artificial object permanence perception implementations, as the presence of a complementary detection actor increases the likelihood of observing surrounding elements which, under normal circumstances might not have been detected using a typical, fully client-oriented method.

The specified system makes use of deep learning techniques in order to perform depth estimation [13] and object detection [2]. Using the outputs of the aforementioned artificial neural networks, the application reconstructs the 3-dimensional space, providing the geographic coordinates of each entity's bounding boxes. The system uses the Map-Reduce paradigm [18], in order to efficiently distribute the computing load across multiple data processing nodes, orchestrated using a central dispatcher. The extracted vector space data is stored in a relational database, in order for it to be provided to autonomous driving vehicles, through the vehicles request fulfillment authority.

Chapter 2

Implementation

2.1 System Architecture

The system is comprised of two different authorities which operate asynchronously:

- The data processing authority (subsection 2.1.1)
- The vehicles request fulfillment authority (subsection 2.1.2).

2.1.1 The Data Processing Authority

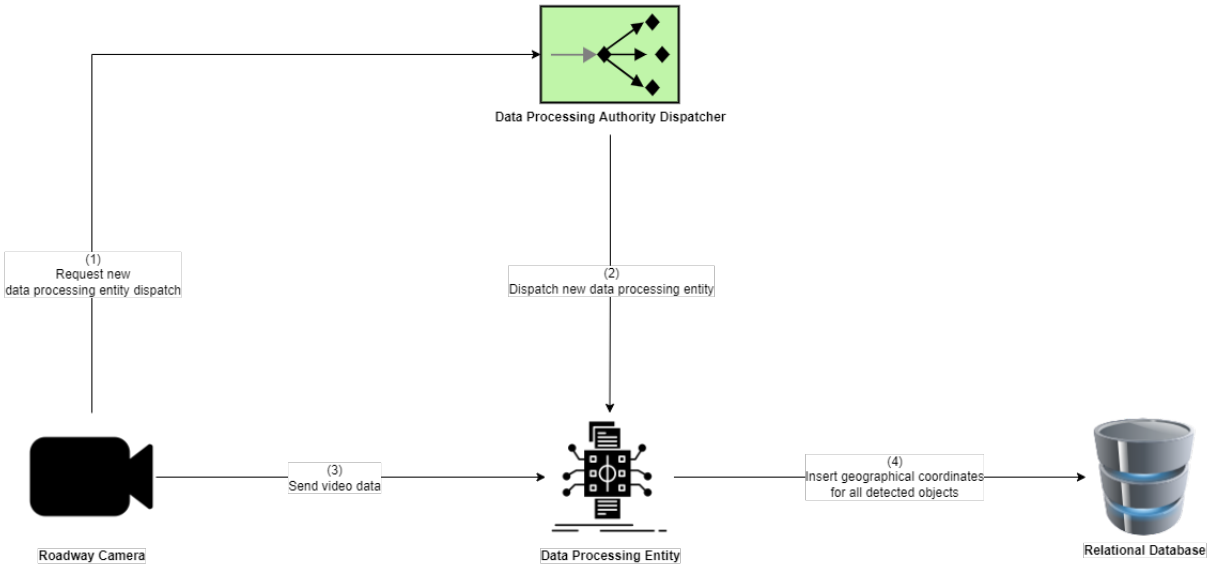

Figure 2.1: Video data processing and vector space reconstruction phase.

The data processing authority (Figure 2.1) is represented by a REST API that dispatches UDP servers per each request, redirecting each client to its designated data processing actor. Each data processing actor receives raw video data from cameras situated above the roadway, then pushes the data through a neural network pipeline, constituted of a pretrained depth estimation network [13] and an object detection network [2]. Then, the output data is joined in order to perform the vector space reconstruction, inserting geographical coordinates for each detected object's bounding boxes, in a relational database.

An important optimization aspect is that the data received from the cameras is being processed in parallel, distributing the load between the data processing actor instances. This data processing pattern is inspired from the MapReduce [18] approach, granting high scalability for the system.

2.1.2 The Vehicle Request Fulfillment Authority

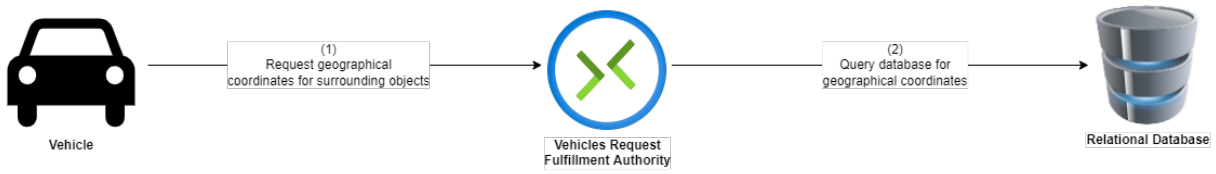

Figure 2.2: Vehicle requests fulfillment phase (providing vehicles with geographical data of their surrounding cars and obstacles).

The vehicles request fulfillment authority (Figure 2.2) is represented by a REST API, which, upon each vehicle's request, queries the data from the relational database and provides geographical data to the specified car, regarding objects situated in its vicinity (other vehicles, or roadway obstacles).

2.2 Data Processing Pipeline

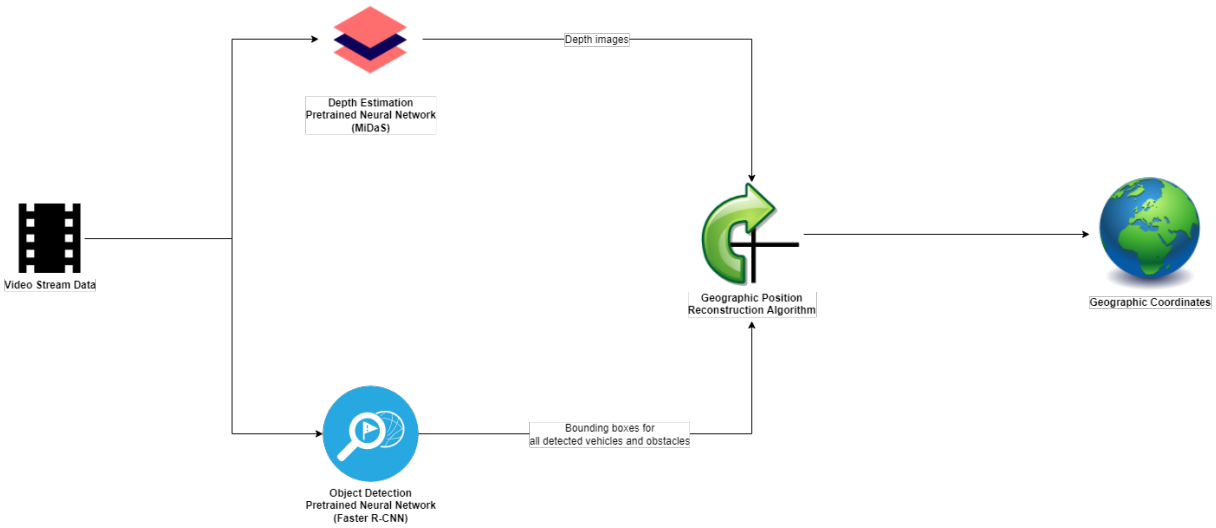

Figure 2.3: The data processing pipeline.

The data processing pipeline is executed by the Data Processing Authority (subsection 2.1.1), when receiving UDP video data from its associated roadway camera client. The data is used in order to extract geographical coordinates for all objects detected by the camera, either vehicles or obstacles. The data processing pipeline is comprised of three sub-modules:

- The depth estimation component (subsection 2.2.1)
- The object detection component (subsection 2.2.2)
- The geographic position reconstruction component (subsection 2.2.3).

2.2.1 The Depth Estimation Component

Figure 2.4: Depth estimation using MiDaS [13].

The depth estimation component is represented by a pretrained neural network, its model being called MiDaS [13], a monocular depth estimation neural network, having a ResNet-based architecture [9]. ResNet's make use of residual blocks (Figure 2.5) which, in addition to traditional convolutional blocks, introduce skip-layers. In order for the model to identify image features on a deeper level, the neural network architecture must also be deeper. The issue that occurs in this case is that neural networks tend to perform gradually worse after a certain depth. Residual blocks fix this issue, providing an alternate path for the gradient to flow through. An example of a depth frame generated using MiDaS can be seen in Figure 2.4.

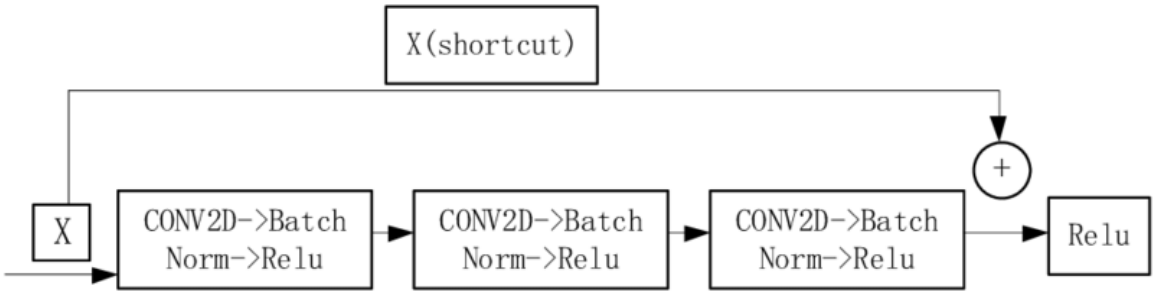

Figure 2.5: Residual block structure [10].

2.2.2 The Object Detection Component

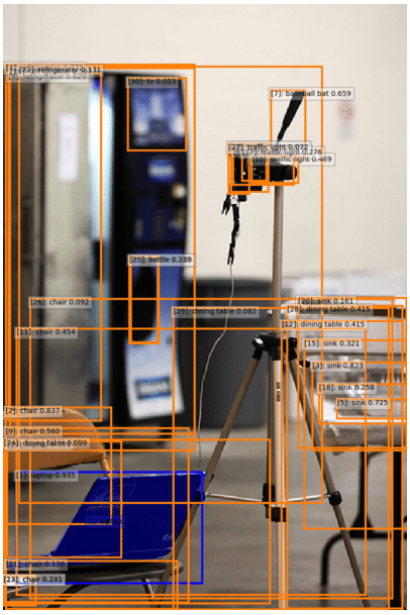

Figure 2.6: Bounding boxes generated using Faster R-CNN [6].

For the object detection component, the system makes use of a pretrained faster residual convolutional neural network (Faster R-CNN [2]), which combines ResNet's residual block-based architecture (Figure 2.5) with a RPN (region proposal network [14] [20]), having a convolutional neural network backbone, which generates proposals for regions that might contain objects that will subsequently be classified. The RPN is also composed of two predictors: a multi-class classifier that performs the classification task on that particular filtered region, and a regressor for generating the coordinates of the proposed region's bounding boxes (Figure 2.7). An example depicting a Faster R-CNN's output can be seen in Figure 2.6.

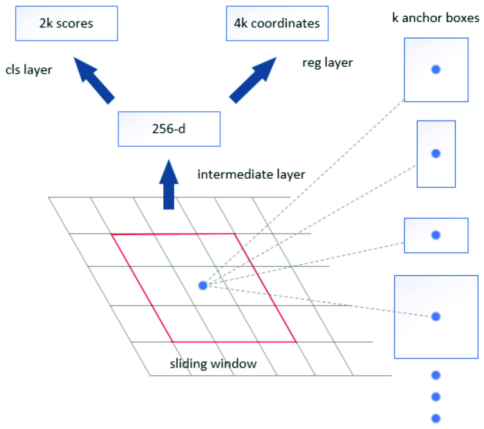

Figure 2.7: Region proposal network (RPN) iterating over the set of filtered regions [20].

2.2.3 The Geographic Position Reconstruction Component

The geographic position reconstruction component constructs the vector space containing the geographic coordinates (latitude, longitude) for all vehicles and obstacles detected in a given roadway camera video frame.

In order to achieve this particular goal, the algorithm executes the following steps:

- Iterate over all bounding boxes for the detected objects
- Given the depth image, for each bounding box vertex, compute its 3-dimensional position, relative to the roadway camera's position and rotation
- Normalize the bounding box vertex coordinates, by transposing from Cartesian coordinates to spherical coordinates, rotating over negative theta and negative phi, and transposing back to Cartesian coordinates.

Chapter 3

Results

3.1 Evaluation Metrics

We define a "positive" sample to be that which detects a specific object in a region of the frame. Similarly, we define a "negative" sample to be that which does not detect a specific object in a region of the frame.

We've sampled a number of 100 frames in order to determine the accuracy of the object detection component over a series of evaluation metrics. The model is expected to achieve lower accuracy than normal due to the fact that we downsize the video frames in order to fit them in one UDP packet, then resizing them to a standard size, therefore optimizing for speed. In a real application setting, we suggest the usage of a specialized video streaming algorithm for sending data from roadway cameras, so as to achieve better object detection accuracy.

During the evaluation phase, the model recorded 169 true positives, 33 false positives, 67 true negatives, and 33 false negatives.

The evaluation metrics considered in order to determine the system performance are the following:

- True positive rate (recall): $\frac{TP}{TP+FN} = 83.66\%$
- False positive rate: $\frac{FP}{FP+TN} = 33\%$
- Precision: $\frac{TP}{TP+FP} = 83.66\%$
- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN} = 78.15\%$
- F-measure: $\frac{2 \cdot Precision \cdot Recall}{Precision+Recall} = 83.66\%$

Additionally, the system is able to process $\sim$128 KB of video data per second, and satisfy $\sim$50 vehicle requests per second, considering that the servers are hosted on machines having 16 GB RAM, 16 logical cores, 2 GHz processors, and the internet connection speed between all the vehicles, cameras and servers have a data transmission rate of 1 GBps.

Chapter 4

Conclusions

4.1 Summary

To conclude, the proposed system can potentially have a positive impact on autonomous driving vehicles' safety considerations, if used appropriately by automobile manufacturers. The application does not aim to replace any of the required sensing capabilities of the automobiles, but instead to complement them, providing means of validating information regarding the positions of surrounding cars and obstacles.

The enhancement brought to safety considerations is provided by the fact that an external roadway camera can provide a better spatial mapping, due to its higher position, relative to the traffic participants. Therefore, the issues with faulty object permanence implementations on vehicle level could result in a lower error rate, while the roadway camera's shouldn't be subject to obstructions as often as an automobile camera would.

Even though alternative implementations for the given task can be achieved with less effort, using specialized depth cameras and stereo camera positioning, the project acts as a Proof-of-Concept for a monocular approach, generating depth images using deep learning models. Hence, the conducted research aims to provide a method for performing the aforementioned task in a cost efficient manner, so as to facilitate a large scale implementation of the proposed system, by optimizing production cost.

4.2 Future Work

There are a series of aspects that should be considered when designing future implementations for this specific task. The first feature to be considered when optimizing should be object detection accuracy. The proposed implementation makes use of Faster R-CNN with a ResNet-50 backbone. Deeper ResNet-based architectures could conceivably achieve better accuracy, the trade-off being that of execution speed.

Another thing to consider is replacing bounding boxes-based object detection, with semantic segmentation models, in order to fit the detected object's shape more accurately. Semantic segmentation can also be achieved using convolutional neural networks models (e.g. Fully Convolutional Network [12]). One undesired implication consists of the fact that storing semantic segmentation data would require more storage space.

When considering the mechanisms for transmitting video data between the roadway cameras and the data processing authority instances, the aspect that affects the object detection accuracy the most is the fact that each frame is being compressed in a single UDP packet. This implies the fact that the deep learning algorithms perform predictions based on low quality video data. Thus, implementing a method that transmits each frame's data over multiple UDP packets, would allow for higher accuracy, while the video frame's resolution would be higher. This variant would also end up affecting the execution speed. In order to combat this issue, a different compression type could be used, that minimizes data transmission over the network. Right now, each video frame is intra-coded (I-frame compression). P-frame or B-frame compression would allow for quicker video frame transfer, since these methods do not require sending data for regions that did not suffer any updates since the last update [1].

References

[1] D. Raveena Judie Dolly, G. Josemin Bala, and J.Dinesh Peter. Adaptation of frames for gop using nsew affine translation for video compression. In *2014 International Conference on Electronics and Communication Systems (ICECS)*, pages 1–6, 2014.

[2] Quanfu Fan, Lisa Brown, and John Smith. A closer look at faster r-cnn for vehicle detection. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 124–129, 2016.

[3] Wael Farag. Track maneuvering using pid control for self-driving cars. *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, 13(1):91–100, 2020.

[4] V Fox, Jeffrey Hightower, Lin Liao, Dirk Schulz, and Gaetano Borriello. Bayesian filtering for location estimation. *IEEE pervasive computing*, 2(3):24–33, 2003.

[5] Fredrik Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82, 2010.

[6] David Hall, Feras Dayoub, John Skinner, Haoyang Zhang, Dimity Miller, Peter Corke, Gustavo Carneiro, Anelia Angelova, and Niko Sünderhauf. Probabilistic object detection: Definition and evaluation. 11 2018.

[7] Shantanu Ingle and Madhuri Phute. Tesla autopilot: semi autonomous driving, an uptick for future autonomy. *International Research Journal of Engineering and Technology*, 3(9):369–372, 2016.

[8] Claus Lang, Guido Schillaci, and Verena V Hafner. A deep convolutional neural network model for sense of agency and object permanence in robots. In *2018 Joint IEEE 8th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 257–262. IEEE, 2018.

[9] Sihan Li, Jiantao Jiao, Yanjun Han, and Tsachy Weissman. Demystifying resnet. *arXiv preprint arXiv:1611.01186*, 2016.

[10] Zhiming Li. Practice of gesture recognition based on resnet50. *Journal of Physics: Conference Series*, 1574:012154, 06 2020.

[11] Peng Liu, Run Yang, and Zhigang Xu. How safe is safe enough for self-driving vehicles? *Risk analysis*, 39(2):315–325, 2019.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.

[13] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, 2020.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[15] Daniel E Rivera, Manfred Morari, and Sigurd Skogestad. Internal model control: Pid controller design. *Industrial & engineering chemistry process design and development*, 25(1):252–265, 1986.

[16] Sergej S Shadrin and Anastasiia A Ivanova. Analytical review of standard sae j3016 taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles with latest updates. *Avtomobil'. Doroga. Infrastruktura.*, (3 (21)):10, 2019.

[17] Alexander Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

[18] Kyuseok Shim. Mapreduce algorithms for big data analysis. *Proceedings of the VLDB Endowment*, 5(12):2016–2017, 2012.

[19] Michael Sivak and Brandon Schoettle. Road safety with self-driving vehicles: General limitations and road sharing with conventional vehicles. Technical report, University of Michigan, Ann Arbor, Transportation Research Institute, 2015.

[20] Denan Xia, Peng Chen, Bing Wang, Jun Zhang, and Chengjun Xie. Insect detection and classification based on an improved convolutional neural network. *Sensors*, 18:4169, 11 2018.