Machine Learning for Automated Driving

© Bertrandt

Automated and autonomous driving are among the core topics of future mobility. In its innovation project "Park and Charge", Bertrandt shows how important environment recognition and precise trajectory planning using artificial intelligence are. Machine learning is used to improve localization, networking and cloud applications.

CURRENT STATUS OF OBJECT DETECTION

Autonomous driving functions are already available in some production models. They allow the car to fulfill the driver's role in a range of situations, but they must be able to take reliable decisions very quickly. As a result, the use of Artificial Intelligence (AI) and, in particular, of Machine Learning (ML) is essential. The decisions are made by pre-trained deep learning algorithms and the input consists of measurements from a variety of sensors, including camera, radar, and lidar systems. The main function of the algorithms is to reconstruct the car's environment using object recognition and to make this information available to the car's systems.

PRINCIPLES OF MACHINE LEARNING AND NEURAL NETWORKS

The concept of AI can be interpreted in a number of different ways. This field of information technology relates to the independent, automatic resolution of problems [1] and a distinction is made here between strong and weak AI. While strong AI involves imitating people and their behavior or thinking (for example, consciousness), weak AI is limited to training algorithms to resolve a clearly defined problem [2].

The most widely known area of AI is ML, which falls into the category of weak AI. Many leading experts in the field of AI believe that ML is one of the most important technologies of the modern age [3]. ML can be divided into three different areas:

- Supervised learning: The algorithm is shown the data and the expected

AUTHORS

Dipl.-Ing. Peter Schiekofer
is Director of Advanced Engineering and Autonomous Driving of the Bertrandt Group in Ehningen (Germany).

Dr. Yusuf Erdogan
is Head of Data Science and Development at Bertrandt in Frankfurt/Main (Germany).

Dipl.-Phys. Stefan Schindler
is Data Scientist in the Machine Learning Department at Bertrandt in Frankfurt/Main (Germany).

Markus Wendl, M. Eng.
is Lead Engineer in the ADAS Department and Subproject Leader Software Development for Autonomous Driving Functions in the “Park and Charge” Project at Bertrandt in Cologne (Germany).

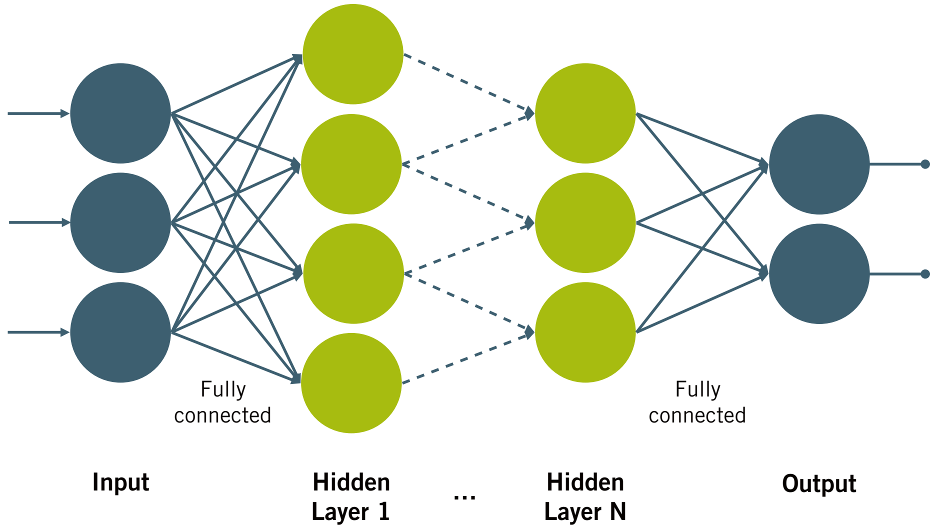

FIGURE 1 Illustration of a neural network with different layers (© Bertrandt)

result and learns to link these two variables together. Once the learning process has been completed, the algorithm can independently find a result for unfamiliar data.
- Unsupervised learning: The algorithm learns to identify and classify patterns solely on the basis of the data.
- Reinforcement learning: The algorithm learns the ideal strategy for solving a problem by being offered rewards and punishments.

Supervised learning is the most widely used of these three methods. In the world of open source tools, there are many different environments available with a variety of architectures. One specific group of architectures consists of neural networks, which have a structure that mimics the neurons in the human brain. These neurons are arranged in layers that fulfill different functions. Each layer consists of a very large number of individual neurons, although this number varies from one architecture to another. Every neuron in the layer is connected to the neurons in the subsequent layer. This connection forms the basis of the architecture. If every neuron in one layer is connected to all the neurons in the next layer, this is described as a Fully Connected Layer (FCL). As only the first and the last layer are generally used for inputs and outputs, the interim layers are referred to as hidden layers, **FIGURE 1**.

Convolutional Neural Networks (CNNs) are used for object recognition in images. The image is supplied to the input layer and broken down into pixel data. The convolution process gradually reduces the dimensions of the information and, at the same time, leads to features being learned which are stored in the neurons in the network. Depending on the architecture, the image passes through hundreds of these layers in the neural network, for example in a Faster-R-CNN [4], which results in this process being described as deep learning. In the output layer, the network is shown the objects in the image and their location. This information is compared with the network’s own result and the anomalies are propagated back into the network to reset the neurons accordingly. Once the network has been shown sufficient objects in the images and the anomalies have been reduced, the network learns to localize and detect these objects independently.

IMPLEMENTING TRAJECTORY PLANNING FOR AUTONOMOUS DRIVING SYSTEMS

A trajectory planning algorithm was developed for the “Park and Charge” project which calculates a valid path to any point in the direction of the destination. During this process, obstacles from the local environment map, which is created by lidar and ultrasound sensors, are taken into consideration and avoided. In addition, highly accurately measured routes can be stored. These are subsequently used by the algorithm to calculate the trajectory to the destination.

The algorithm is based on a modified, nested A* algorithm with a mathematical comparison that significantly reduces the computing time and defines the final direction of the calculated path.

Data specific to the car are needed to calculate a drivable path. These must define the car’s limits in terms of driving dynamics, among other things. The calculation of the possible paths is not restricted by the algorithm, but by the agility of the car, for example. Therefore, the option of calculating an agile target trajectory for a forklift is also available.

THE DRIVING SCENARIO FOR "PARK AND CHARGE"

In the driving scenario that is worked toward, the car is sent a recommended route to the parking space via the in-house Bertrandt cloud. With the help of the algorithm developed, the recommended route can be improved in real time using environment data and then followed to the destination, **FIGURE 2**.

As soon as the car has reached the destination, the specified parking space is validated using the environment sensors and the parking process is calculated in one to three moves. The type and direction of the parking space are irrelevant in this respect. An example of a parking process is shown in **FIGURE 2** using simulated data. The specified route on the map is recessed and the walls and vehicles are raised.

This indicates the areas where the vehicle can move independently without external specifications. In addition, it is possible to define zones where the vehicle cannot go. These include, for example, the oncoming lane, pedestrian crossings and sidewalks.

Currently, a methodology is developed for predicting possible collisions by continuously comparing the data from the environment sensors and the local map. This also involves a comparison between the extrapolated movement of the car and the dynamic and static environment data.

TRAINING IN OBJECT DETECTION

The approach taken towards camera-based object detection allows the work involved in generating training images to be reduced considerably [5]. This method can be used to detect objects with an appearance that does not vary significantly. Suitable objects include road signs and traffic signals, while cars, animals, pedestrians and cyclists are unsuitable in this case.

The objects are initially photographed from the various perspectives that they can be detected from in the use case and against a homogeneous background. Then the objects are automatically cut out from a representative quantity of the photographs via image processing algorithms. The objects that have been removed are then inserted into different background images of the target environment (for example street scenes).

The experts at Bertrandt have experimented with different blending processes to improve the results.

In order to simulate different environmental influences and generally to avoid overfitting during the training process, the appearance of the objects and backgrounds is additionally varied using data augmentation. **FIGURE 3** shows the objects that the car must detect while it is moving. These objects can be used, for example, as landmark reference points to improve the positioning of the car. If the IoT traffic signal is switched to green by the back-end system, the car must be able to detect that the light has changed and continue its journey as soon as it has environment clearance.

In **FIGURE 3** (bottom), it is clear that the positions of the inserted objects and their relationship to the objects in the background are not realistic. Due to the structure of convolutional neural networks, only local features are relevant for training or for achieving a high number of hits, and not global image relationships such as the position in the overall image.

The neural network is initially trained using artificially generated data only. Subsequently, it is validated and fine-tuned using a small quantity of real data from images taken by own test cars.

SYNTHETICALLY GENERATED DATA SAVE TIME AND MONEY

Therefore, a method for creating synthetically generated training data was applied that enables the time-consuming and expensive insertion of real data and subsequent annotation to be kept to an absolute minimum. The approach described here allows annotated training images of new road signs to be generated quickly and cost-effectively.

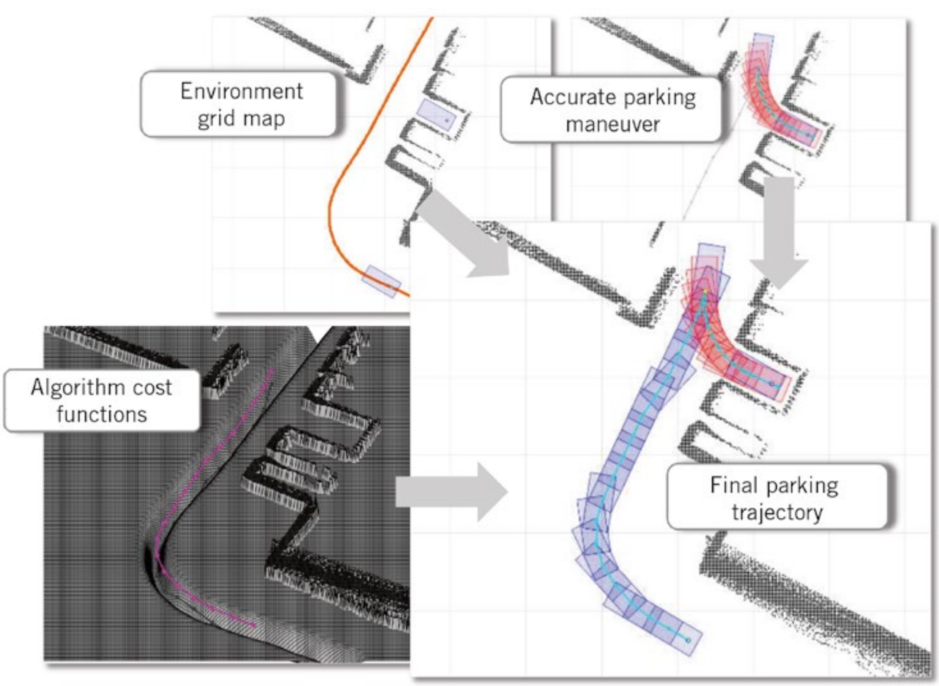

FIGURE 2 Trajectory planning algorithm (© Bertrandt)

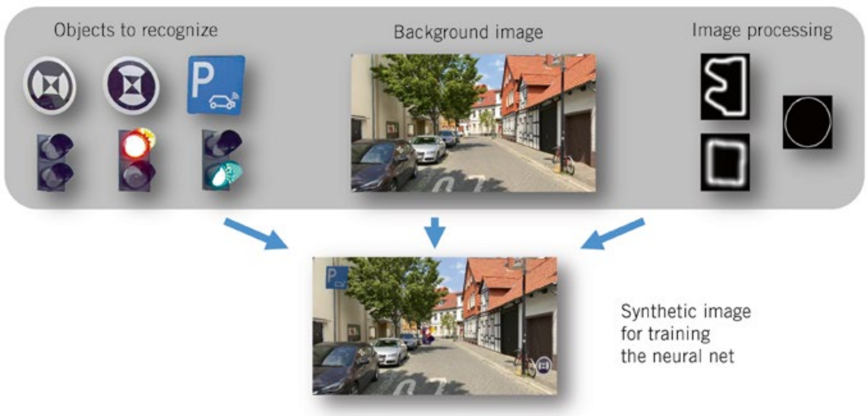

FIGURE 3 Graphical representation of the approach for creating artificial training images (© Bertrandt)

After the neural network has been trained, the inference time (runtime) is optimized by means of a low quantization of the weights (from 32-bit float to 8-bit integer) for the target hardware (Nvidia Drive PX2 Autochauffeur) and compliance with the necessary runtime criteria.

AUTOMATIC GENERATION OF GROUND TRUTH DATA

Bertrandt is currently in the process of developing a product that can be used in future to generate ground truth data from sensor data automatically. The software is based on computer vision algorithms and pre-trained neural networks. These neural networks can detect different objects on camera frames independently and annotate them with a high level of accuracy. The computer vision algorithms are used to track the annotated objects over a sequence of several frames. Together, the two algorithms can either help the user with the manual annotation of camera data or fully automate the entire annotation process. The Bertrandt labeling tool will be made available this year to a number of customers in the form of Software as a Service (SaaS). The software will also be used to generate the training data for the neural networks described in this article. In subsequent phases of the development process, functions for object detection from lidar data will be added. The experts from Bertrandt have acquired the expertise needed to develop the product and, in particular, the intelligent algorithms, during the mentioned "Park and Charge" innovation project.

OBJECT DETECTION USING CAMERA AND LIDAR DATA

As the project progresses, ML will be used to develop camera- and lidar-based object detection which will then be optimized for the target hardware. For this purpose, the latest approaches from research are applied, in which a neural network learns not only the features of the objects to be detected in camera and lidar data, but also the fusion of these data. Camera- and lidar-based object detection can combine the benefits of each type of sensor and compensate for their disadvantages. This project is being handled by an interdisciplinary team from different Bertrandt sites.

REFERENCES

[1] Michalski, R. S.; Carbonell, J. G.; Mitchell, T. M.: Machine Learning. Berlin, Heidelberg: Springer-Verlag, 1983

[2] Goertzel, B.: Toward a Formal Characterization of Real-World General Intelligence. Proceedings, 3rd Conference on Artificial General Intelligence, Lugano, 2010, No. 10.2991/agi.2010.17

[3] Brynjolfsson E.; McAfee, A.: The business of artifcial intelligence. Havard Business Review, 2017

[4] Ren, S.; He, K.; Girshick, R.; Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: IEEE Trans. Pattern Anal. Mach. Intell., Vol. 39, No. 6 (June 2017), pp. 1137-1149

[5] Dwibedi, D.; Misra, I.; Hebert, M.: Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In: arXiv: 1708.01642 (2017)