Advanced Logistic Systems – Theory and Practice, 13(1), 37-45, https://doi.org/10.32971/als.2019.010

SIMULATION OF IMAGE DATA TO SUPPORT THE TRAINING OF CONVOLUTIONAL NEURAL NETWORKS FOR OBJECTS RECOGNITION

HAGEN BORSTELL[1]-JAN NONNEN[2]

Abstract: The recognition of logistics objects is an essential prerequisite for the optimization of operational logistics processes and can be performed among others via image-based methods. However, the lack of available data for training domain-specific recognition models remains a practical problem. For this reason, we present an approach to solving this problem. The core principle of our approach is the automated generation of image data from 3D models, in which the appearance of the objects varies through variations of different parameters. The first results are promising: Without any real image data, we have created a neural network for recognition of real objects with a recall quality of 86%.

Keywords: Logistics; Image Processing; Deep Learning; Simulation

1. INTRODUCTION

The recognition of logistics objects is an essential prerequisite for the optimization of operational logistics processes. Image-based object recognition has evolved enormously in recent years due to the emergence of deep neural networks [13]. Databases with training data as well as pre-trained neural networks are available for different applications scenarios, e.g. the YOLO v2 deep neural network for real-time object detection [19]. Moreover, advanced but easy to use deep learning software such as TensorFlow is available [1]. From a perspective of identifying logistics objects, the question arises whether identification can be performed either based on pre-trained networks or by means of training based on existing image databases. This question can be answered as follows: Although the number of training data and networks is constantly increasing, the resulting models are not always sufficient for domain-specific applications in logistics. In Figure 1 a simple example is shown. Here, the YOLO v2 deep neural network was used for object detection. Although the network is applicable without any model adjustment for monitoring parking areas and people detection for transshipment monitoring, it cannot be used directly for detection of forklifts (see Figure 1).

Therefore, if objects need to be recognized that were not represented in the training data set, training and test data for these objects must be collected first. Afterwards, neural networks can be built. Since usually a lot of (several thousand or more) data samples (images) per object are necessary for the training process, the time-consuming creation of the training data set is a significant problem in this context. Approaches to solving this problem include transfer-learning [18] and data augmentation [16]. Transfer learning reduces this problem by using a small amount of training data to adapt a pre-trained

[1] Otto von Guericke University Magdeburg
hagen.borstell@ovgu.de
Germany
[2] Viaboxx GmbH
jan.nonnen@viaboxx.de
Germany

This paper is a secondary publication of the article presented at the COMEC 2019 conference in Cuba, published with the permissions of the organization board of the conference and the authors.

network to a specific application domain. Data augmentation reduces this problem by artificially extending the training data through a series of image transformations. However, a considerable effort remains for data collection. For this reason, we present an approach to support the training of neural networks with simulated image data that can be generated automatically.

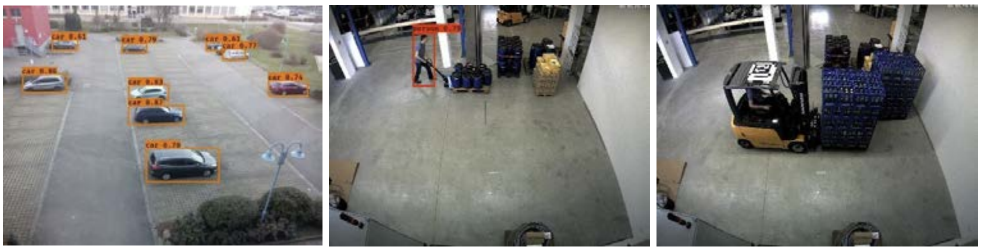


Figure 1. The detection result of logistics applications scenarios based on the YOLO v2 deep neural network [19]: (left) Good performance in car detection for parking lot monitoring, (middle) Good performance in people detection for transshipment monitoring, (right) Bad performance in forklift detection for transshipment monitoring.

2. Related Work

Image-based objects detection can contribute to various application fields of logistics, such as traceability and trackability, occupancy detection of storage and traffic areas, security and protection of infrastructure, manual picking and packing, or automation of material handling systems [5]. From a methodological point of view, image-based object recognition can be differentiated according to the type of feature generation (and selection) and feature evaluation within the classification procedure. In a classical approach features (or rules) are generated manually and evaluated regarding a certain class either via neural networks, statistical methods such as the k-nearest-neighbor-algorithm, or rule-based methods [6]. Since feature generation and selection play a decisive role, much research has been carried out in this area [11]. In the field of image-based object recognition, the advent of convolution neural networks with automated feature generation has resulted in an enormous increase in the performance of classifiers [2]. This has opened up new areas of application in logistics as well, such as the monitoring of parking lots [3], packaging processes [15], or picking processes [9]. Since the generation of data for the creation of algorithmic models for image-based object recognition and detection is time-consuming, researchers have begun to address simulation of image data to support the development of image-based monitoring systems. Thamer & Weimar used simulated 3D data in order to support the development of automated handling processes [22]. Borstell & Reggelin (2019) have embedded the simulation of image data into a virtual commissioning framework to support design, implementation, and testing of image processing algorithms for logistics application scenarios. Regarding convolution neural networks, first attempts have been made to use simulated image data. Li et al. [14] proposed a virtual image data set for traffic vision research. Sarkar, Varanasi, and Stricker [20] examined the usage of non-photorealistic 3D CAD models for real-world object detection.

3. RESEARCH FRAMEWORK

The core principle of the research framework, shown in Figure 2, is the automated generation of image data from 3D model libraries, in which the appearance of the objects varies through variations of parameters such as illumination, rotation, material properties.

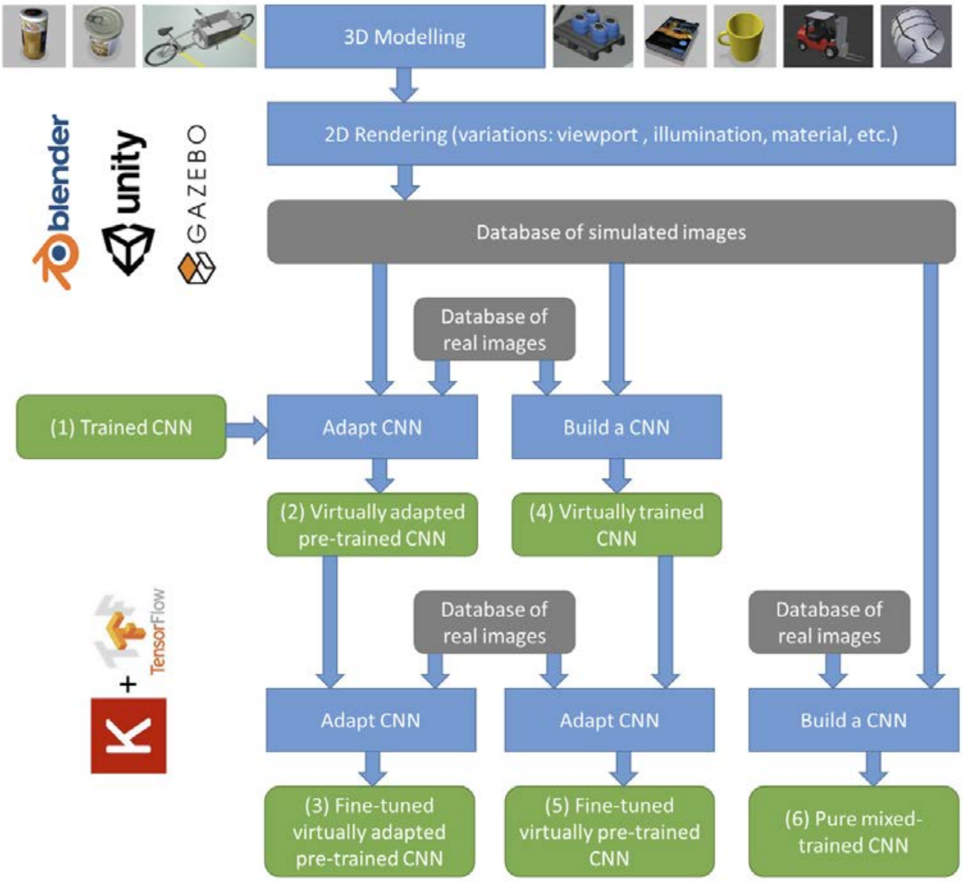

3D Modelling
2D Rendering (variations: viewport , illumination, material, etc.)
Database of simulated images
blender
unity
GAZEBO
Database of real images
(1) Trained CNN
Adapt CNN
Build a CNN
(2) Virtually adapted pre-trained CNN
(4) Virtually trained CNN
TensorFlow
+
K
Database of real images
Database of real images
Adapt CNN
Adapt CNN
Build a CNN
(3) Fine-tuned virtually adapted pre-trained CNN
(5) Fine-tuned virtually pre-trained CNN
(6) Pure mixed-trained CNN

Figure 2. Research Framework

Tools such as Blender [4], Unity [23] or Gazebo [12] can be used for this purpose. Within these tools, the rendering process can be automated via programming languages such as Python or C#. As an example, various positions of the rendering device can be generated in Blender (see Figure 3, top). In the same way, other properties of the objects or scene can be automatically adjusted in the animation sequence. This way multiple images with different properties of the objects can be rendered with one animation procedure (see Figure 3, bottom). The generated images form image databases, which can be used for the creation of neural networks or for the adaption of pre-trained neural networks. This idea assumes that essential image characteristics of the real objects are also represented to a certain degree by the virtual objects.

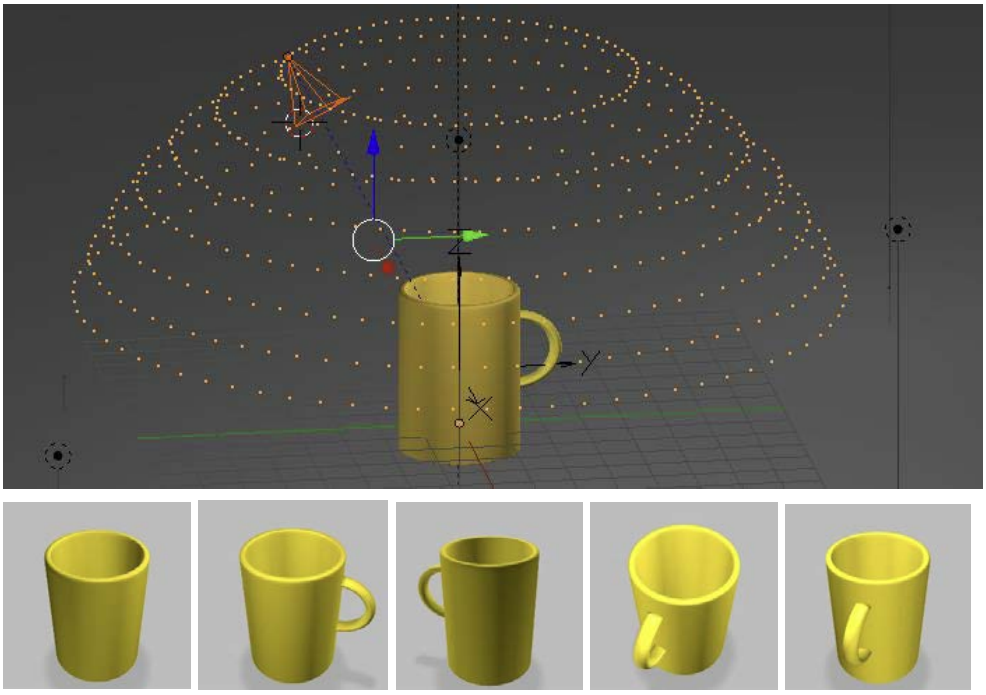


Figure 3. (Top) Automated rendering in Blender, adapted based on http://thoro.de/page/3dnp-introduction-en, (bottom) Example rendering results.

A variety of tools and libraries are available for training neural networks [8]. In our framework, we use Keras with Tensorflow backend [7] for this purpose. The simple creation of models using Keras and the strengths of TensorFlow, including evaluation tools such as TensorBoard, led to this selection. When using simulated data for training a neural network, which is later applied to real data, one faces the problem of training and testing with different distributions. A strategy to mitigate this problem is to use a small amount of data of the target distribution [17]. Another strategy could be to achieve a maximum degree of reality of simulated data. As shown in Figure 2, our approach also integrates the ideas of transfer learning. In this regard, the aim is to examine how existing networks can be adapted to domain-specific tasks using virtual data. In a nutshell, various strategies for applying models exist and will be considered in research:

1) Creating a network from scratch or using a pre-trained neural network directly to classify objects (based on real images)
2) Adapting network type (1) by using simulated image data
3) Fine-tuning network type (2) by using real image data
4) Creating a neural network from scratch by using simulated image data
5) Fine-tuning network type (4) by using real image data
6) Creating a neural network from scratch by using real and simulated image data in a mix

The first strategy should be applied if either enough real images are available to train a neural from scratch or a pre-trained network is already available, e.g. YOLO v2 deep neural network [19] for parking lot monitoring. If this is not the case, the strategies with simulated image data can provide a possible solution. Currently, only a small amount of data is available regarding these strategies and therefore have to be generated first. Various research questions arise in this context:

- With what level of detail images have to be simulated in order to achieve sufficient model performance?
- Which performance can be achieved by the different strategies of using simulated image data in order to detect logistics objects?
- How can the best strategy for the development of a model be selected for a given application scenario?

These questions are to be clarified within the framework of the research on the basis of logistics application scenarios in the future.

4. Initial Evaluation and Early Findings

As a first research step within the research framework, we used a picking and packing scenario in mail-order business. In addition to the basic designing, implementation and testing of the components of the research framework, the objective of this application scenario was to gain initial findings regarding the simulation of image data in a field with practical relevance (Figure 4).

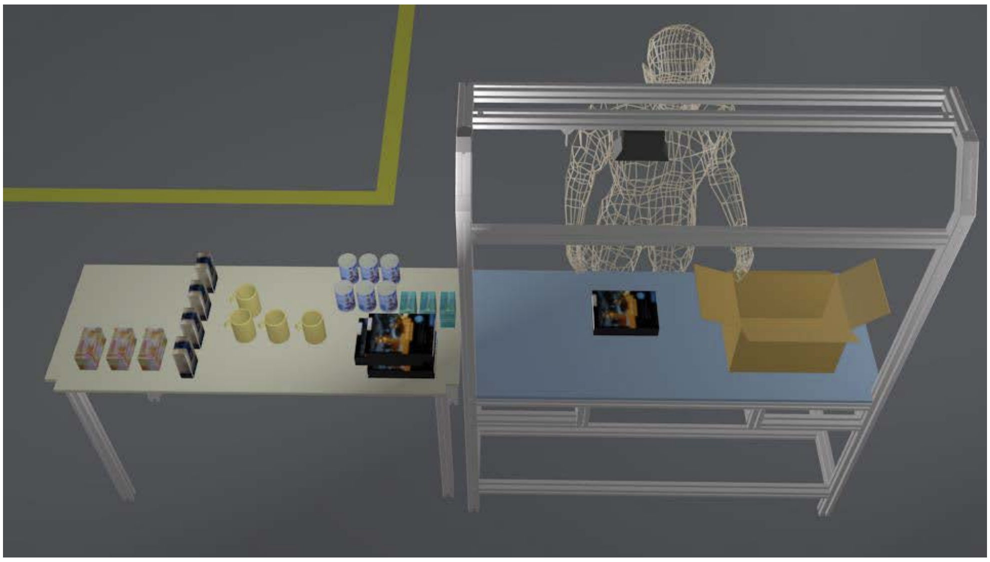


Figure 4. Application scenario: Picking and packing in mail-order business

In this test scenario, six different types of goods in the picking and packing process are to be recognized. It is assumed that no image data is available to train the models, but only information about the geometry and appearance (e.g. color, texture) of the objects to be recognized. This assumption seems to be valid and reasonable, as users often have no visual data available about their objects, but these are often available via product databases or design drawings. Based on this information, textured 3D models of the six objects were created. The rendering process has been automated and thus images from different directions have been rendered with different light sources and different background (see Figure 3).

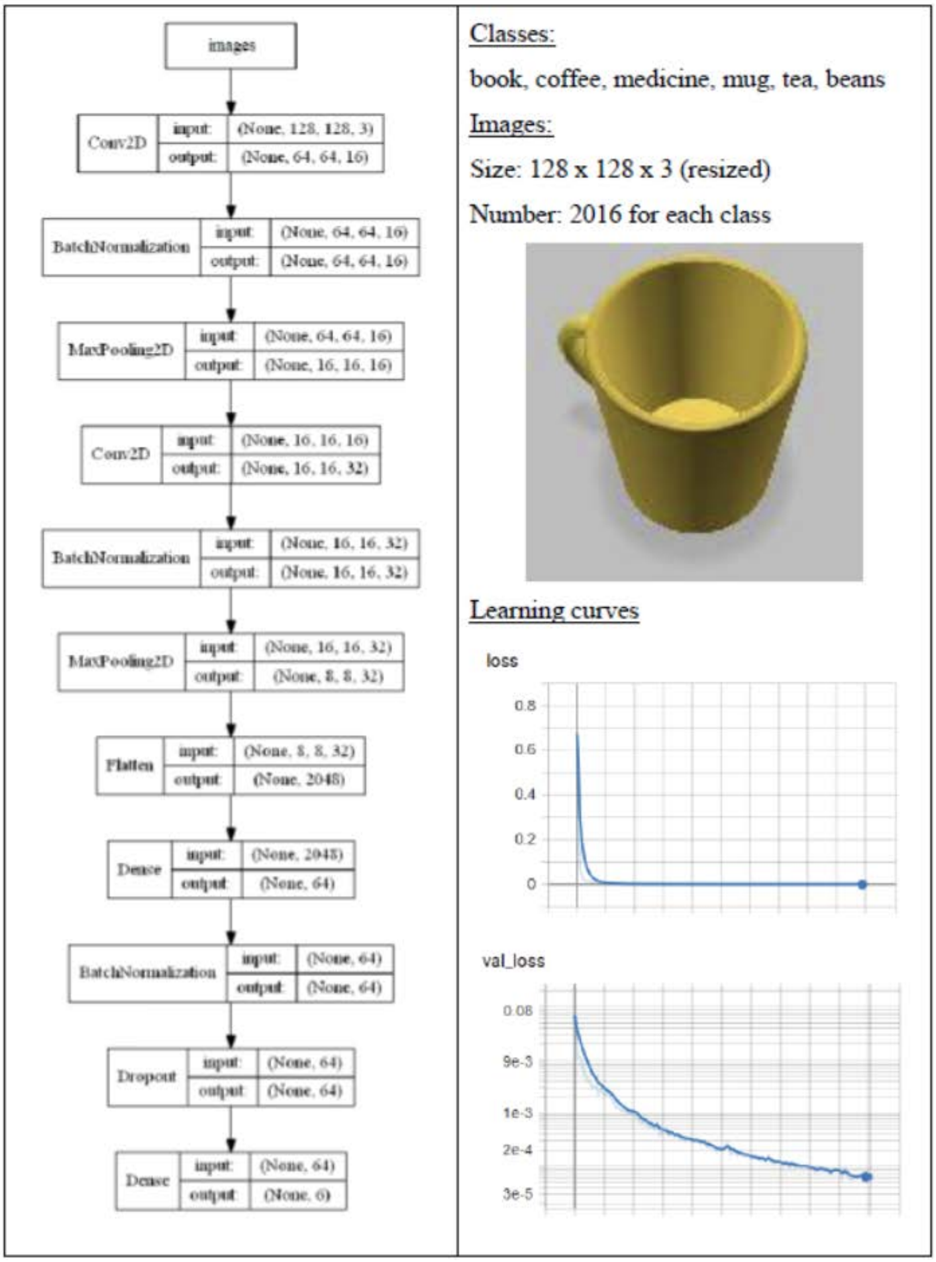
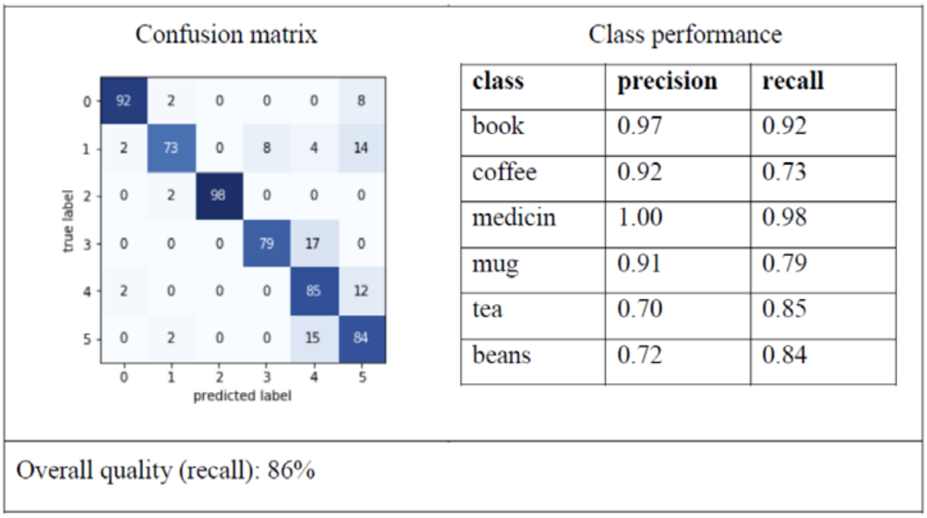

images
Conv2D | input: (None, 128, 128, 3) | output: (None, 64, 64, 16)
BatchNormalization | input: (None, 64, 64, 16) | output: (None, 64, 64, 16)
MaxPooling2D | input: (None, 64, 64, 16) | output: (None, 16, 16, 16)
Conv2D | input: (None, 16, 16, 16) | output: (None, 16, 16, 32)
BatchNormalization | input: (None, 16, 16, 32) | output: (None, 16, 16, 32)
MaxPooling2D | input: (None, 16, 16, 32) | output: (None, 8, 8, 32)
Flatten | input: (None, 8, 8, 32) | output: (None, 2048)
Dense | input: (None, 2048) | output: (None, 64)
BatchNormalization | input: (None, 64) | output: (None, 64)
Dropout | input: (None, 64) | output: (None, 64)
Dense | input: (None, 64) | output: (None, 6)
Classes:
book, coffee, medicine, mug, tea, beans
Images:
Size: 128 x 128 x 3 (resized)
Number: 2016 for each class
Learning curves
loss
0.8
0.6
0.4
0.2
0
val_loss
0.08
9e-3
1e-3
2e-4
3e-5

Figure 5. (left) Network structure, (right) Data, classes, and learning curves

To generate models for the recognition of objects in images, neural networks with convolution layers are particularly suitable. The concrete structure (number and type of layers) of the network with convolutional layers depends on various factors such as the number of data records, the number of classes and the available computational resources. The structure of the neural network can range from simple sequential structures with few convolution layers to complex structures such as the Inception network [21] or residual neural networks [10]. In our early experiments, we initially used very basic convolutional networks with two or three convolutional layers, supported by layers to avoid overfitting (Batch normalization, dropout, pooling) and to achieve a certain level of robustness to deviations of scale or orientation (pooling). An example is shown in Figure 5 (left). Here, a convolutional neural network, fed by images of size 128x128x3 (i.e. three-channel color images of size 128x128 pixels) and with two convolutional layers and one fully-connected layer, is shown. In this example, the network was trained with 12096 virtual images to recognize objects regarding six classes: book, coffee, medicine, mug, tea, beans. The training and validation dataset was created by a split of 0.8/0.2. In Figure 5 (right), the training process is shown. Both the training loss and the validation loss are continuously reduced and reach their minima after 100 epochs.

An accuracy of 100% is achieved. However, since real images are used in the recall, a lower recognition rate can be expected due to the different distribution of the image characteristics during training (virtual data) and recall (real data). This property is associated with the term data mismatch. To examine the effects of data mismatch, we have applied the trained neural network to real data. For this purpose, we have captured 100 images per class with two different cameras in the scenario described above. The results are shown in Figure 6. We achieved an overall recall quality of 86%. This initially confirms the viability of the concept. However, we have also observed problems when the number of classes becomes too high or when there is too little shape or texture information for the virtual images to be created.



Confusion matrix

true label \ predicted label	0	1	2	3	4	5
0	92	2	0	0	0	8
1	2	73	0	8	4	14
2	0	2	98	0	0	0
3	0	0	0	79	17	0
4	2	0	0	0	85	12
5	0	2	0	0	15	84

Class performance

class	**precision**	**recall**
book	0.97	0.92
coffee	0.92	0.73
medicin	1.00	0.98
mug	0.91	0.79
tea	0.70	0.85
beans	0.72	0.84

Overall quality (recall): 86%

Figure 6. (left) Classification result of real images with a CNN that was created without any real images: (left) Confusion matrix, (right) Class performance

5. CONCLUSIONS

The recognition of logistics objects is an essential prerequisite for the optimization of operational logistics processes. Image-based object recognition has evolved enormously through the advent of convolutional neural networks, but the lack of available data for training domain-specific networks remains a practical problem. As shown in the paper, the usage of simulated image data can be a solution approach. The first results are promising: Without any real image data, we have created a neural network for classifying real objects by rendering simulated images with 3D software using geometry and texture information that could be available in product databases, for example. Within the presented research framework, however, further research is now necessary in order to further automate and generalize the processes. In particular, research questions have to be answered regarding the necessary degree of detail of the image data to be simulated. With the knowledge and methods resulting from this, service-oriented solutions for the automated creation of neural networks for specific object classes in specific application scenarios could be developed.

Acknowledgements

The authors are partly funded by the Federal Ministry for Economic Affairs and Energy on the basis of a resolution of the German Bundestag under the "Central Innovation Programme for SMEs" (ZIM) in the R&D project "Track4Goods". The R&D project was developed within the ZIM network "NekoS".

References

[1] Abadi, M. et all (2016). *TensorFlow: A system for large-scale machine learning.* Retrieved from http://arxiv.org/pdf/1605.08695v2

[2] Alom, M. Z. et all (2018). *The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches.* CoRR, abs/1803.01164.

[3] Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C. & Vairo, C. (2017). Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications, 72*, 327–334. https://doi.org/10.1016/j.eswa.2016.10.055

[4] Blender Online Community. (2019). *Blender - a 3D modelling and rendering package.* Retrieved from http://www.blender.org

[5] Borstell, H. (2018). A short survey of image processing in logistics. In M. Schenk (Ed.), *11th International Doctoral Student Workshop on Logistics,* Magdeburg: Universität Magdeburg. 43–46.

[6] Cho, T.-H., Conners, R. W. & Araman, P. A. (1991). A comparison of rule-based, k-nearest neighbor, and neural net classifiers for automated industrial inspection. In J. Feinstein (Ed.), *Proceedings of the IEEE/ACM International Conference on Developing and Managing Expert System Programs:* September 30-October 2, 1991, Washington, D.C. Los Alamitos, Calif: IEEE Computer Society Press. 202–209. https://doi.org/10.1109/DMESP.1991.171738

[7] Chollet, F., & others. (2015). *Keras*. Retrieved from https://github.com/fchollet/keras

[8] Erickson, B. J., Korfiatis, P., Akkus, Z., Kline, T. & Philbrick, K. (2017). Toolkits and Libraries for Deep Learning. *Journal of Digital Imaging, 30*(4), 400–405. https://doi.org/10.1007/s10278-017-9965-6

[9] Grzeszick, R., Feldhorst, S., Mosblech, C., Fink, G. A., & Hompel, M. ten. (2016). Camera-assisted Pick-by-feel. *Logistics Journal: Proceedings, Vol. 2016.* https://doi.org/10.2195/lj_Proc_grzeszick_en_201610_01

[10] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR). IEEE. 770–778. https://doi.org/10.1109/CVPR.2016.90

[11] Katz, G., Shin, E. C. R., & Song, D. (2016). Explorekit: Automatic Feature Generation and Selection. In F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z.-H. Zhou, & X. Wu (Eds.), *16th IEEE International Conference on Data Mining:* 12-15 December 2016, Barcelona, Catalonia, Spain: proceedings. Piscataway, NJ: IEEE. 979–984. https://doi.org/10.1109/ICDM.2016.0123

[12] Koenig, N., & Howard, A. (2004). Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2149–2154, https://doi.org/10.1109/IROS.2004.1389727

[13] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

[14] Li, X., Wang, K., Tian, Y., Yan, L., Deng, F., & Wang, F.-Y. (2018). The ParallelEye Dataset: A Large Collection of Virtual Images for Traffic Vision Research. *IEEE Transactions on Intelligent Transportation Systems*, 1–13. https://doi.org/10.1109/TITS.2018.2857566

[15] Logivations GmbH. (2018). *Kamerabasierte Objekterkennung - Deep Machine Learning in Verbindung mit Computer Vision.* Retrieved from https://www.logivations.com/de/land/pdf/KameraBasierte_Objekterkennung.pdf

[16] Luke Taylor, & Geoff S. Nitschke. (2017). Improving Deep Learning using Generic Data Augmentation. *CoRR*, abs/1708.06020, https://doi.org/10.1109/SSCI.2018.8628742

[17] Ng, A. (2018). *Machine Learning Yearning: Technical Strategy for AI Engineers, In the Era of Deep Learning.* Retrieved from https://www.mlyearning.org/

[18] Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[19] Redmon, J., & Farhadi, A. (2016). *YOLO9000: Better, Faster, Stronger*. Retrieved from http://arxiv.org/pdf/1612.08242v1, https://doi.org/10.1109/CVPR.2017.690

[20] Sarkar, K., Varanasi, K., & Stricker, D. (2017). Trained 3D Models for CNN based Object Recognition. In *VISIGRAPP 2017*, 130-137, https://doi.org/10.5220/0006272901300137

[21] Szegedy, C. et all. (2015). Going Deeper with Convolutions. In Computer Vision and Pattern Recognition (CVPR). Retrieved from https://arxiv.org/abs/1409.4842, https://doi.org/10.1109/CVPR.2015.7298594

[22] Thamer, H., & Weimer, D. (2013). Software simuliert Sensorik - 3D-Bildverarbeitung für die Logistikautomatisierung. *Hebezeuge Fördermittel, 2016*(5), 252–254.

[23] Unity Technologies. (2019). *Unity User Manual (2019.1 beta).* Retrieved from https://docs.unity3d.com/2019.1/Documentation/Manual/