A matter of reality

case study: comparing the training of neural networks with real training data and virtual training data

Julia Hartung, B. Sc.
University of Applied Science
Esslingen
Esslingen, Germany

Nico Kuhn, B. Eng.
University of Applied Science
Esslingen
Esslingen, Germany

Patrick Quell, B. Eng.
University of Applied Science
Esslingen
Esslingen, Germany

Konstantin Wacker, B. Sc.
University of Applied Science
Esslingen
Esslingen, Germany

Prof. Dr.-Ing. Reiner Marchthaler
University of Applied Science
Esslingen
Esslingen, Germany

Abstract. Due to the increasing relevance of data, more and more data from various sources is accumulated for a variety of purposes. At the same time, however, there is a shortage of data in areas where it is urgently needed. Particularly in the field of machine learning, there is a lack of good and usable training data. Therefore, this research paper is concerned with the virtual data acquisition for the training of neural networks. For this purpose, first an application was developed that aims to generate virtual, automatically labeled data. Subsequently, a neural network was trained on the generated virtual data and tested on real data.

***Keywords*—neural networks, data analysis, data mining, training data, virtual**

I. Introduction

IN the automotive industry, there are constant advancements in technology, with autonomous driving taking an increasingly important role. With the help of autonomous driving, both the number of accidents and congested traffic are to be reduced. In addition with all the advancements in driver assistance systems there is a change in values among drivers, in which comfort and safety is seen more important than the ability to manually operate a car. Sensors, algorithms and actuators play an important role in autonomous driving. Together with the mechanical advancements in the automotive development, there is also a digital development. With the help of artificial intelligence and adaptive neural networks, the vehicle is becoming smarter and smarter. [1]

In order to train the neural networks, a large amount of data is required, which must be collected and preprocessed in a complex manner. As a result, data generation in the automotive sector takes an increasingly important role. [2]

In the general IT trends of 2018, the technology area of data is also in second place across all industries. [3] On the one hand, a great deal of data is collected in various areas, however this data is not always usable due to data quality and data privacy protection. [4] On the other hand, a great deal of data is needed to advance technologies such as machine learning.

The sensible use of data offers potential for social and economic changes that only occur once or twice within a century. [4] Especially in the area of autonomous driving groundbreaking progress can be achieved through improved data processing, which can be expanded into a clear competitive advantage. Thus, Tesla concentrates largely on optimizing self-learning algorithms based on the input of large amounts of data. Daimler, BMW and VW are also focusing on the acquisition of training data, using large vehicle fleets for this purpose. [5]

Fabian Patterson supports this realization on neural networks stating that it is not only the algorithms but above all the data that lead to success. [6] Collecting and correctly processing data is very time-consuming and crucial for success, especially in the context of neural networks. For the pure classification of easily recognizable objects in images, a value of at least 1,000 sample images per class is required in order to achieve satisfactory results. This value is even higher for object detection, which is required for autonomous driving. [7] The generation of training and test data requires a lot of time and money.

The development in the machine learning area forms the basis for a new business model in the form of SaaS (Software as a Service), which tracks the generation of training data or contains pre-trained networks. [8] In this research work, a different, more flexible approach is chosen: the creation of computer-generated, virtual training data as a substitute for real data.

The research provides a brief overview of neural networks and their relation to training data. Then the generation of virtual data, which is used to train neural networks, is explained in more detail. A brief insight into comparable

research will provide a comprehensive overview before the results are validated. At the end follows an outlook, which offers a direction for future research work.

II. Basics

A. Neural networks

Artificial neural networks (KNNs) are part of machine learning. The KNNs are based on functions of the human brain. As in the biological field of application, an artificial neural network also consists of many neurons which are linked to each other. The connections between the individual neurons are weighted, whereby the influence of the respective neuron is controlled. Before a neural network can be applied to a specific scenario, it has to be trained on scenario-specific sample data. [14][19]

The most widespread method for training is supervised learning, in which the expected result for a given input is known. The outputs during the learning process can thus be compared with the expected results and evaluated. The individual weights of the neurons are then adjusted in the backpropagation step, trying to minimize the error determined during the evaluation of the previous step. [2][14][19]

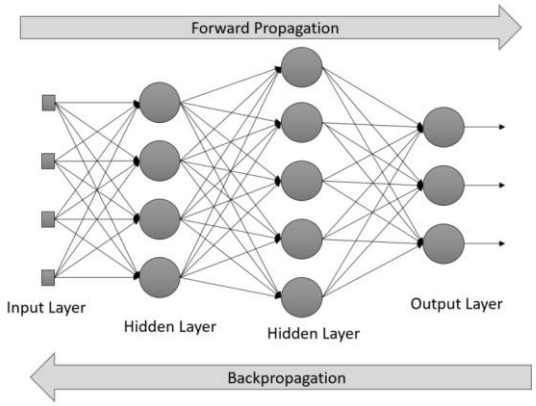

Figure 1 Artificial neural network

The output value of each neuron is calculated from the weighted input values and the activation function. The most commonly used activation functions are the linear, tanh, sigmoid and Gaussian functions. The activation function can be set individually for each neuron. [14]

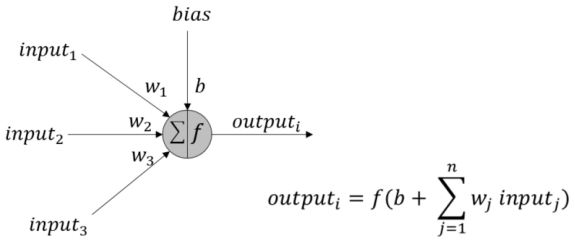

Figure 2 Simplified model of an artificial neuron

When creating a new neural network the individual weights of the neurons are initialized randomly thus making it necessary to adjust them over many training steps to get a robust neural network. [2]

B. Vision Tasks

A very common application of neural networks is object recognition in images. A distinction is made between different recognition levels. The simplest task is the classification of an image. Here images are assigned exactly one of several pre-defined classes. Object detection expands on this and entails the classification and localization of any number of objects in an image. The localization is thereby done through the use of bounding boxes. The localization process can be described in more detail during image segmentation by describing the exact pixel regions of the respective objects. [9]

In general, large amounts of training data are required for supervised learning. The more data available, the better the result. Particularly in the areas of localization and object detection, where, in addition to the classification of objects, their position in the image is important, a great deal of training data must be used. [2]

However, oftentimes sample data is hard to come by resulting in a severe lack of availability of suitable training data. One step towards increasing the amount of training data is their transformation of existing data. In case of image data this could be distortion or rotation of existing images. In addition to the amount of data, however, the quality and representativeness of the data is also decisive for satisfactory results. [10][15][16]

C. Over- and Underfitting

A widespread problem in the training of neural networks is over- or underfitting. With the phenomenon of overfitting, the network achieves good results on the training data, but new, previously unseen data is not reliably detected. Overfitting occurs when a neural network is trained with a data set which is not suitable for the problem. This can oftentimes be traced back to a low number or little variability in the training data. Too much training on the same data set can also result in overfitting. With an increasing number of training steps, the recognition performance increases. However, as can be seen in the following figure, a saturation phase is reached at a certain point. From this phase on, the recognition performance on previously unseen data decreases significantly, as the model adapts too much to the specific characteristics of the training data. [17]

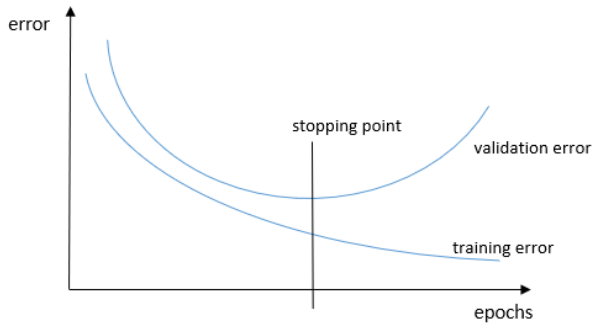

Figure 3 Typical error graph for a neural network

Underfitting describes the training of a network with too little data, whereby not all weights can be optimized sufficiently resulting in a low prediction accuracy. [17]

D. Training data

Before the training data can be used, most cases it must be pre-processed and scaled in advance. For supervised learning, it is necessary to label the training data correctly. For image classification, it is sufficient to label the images with a corresponding class. The labeling process for localization and object detection however requires type and the exact position of the contained objects of interest within an image. The coordinates of the objects are usually indicated by the placement of rectangular bounding boxes over the relevant areas in the image. For the purpose of labeling image data a variety of tools exist. Examples include the open source software "LabelImg" or "VoTT". [11][12] These tools generate an Extensible Markup Language (XML) file, which has a standardized format and can be used for further training. [11] For a sufficient amount of training data, however, the time required for labeling is enormous. Gathering and processing the data takes up the majority of the time in ANN development and is still oftentimes insufficient for a satisfactory result. In many cases employees have to be hired specifically for the labeling of data or labeled data has to be purchased externally. For smaller companies or research groups in particular, this considerable cost factor associated with the acquisition of training data is oftentimes unreasonable. [8][13][16]

An alternative approach is the creation of virtual training data. Thus the creation of large amounts of sample data can be automated and created very quickly. Also higher flexibility is achieved making it easy to adapt to changes in the required sample data. For image data the environment can be easily adapted and data for new object types can be quickly created by extending a virtual scenario. The real added value, however, is that the data can be labeled automatically. This considerably reduces the time and effort required for the acquisition of training data.

E. Virtual Engine

For the purpose of creating virtual training data, in the scope of the research the Unreal Engine 4 is used. The Unreal Engine is a game engine developed by Epic Games. It was published in 1998 and has since been used in the development of various games on various platforms. [18]

In this research the Unreal Engine 4 is used to create virtual scenarios, from which training data is generated. The virtual scenarios are built to be optically strongly reminiscent of the real training images. The virtual images are captured using a virtual camera placed in the scenario. In the scenario different types of traffic signs are placed and backgrounds and environments are switched randomly in between captures taken by the camera. The captured images stored and a corresponding XML file containing all relevant label information is created. Object types and their positions in the captured image are thereby retrieved and calculated from the known positions of all objects within the game engines scene and their relation to the virtual camera. By determining the relative position of the objects to the virtual camera, the corner points of the objects in the generated image are calculated and a bounding box is placed around the object.

This shows the great potential and flexibility of using virtual training data. The automated calculation of bounding boxes speeds up the labeling process dramatically and reduces the required effort to the initial setup of the scene. Once a virtual scenario suitable for the required image data is set up, training data can be created very quickly and adjustments or extensions can be realised very easily.

III. Other Approaches

Large companies such as Google, Facebook or automotive companies are much better positioned to collect training data than smaller companies or start-ups. They use large fleets of vehicles to collect data for their research and development in areas like autonomous driving. This puts organizations with less resources at a severe disadvantage due to the fact that the amount of training data is a key factor in the improvement of machine learning algorithms.

For such companies, synthetic training data represents an opportunity to catch up to well-established companies with far more resources. It was out of this motivation that employees at the Xerox Research Center Europe began developing virtual environments to help train neural networks. For their project, Xerox used the games engine Unity to generate virtual scenes and environments. Due to the extensive asset store of the Unity engine, many common objects already exist and do not have to be modeled from scratch. Various camera settings and exposures made it possible to extract a wide variety of images. Xerox was thus able to generate variant-rich data, which were of similar quality as real data. For evaluation purposes, Xerox converted a real scene into a virtual scene using a laser scanner. [20]

Game engines therefore offer a promising approach to training AIs. The open-world game Grand Theft Auto 5 has an extremely realistic virtual environment with very realistic road traffic. In the game, many different scenarios can be represented, such as variations in exposure or different

weather scenarios. Situations that are difficult to simulate in reality, such as car accidents, are also easy to depict in video games.

A group from TU Darmstadt worked together with employees from Intel Labs on the generation of training data from the game GTA5. However, although the video game offers very realistic environments, as a commercial product there are no official APIs or access to the code provided. To circumvent this the researchers implemented a software layer between the game and the graphics hardware that allows the communication between these two components to be monitored and modified. The software identifies the used mesh, texture and shader for each pixel of an extracted image. From the combination of these properties, conclusions can be drawn about the existing objects. The software also recognizes relationships between these resources and labels that have already been found and establishes rules that are used for labeling in images viewed later. However, the algorithm cannot automatically label every pixel, meaning some images still have to be processed manually. [21]

The project group extracted about 25,000 pictures from the game GTA5. The labeling process was completed after 49 hours. [22] Thereby 98.3% of the pixels could be assigned to corresponding classes.

As the project of the TU Darmstadt shows, there is no possibility to get the code for large, extensive games like Grand Theft Auto. Furthermore, there are often licensing problems with commercial titles. For these reasons, a research group at John Hopkins University of Baltimore developed the UnrealCV tool, which extends the Unreal game engine with functions for interacting with a virtual environment created with the engine. It also enables communication between the engine and external programs. The tool consists of two components - the client and the server. The client can be used to send commands to the server to read information from the game environment, such as the positioning of an object. The server processes the requests, generates the requested data and sends the result back to the client. With the editor plugin from UnrealCV, the virtual environment can also be edited and adapted. With these functions, large data sets can be created directly from the engine. [23]

IV. Experiments

The purpose of this paper is to evaluate whether virtual training data can replace real training data when training neural networks. As a basis for comparison real training data is collected by taking images with an onboard camera during a ride with an RC car through a test track with five different types of traffic signs. To generate the virtual training data the game engine Unreal Engine is used. In the engine a scenario is set up to automatically generate random images of traffic signs in varying environments. The following traffic signs were used: pedestrian crosswalk, 30-km-per-hour zone, end of 30-km-per-hour zone, give way sign, parking sign.

On these virtual data, several neural networks based on the Yolo V2 architecture are trained [24]. By continually adapting the virtual scenario it is attempted to improve the results during the validation of the neural network on real image data.

To acquire real image data for the validation, initially onboard videos of the RC car driving through the test track are taken and split into individual frames. Out of these frames 300 images for each type of traffic sign are randomly selected and labeled by hand. Then a Yolo V2 network is trained on the data set as a base line. The network is validated with 100 test images of real data. The results show that the predicted labels and bounding boxes match the true label and bounding boxes closely. The intersection error in each image is plotted in a diagram. The diagram shows how big the intersection error of the bounding boxes is for each class. These first results should serve as a baseline to evaluate the subsequent results after training with virtual training data. It is expected that the virtual training data can come close to the results of the real training data.

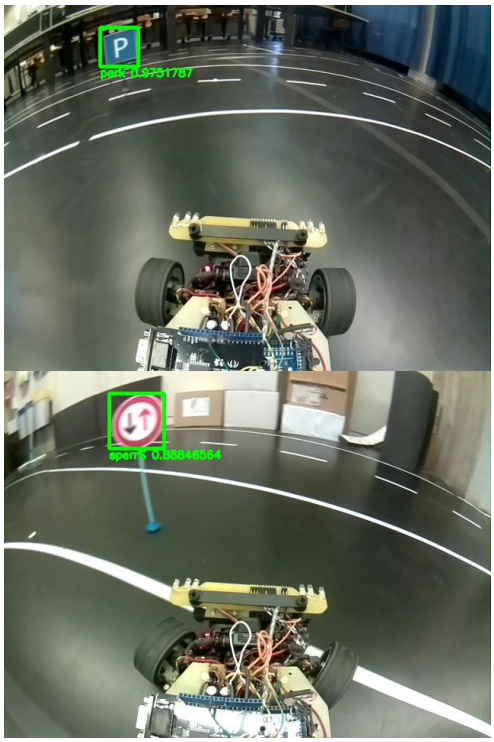

Figure 4: Output of neural network trained on real data

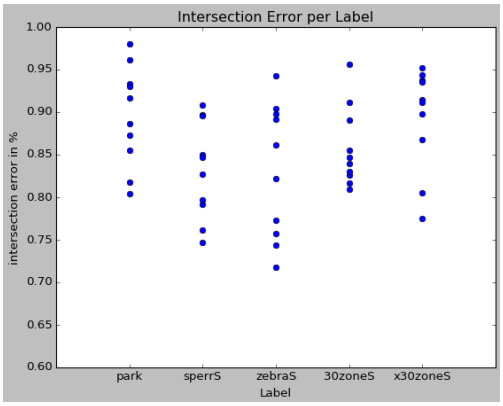

Figure 6: Bounding box intersection error of neural network trained on real data

For the generation of the virtual data the scripts for the Unreal Engine 4 that were developed are used to generate and automatically label 300 virtual images for each type of traffic sign. On this virtual training data a Yolo V2 network is trained and tested on the aforementioned 100 real test images. In the first attempt the neural network trained on the virtual data did not achieve satisfying results with classification results being significantly worse than the baseline network trained on real data. Many of the traffic signs were only recognized with a low accuracy or were not recognized at all. However, the intersection error of the bounding boxes that were found was very low in most cases which surprisingly outperformed the intersection error of the baseline networks predictions. This can be seen in the following pictures.

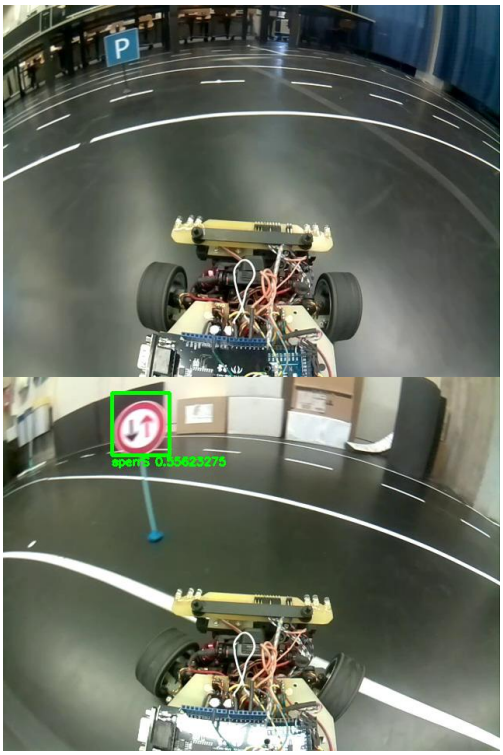

Figure 5: Output of neural network trained on virtual data V1

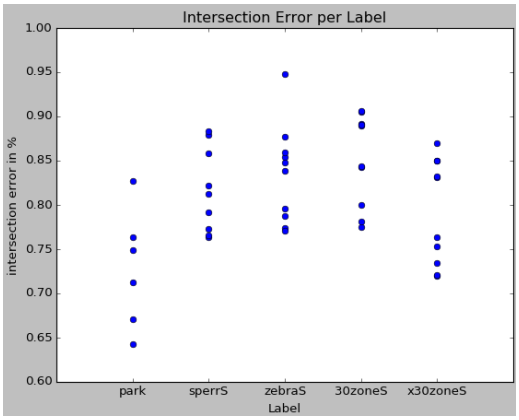

Figure 7: Bounding box intersection error of neural network trained on virtual data V1

In an effort to improve the results achieved with training on virtual data, it is first attempted to generate and use a higher number of virtual training data. This however did not improve results significantly while increasing training times exponentially, leading to the conclusion that the virtual data generated is not representative enough of the real scenario - garbage in, garbage out.

This is why as a next step it is attempted to improve the quality of the virtual data. Since problems in the object detection seem to arrive with images of blurred, distorted or tilted traffic signs as well as objects on the image borders it is attempted to reflect these issues in the virtual training data. Measures taken include adapting color and lighting condition of the scene and introducing motion blur as well as increasing the variety of randomly selected backgrounds for the scene making the virtual image data resemble the real image data as close as possible. Incremental changes regarding these parameters show varying improvements on different issues in the object detection with each new version. With the subsequent training of new neural networks on training data incorporating the described changes, the results continued to improve and the error rate and the number of misclassified signs declined.

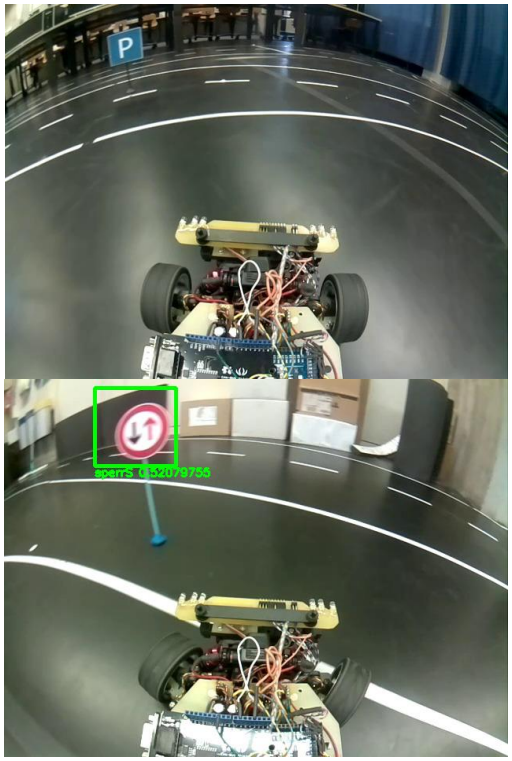

Figure 8: Output of neural network trained on virtual data V2

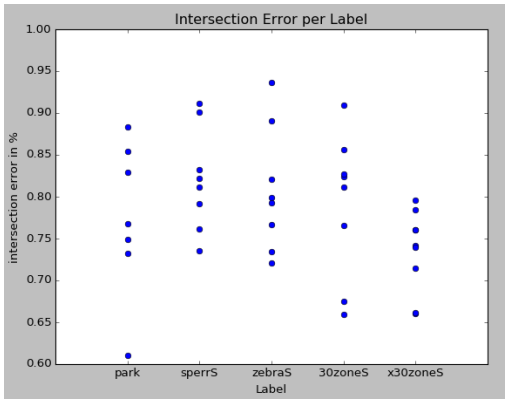

Figure 9: Bounding box intersection error of neural network trained on virtual data V2

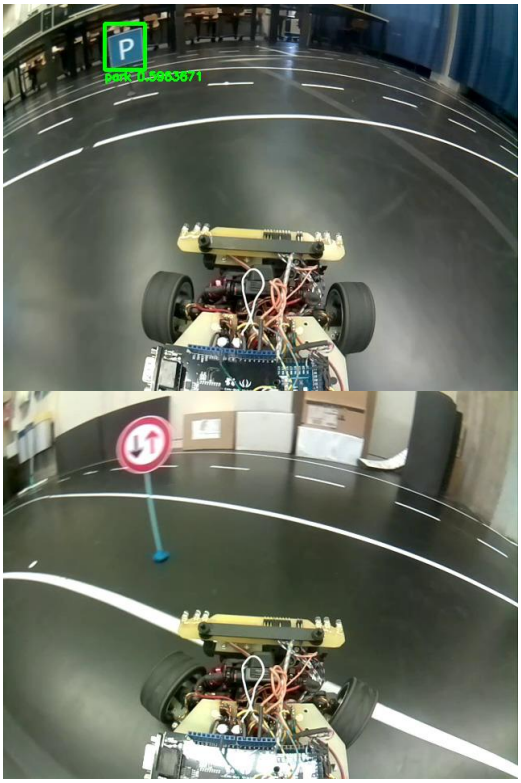

Figure 10: Output of neural network trained on virtual data V3

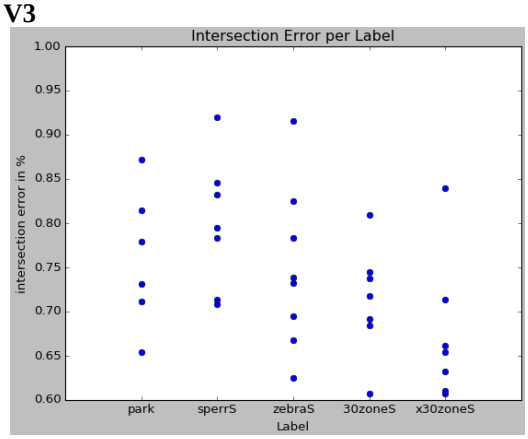

Figure 11: Bounding box intersection error of neural network trained on virtual data V3

The end result combining all the incremental improvements in the generation of virtual training data show a correct classification rate of 88% with an average detection accuracy of about 83%. This is lower than the accuracy of the baseline network trained on real data which classified all sample data correctly and achieved an average accuracy of about 95%. However the misclassification rate and detection accuracies kept improving with each measure taken to achieve more realistic virtual image data. This leads to the conclusion of the proof of concept that with a thorough setup of a virtual scene and incremental adaptation validated on a set of real and representative image data, virtual image data can be auto-generated and used to train neural networks for use with real image data successfully.

NEURAL NETWORK	PREDICTION ACCURACY	BOUNDING BOX INTERSECTION ERROR
BASELINE TRAINED ON REAL DATA	98%	0,855%
TRAINED ON VIRTUAL DATA V1	72%	0,806%
TRAINED ON VIRTUAL DATA V2	69%	0,768%
TRAINED ON VIRTUAL DATA V4	76%	0,705%
V1 + V2 + V3 COMBINED	83%	0,810%

Figure 12: Table of results

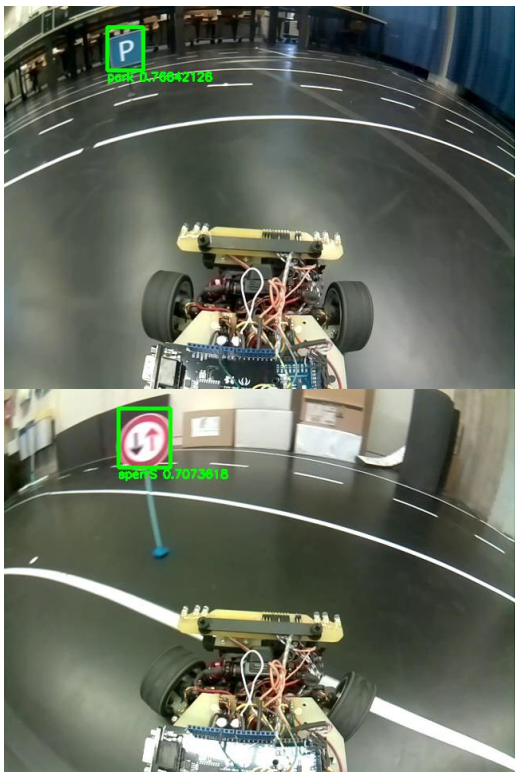

Figure 13: Output of neural network trained on virtual data V1+2+3

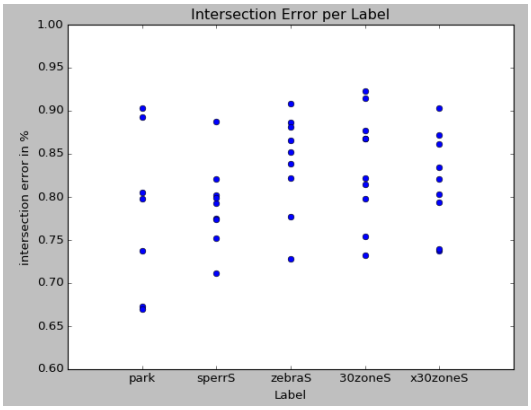

Figure 14: Bounding box intersection error of neural network trained on virtual data V1+2+3

V. Conclusion

As seen in the case study of this paper, virtual training data can be used to supplement real training data. However to achieve good results the virtual data has to resemble the general characteristics of the real data as close as possible. This includes distortions, changing lighting conditions and other imperfections that occur during the capture of real image data that should be reflected in the virtually generated image data. This initial process of setting up a suitable virtual scenario can be time consuming and needs to be validated on a real dataset. After this initial setup however limitless amounts of new training data can be created and labelled automatically in a fraction of the time needed to gather and label real image data. Additionally the virtual scenario can be easily adjusted and extended making it possible to quickly generate new training data to quickly adapt to changing requirements.

References

[1] Zukunft der Automobilindustrie. Online: https://www.tab-beimbundestag.de/de/pdf/publikationen/berichte/TAB-Arbeitsberichtab152.pdf (15.08.2018).

[2] National Science and Technology Council, Committee on Technology. 2016. Preparing for the Future of Artificial Intelligence. 2016.

[3] Capgemini. 2018. Online: https:// www.capgemini.com/it-trends (16.08.2018)

[4] DPA. 2018. Online: https://www.lead-digital.de/autonomes-fahren-wem-gehoeren-die-daten/ (16.08.2018)

[5] Marlon Bonazzi. 2018. Online: https://www.fool.de/2018/06/24/bmw-daimler-und-vw-droht-es-beim-autonomen-fahren-komplett-von-tesla-abgehaengt-zu-werden/ (15.08.2018)

[6] Fabian Patterson. 2018. Online: http://2018.wohnzukunftstag.de/wp-content/uploads/sites/16/2018/06/KI-StandderDingeundwodieForschunghingeht.pdf (15.08.2018)

[7] Scherk Johannes, Pächhacker-Tröscher Gerlinde, Wagner Karina. 2017. Online:https://www.bmvit.gv.at/innovation/downloads/kuenstliche_intelligenz.pdf (15082018)

[8] Atos. 2016. Journey 2020. Digital Shockwaves in Business.

[9] Li Fei-Fei, Johnson Justin, Karpathy Andrej. 2016. Online: http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf (15.08.2018).

[10] KI Basics. 2017. Machine / Deep Learning - Wie lernen künstliche neuronale Netze?. Online. https://jaai.de/machine-deep-learning-529/ (15.08.2018).

[11] Github tzutalin. 2015. Online: https://github.com/tzutalin/labelImg (15.08.2018).

[12] Github Microsoft VoTT. 2018. Online: https://github.com/Microsoft/VoTT (15.08.2018).

[13] Greis Friedhelm. 2018. Aufwand Labeling. Online: https://www.golem.de/news/neuronale-netze-wiestudenten-autonome-autos-schlau-machen-1807-135579-2.html (30.07.2018)

[14] Da Silva Ivan Nunes, Spatti Danilo Hernane, Flauzino , Rogerio Andrade, Bartocci Liboni Luisa Helena, Dos Reis Alves Silas Franco. 2017. Artificial Neural Networks. Cham: Springer International. 2017.

[15] RAND. 2017. An Intelligence in our Image. The Risks of Bias and Errors in Artificial Intelligence.

[16] AI Now Initative. 2016. The AI Now Report. The Social and Economic Implications of Artificial Intelligence Technologies in the Near-Term. A summary of the AI Now public symposium, hosted by the White House and New York University's Information L aw Institute, July 7th, 2016.

[17] Lean Yu, Shouyang Wang, Kin Keung Lai. 2007. Foreign-Exchange-Rate Forecasting with Artificial Neural Networks. International Series in operations research and management science. New York: Springer Science+Business Media. 2007.
[18] Epic Games, Inc. Online: https://www.unrealengine.com/ (01.08.2018)
[19] Crisp Research. 2017. Machine Learning im Unternehmenseinsatz. Künstliche Intelligenz als Grundlage digitaler Transformationsprozesse.
[20] https://www.forbes.com/consent/?toURL=https://www.forbes.com/sites/kevinmurnane/2016/05/05/how-deep-learning-networks-can-use-virtual-worlds-to-solve-real-world-problems/
[21] https://download.visinf.tu-darmstadt.de/data/from_games/data/eccv-2016-richter-playing_for_data.pdf
[22] https://download.visinf.tu-darmstadt.de/data/from_games/data/eccv-2016-richter-playing_for_data.pdf
[23] https://www.forbes.com/consent/?toURL=https://www.forbes.com/sites/kevinmurnane/2016/05/05/how-deep-learning-networks-can-use-virtual-worlds-to-solve-real-world-problems/
[24] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A.; 2016; https://pjreddie.com/publications/ (28.10.2018)