Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2024) 000–000

www.elsevier.com/locate/procedia

18th CIRP Conference on Intelligent Computation in Manufacturing Engineering

A machine learning based approach for value stream map digitization

Florian Mitschke*[,a], Sven Spieckermann[b], Joachim Metternich[a]

[a]Institute for Production Management, Technology and Machine Tools | TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, Germany
[b]SimPlan AG, Sophie-Scholl-Platz 6, 63452 Hanau, Germany

* Corresponding author. Tel.:+49-6151-8229641; *mail address:* f.mitschke@ptw.tu-darmstadt.de

Abstract

The Value Stream Mapping method provides a comprehensive overview of a process, pinpoints inefficiencies, and thereby offers a roadmap for eliminating waste and enhancing added value. Simplifications are often made enabling manual evaluations but at the same time sacrificing valuable information about dynamic process behavior. Simulation software enables the aspect of dynamics to be considered. This software requires a digital representation of the value stream map created during Value Stream Mapping, which is typically created in non-digital formats unusable in software. This paper presents an approach that enables the automated digitization of such non-digital value stream maps.

Peer-review under responsibility of the scientific committee of the 18th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 10-12 July, Gulf of Naples, Italy

Keywords: Value Stream Mapping, synthetic image data, computer vision, schematic digitization, machine learning

1. Introduction

Value Stream Mapping is recognized as an effective lean tool for visualizing, analyzing, and improving production processes by identifying and eliminating waste, thus creating added value. Originating from the Toyota Production System [1], Value Stream Mapping has been widely adopted across various industries to optimize production and service operations. The method involves mapping out all activities required to bring a product or service from conception to delivery to the customer, categorizing these activities into value-adding and non-value-adding, and then strategizing on eliminating the latter. [2]

The usage of simulation software for value streams brings the potential of analyzing their dynamic behavior compared to traditional non-simulation-aided analysis in which simplifications have to be assumed that do not allow for dynamic behavior analysis [3]. Such software needs a digital representation of the value stream map (VSM) that defines a value stream's properties by illustrating and parameterizing the underlying logic of process execution [4].

VSMs are traditionally created using a pen-and-paper approach [5], which typically involves whiteboards or large sheets of paper. Manual digitization of these VSMs is time-consuming and creates the feeling of having to do the same work twice. This may already lead to an inhibition threshold that results in a rejection of the use of simulation software which reflects the necessity of developing a functionality for improved workflow in this regard.

In this paper, a novel approach for the automated digitization of analog VSMs is proposed. This study focuses on two things. First, the overall methodology in the form of a computation pipeline beginning from the photo of a VSM to a fully digital representation is introduced. This pipeline includes machine learning (ML) models used for image interpretation. Secondly, an approach for dataset enrichment is proposed. This approach aims to solve the problem of data scarcity hampering the development of performant ML models for this use case.

2212-8271
Peer-review under responsibility of the scientific committee of the 18th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 10-12 July, Gulf of Naples, Italy

2. Background

Despite the considerable age of the Value Stream Mapping, it is still the subject of current research. This method is being expanded to include aspects such as sustainability [6], worker ergonomics and health [7], complex information flows regarding Industry 4.0 [8], and dynamic process behavior [9]. These developments show that there is still great potential for expansion in the Value Stream Mapping method.

The VSM is a schematic visually depicting process entities, material flows, and information exchange within the value stream. Either ready-made templates in the form of magnetic cards, sticky notes, or pure hand drawings can be used when creating the VSM. Figure 1 depicts a template-based and a hand-drawn example of a VSM.

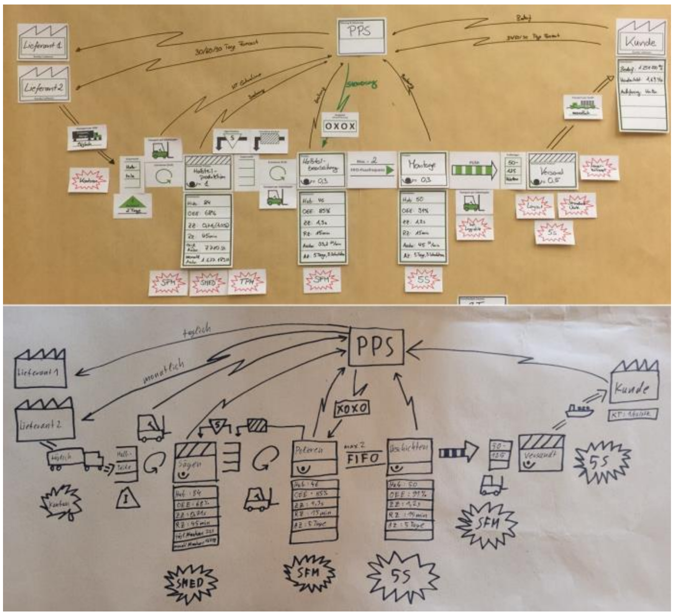

Fig. 1: Hand-drawn and template-based VSM.

Traditional Value Stream Mapping using a manual pen-and-paper approach is typically based on a lot of simplifying assumptions. For process times, replenishment times, throughput times, stock levels, etc., average values are applied. However, it is often desirable to understand how backlogs in buffers are building up, why processes are over- or under-utilized, and how shared resources may be operated best. The main motivation for simulating value streams is to overcome the use of simplified averages, thus enabling dynamic evaluations of e.g., utilization of processes, product stocks, etc. The vast amount of work on the combination of simulation and lean approaches presented e.g., in the comprehensive literature review by [10], demonstrates how beneficial this connection is.

VSMs embed semantic details through the specific symbols and arrows utilized, alongside parametric details via handwritten notes on these symbols, including any data boxes [11]. Both, the written and the information corresponding to the symbols present, are of interest for VSM digitization. While Toyota played a significant role in popularizing Value Stream Mapping, it is important to note that the specific symbols used may have been refined and adapted by various Lean practitioners over the years [12]. As a result, there is not a single, universally standardized set of symbols, and variations may exist based on specific industry practices or individual preferences. These high possible variances make manual feature definition practically impossible which predestines this task for being solved using ML [13]. Publicly available implementations of architectures such as Rotated RetinaNet [14] or YoloV5 OBB [15] enable the rotation sensitive detection of objects (ROSD) in images as well as an estimation of their corresponding rotations with a high quality and acceptable latencies. Further, foundation ML models, especially Google's Cloud Vision API [16], have recently shown superior performance in optical character recognition (OCR) and offer easy-to-implement API access [17].

Value Stream Mapping can be regarded as a niche problem with a very limited database for ML model training. Recent research has shown the potential of video games and simulation software as well as generative artificial intelligence in the creation of synthetic training data [18–20]. Synthetic data is often seen as an inferior substitute for real-world data. However, this view ignores the true potential of synthetic data. A study by Gartner predicts that synthetic data will overshadow real-world data by 2030 [21]. Synthetic data is generated artificially which distinguishes it from real-world data that originates from observations or measurements in the real world. Although real-world data, when available, is still the best way to represent information that is indicative of the underlying problem, synthetic data can address typical problems inherent in real-world data. The following are of particular interest for the underlying use case:

- Data privacy and sensitivity: Real-world datasets often contain sensitive or personally identifiable information, raising privacy concerns and legal issues e.g., VSMs contain sensitive information that is of interest to competitors [22].
- Annotation costs and accuracy: Annotating data can be severely time-consuming and costly. Furthermore, manual annotations are prone to errors. Synthetic data can be annotated automatically with perfect accuracy. [18, 19]
- Model robustness, generalization, and edge cases: Synthetic data can be generated on a large scale, and, depending on the quality, large variances can be induced that cover a wide range of edge cases that are rarely, if ever, present in the limited real-world data available. [23]

3. Methodology

In this section, the underlying processing pipeline for automatized VSM digitization is introduced. Further, a simple generator for synthetic hand drawing data generation to overcome the problem of data scarcity present in this use case is presented.

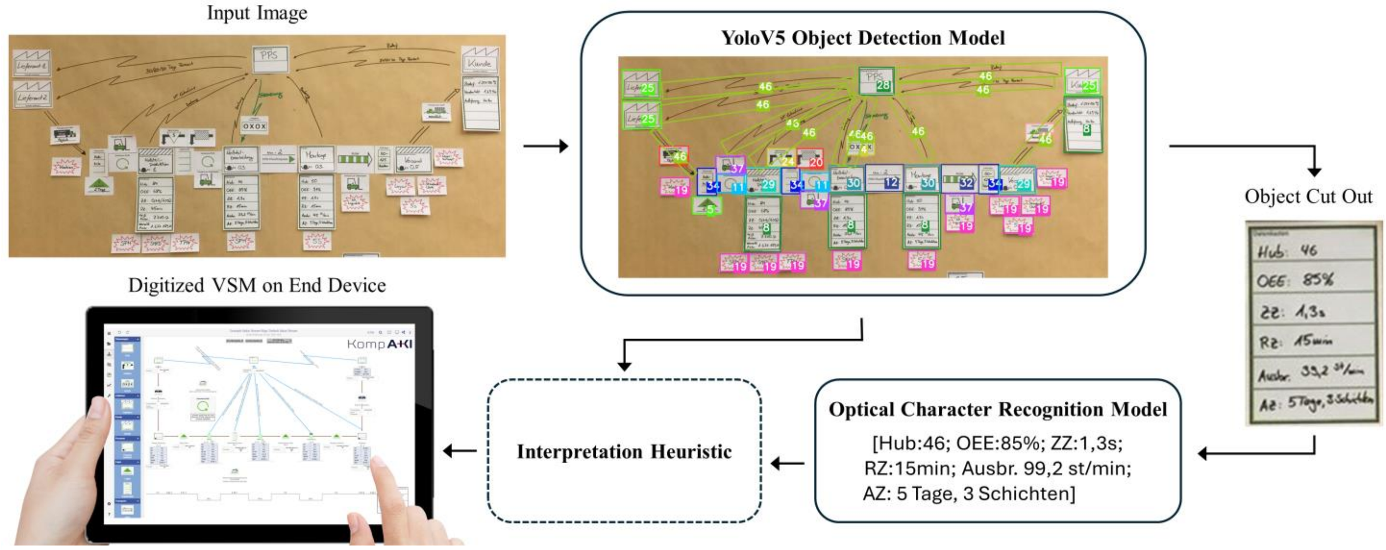

Fig. 2: VSM digitization processing pipeline.

3.1. VSM digitization.

Figure 2 depicts the overall methodology for VSM digitization. RSOD is used to identify and infer the position and orientation of symbols, enabling the reconstruction of their overall arrangement. For the reasons described in Chapter 2, an ML model is used in this context. After that, an OCR model is used to extract the parametric information associated with each symbol respectively.

A heuristic can be used to reconstruct the holistic VSM. The semantics of the VSM can be inferred using neighborhood information of the symbols as well as connections via arrows and by considering a predefined set of fundamentally possible connections. Process parameters are labeled in VSMs using keywords and abbreviations. These can be used to assign corresponding values to the parameters considered in the digitized VSM representation. This avoids the need for training data to tune an ML model for solving this task and represents an explainable approach.

3.2. Addressing data scarcity for hand drawing recognition.

As described in Chapter 2, VSMs can be created using premade templates or they can be drawn entirely by hand. Both, template-based and purely hand-drawn VSMs, show hand-drawn arrows carrying important information about a VSM's schematics. Consequently, the recognition of hand drawings is key for VSM digitization. The experiments described in Chapter 4 conclude that there is a need for more quality training data of hand-drawn arrows to enable the recognition of VSMs.

A targeted creation and annotation of sample data is time-consuming and would also only produce data that shows a low degree of variance. The low variance is to be expected because only a manageable number of creators would be available for data generation, and these can only deviate to a limited extent from their characteristic drawing style. This also reflects in the fact that every person has a very characteristic handwriting from which it is difficult to deviate.

To address this problem of data scarcity, a novel approach for the generation of hand drawing data for ML model training is presented. The approach is based on the idea of creating data synthetically instead of conducting manual creation. Figure 3 shows the logical sequence of the processing steps the generator uses to create a simple arrow in a hand-drawn style.

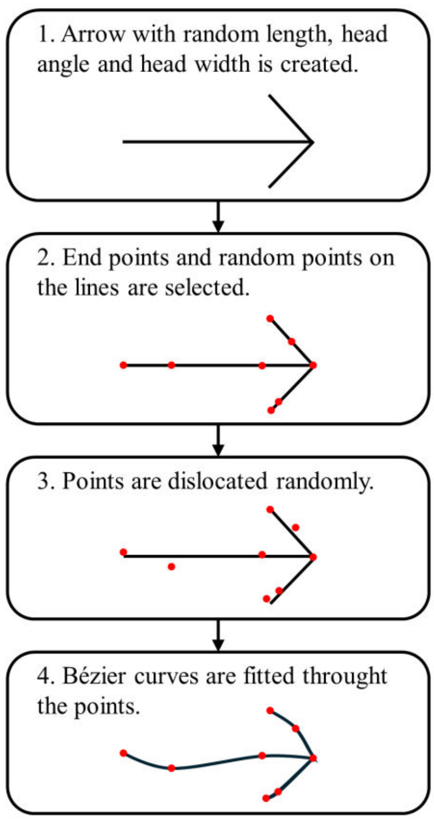

Fig. 3: Arrow generator processing pipeline.

The drawings used in VSMs are function-oriented and are not embellished with complex textures, shading, or non-purposeful components. All types of value stream symbols can be composed of simple geometric shapes such as straight lines, circles, or arcs. This means that it is possible to simply assemble symbols from basic geometric objects that can be defined in the form of code.

Firstly, a general definition of the arrow is made, with a minimum and a maximum for the arrow length, head angle, head width, and curvature of the lines that make up the arrow. Now a first completely straight arrow is created, whose length, head angle, and head width correspond to random values in the predefined minimum and maximum range. Points are randomly selected on the lines that make up the arrow and are moved orthogonally to the respective line within a random predefined range. The number of points selected, and the minimum and maximum shift range determine how and to what extent an arrow is curved. The curvature is achieved by drawing Bézier curves through the shifted points [24]. The generator can produce and annotate vast amounts of arrows in a short time. As their appearance is based on random components, a large spectrum of variations is covered. After completing the generation pipeline, each arrow's line thickness is determined randomly, the arrow is rotated, and placed on a canvas mimicking brown moderation paper or white boards located in a room with dynamic shadow throw present. Figure 4 depicts a selection of arrows created by the current generator implementation which are ready to be added to a dataset for hand drawing recognition.

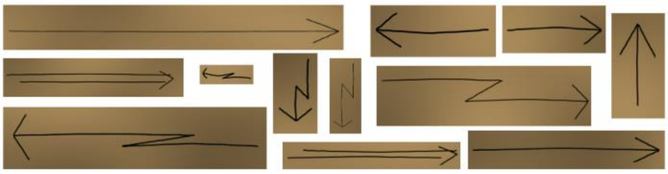

Fig. 4: Arrows created by generator.

4. Experiments

In this chapter, two things are pointed out. Firstly, the performance of an RSOD model trained on real-world data of template-based VSMs will be demonstrated. Secondly, it is shown that it is fundamentally possible to have an RSOD model learn key characteristics of hand drawings using synthetically generated data. The aim is to pave the way towards an RSOD model for completely hand-drawn VSM recognition.

4.1. Model training for template-based VSM recognition.

To evaluate the potential of RSOD in the context of template-based VSM recognition, a model is trained on a real-world dataset of 229 photos of VSMs in which each VSM can be present up to 5 times photographed under different environmental conditions. The dataset shows annotated examples of 1299 arrows as well as 7217 template-based value stream symbols. During training, the standard augmentations scaling, rotating, shearing, blurring, adding of noise, translating, perspective distortion, and varying hue, saturation, and color value are used [25]. The framework used for training the RSOD model is based on a YoloV5 architecture and allows the use of rotated bounding boxes [15].

The model performance is evaluated using the Mean Average Precision metric (mAP) for object detection [26]. The metric values are calculated for an Intersection over Union (IoU) threshold of 0.5 and as an average of results yielded for an IoU threshold range from 0.5 to 0.95 following the benchmarking principles introduced with large-scale object detection benchmark dataset COCO, which are standard in the field of object detection [26]. The IoU threshold is a criterion used to determine whether a predicted bounding box in object detection accurately matches a ground truth box [27].

The dataset is split into 183 samples for training, 23 for validation, and 23 for testing. It is ensured that no VSM is present in multiple subsets. Table 1 shows the yielded mAP values for symbols and arrows on the unseen test subset.

Table 1: Yielded mAPs for arrows and symbols.

Test dataset performance	mAP 0.5	mAP 0.5-0.95
Arrows and Symbols	0.82	0.63
Symbols	0.84	0.65
Arrows	0.54	0.30

The performance when recognizing arrows is, as expected, significantly lower than when recognizing template-based symbols. Figure 5 depicts the results of an inference run for a commonly used IoU threshold of 0.5 [28]. The example given shows the typical inference behavior of the trained model. While all the symbols are flawlessly detected, only one arrow is detected correctly. It can be assumed that the dataset provided for training does not cover a sufficiently wide range of variance in arrow appearance to force the model to learn the necessary generalized features describing arrows.

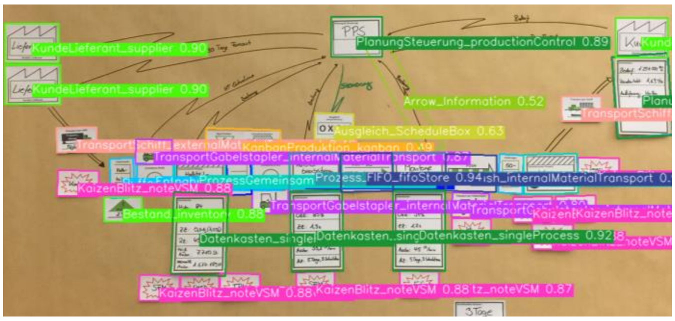

Fig. 5: Inference result from model trained on real-world data.

4.2. Learning key characteristics from synthetic data.

To evaluate the potential of generators for hand drawings in training data generation, a model is trained on the real-world dataset enriched with an additional 3583 generated arrows arranged on blank backgrounds. For comparison, a dataset exclusively containing real-world data and a dataset exclusively containing generated arrows are used to train additional models under the same settings.

To validate that it is generally possible to train an ML model to recognize real-world arrows using this data, a real-world test dataset with 200 hand-drawn arrows is used. The arrows in this dataset are based on the appearance that the generator can generate which leaks the typical strong curvature of arrows in real-world VSMs. To demonstrate the general applicability of the presented approach, the conditions in the experiment conducted are idealized. Figure 6 depicts a labeled example data from this real-world test dataset.

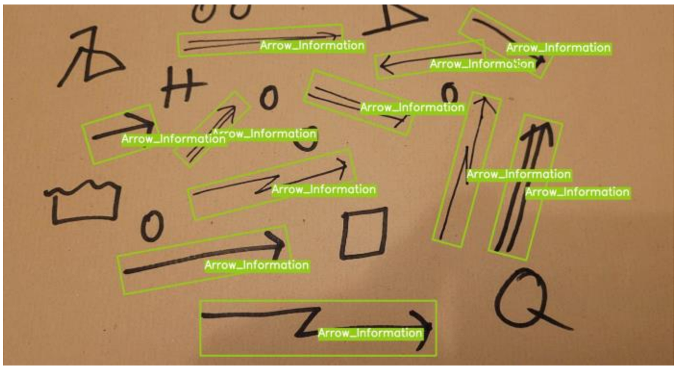

Fig. 6: Example data from the real-world test dataset.

Table 2 lists the mAPs yielded on the real-world test dataset for different underlying train dataset configurations.

Table 2: Yielded mAPs for different train dataset configurations.

Training dataset configuration	mAP 0.5	mAP 0.5-0.95
Synthetic data exclusively	0.10	0.01
Real-world data exclusively	0.17	0.06
Real-world and synthetic data combined	0.63	0.17

The results indicate that the synthetic data has enabled the model to learn key characteristics of the real-world arrows in a specific hand-drawn style as the model trained using a combination of real-world and synthetic data yields the highest mAP values. At the same time, the exclusive use of synthetic data seems not to be promising. While the model trained on synthetic data archived over 0.98 mAP (0.5 IoU thres.) on unseen synthetic data in additional trails, the domain shift towards real-world data seems to be a performance inhibitor as the performance when testing on real-world data drops to an mAP of 0.1 (0.5 IoU thres.).

5. Conclusion

Simulation software can help to better understand complex relationships and the dynamic behavior of value streams. The necessary digitization of the VSMs, which are typically created in analog form using pen-and-paper, stands in the way of a seamless workflow when using such software and creates a demand for an automated VSM digitization solution.

The presented RSOD approach shows useful performance in the recognition of template-based symbols. In combination with state-of-the-art OCR solutions, these recognized symbols can serve as a possible basis for the recognition of the overall semantics and parametrization of VSMs. At the same time, the RSOD model trained on real-world data of VSMs shows significant weaknesses in the recognition of arrows. Arrows are essential key elements that must be considered when deriving the semantics of a VSM. Accordingly, a significant improvement in the performance of arrow recognition is crucial in this context. The manual creation and annotation of arrows is severely time-consuming. In this regard, it has been shown that is generally possible to use generated arrows as an alternative to real-world data for ML model performance increase.

6. Outlook and future work

As described, the general appearance of arrows the generator can produce is not close enough to the arrows typically seen in VSMs. Major adjustments regarding the generator must be made so that its usability in a non-idealized setting is granted. While it is still possible to define arrows and the associated variation ranges for derivation calculation in just a few lines of code, this procedure becomes significantly unpractical with somewhat more complex symbols such as the kaizen flash or the forklift symbol. One possible solution could be in the form of a graphical user interface in which symbols and corresponding variation ranges are defined graphically and not in the form of code.

As shown, the performance of ML models is likely to suffer from domain shift. Synthetic data might show properties that allow a clear conclusion to be drawn about an existing symbol, but which do not occur in the real-world data. This can lead to an exploitation of the Clever Hans effect during training [29]. Addressing domain shifts and improving robustness with optimized augmentation methods and other approaches suitable to mitigate the effects of domain shifts in image recognition is currently subject to intense research [30, 31]. Future works may concern such approaches addressing the problem of domain shift.

References

[1] T. Ohno and N. Bodek, *Toyota Production System*: Productivity Press, 2019.

[2] M. Salwin, I. Jacyna-Gołda, M. Bańka, D. Varanchuk, and A. Gavina, "Using Value Stream Mapping to Eliminate Waste: A Case Study of a Steel Pipe Manufacturer," *Energies*, vol. 14, no. 12, p. 3527, 2021, doi: 10.3390/en14123527.

[3] S. Samant and R. Prakash, "Achieving lean through value stream mapping with constraint programming and simulation technique for complex production systems," *IJISE*, vol. 37, no. 1, p. 119, 2021, doi: 10.1504/IJISE.2021.112470.

[4] T. Meudt, J. Kaiser, J. Metternich, and S. Spieckermann, "Wertstrommodellierung und -simulation im Zeichen von Digitalisierung und Industrie 4.0," *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, vol. 112, no. 12, pp. 865–868, 2017, doi: 10.3139/104.111846.

[5] Rene Berndt, Nelson Silva, Christian Caldera, Ulrich Krispel, and Dieter W. Fellner, *VASCO - Digging the Dead Man's Chest of Value Streams*, 2016. [Online]. Available: https://www.researchgate.net/profile/gerhard-reisinger/publication/313057972_vasco_-_digging_the_dead_man's_chest_of_value_streams

[6] A. Ebrahimi, R. Khakpour, and S. Saghiri, "Sustainable setup stream mapping (3SM): a systematic approach to lean sustainable manufacturing," *Production Planning & Control*, vol. 34, no. 4, pp. 311–329, 2023, doi: 10.1080/09537287.2021.1916637.

[7] B. Rathore, A. K. Pundir, R. Iqbal, and R. Gupta, "Development of fuzzy based ergonomic-value stream mapping (E-VSM) tool: a case study in Indian glass artware industry," *Production Planning & Control*, vol. 34, no. 16, pp. 1618–1638, 2023, doi: 10.1080/09537287.2022.2035447.

[8] C. Urnauer and J. Metternich, "Die digitale Wertstrommethode," *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, vol. 114, no. 12, pp. 855–858, 2019, doi: 10.3139/104.112203.

[9] Dirk Wortmann, "Digitales Wertstrommanagement - Wertstrom mobil simuliert und analysiert," *IT & Production*, no. 20, pp. 64–65, 2019.

[10] A. Goienetxea Uriarte, A. H.C. Ng, and M. Urenda Moris, "Bringing together Lean and simulation: a comprehensive review," *International Journal of Production Research*, vol. 58, no. 1, pp. 87–117, 2020, doi: 10.1080/00207543.2019.1643512.

[11] M. Rother and J. Shook, *Learning to See: Value Stream Mapping to Add Value and Eliminate Muda*: Lean Enterprise Institute, 2003.

[12] T. Sinha, N. Ward, Christine Welch, and Briony Boydell, "Value stream mapping in a service environment: a comparison of approaches," *17th Annual EuROMA Conference: Managing Operations in Service Economies*, 2010. [Online]. Available: https://researchportal.port.ac.uk/en/publications/value-stream-mapping-in-a-service-environment-a-comparison-of-app

[13] N. O.' Mahony *et al.,* "Deep Learning vs. Traditional Computer Vision," *in Advances in Computer Vision Proceedings of the*, doi: 10.1007/978-3-030-17795-9.

[14] Y. Pu *et al.,* "Adaptive Rotated Convolution for Rotated Object Detection," 2023. [Online]. Available: http://arxiv.org/pdf/2303.07820

[15] Kelly M, *YOLOv5 for Oriented Object Detection*. Roboflow Blog. [Online]. Available: https://blog.roboflow.com/yolov5-for-oriented-object-detection/

[16] Google, *Google Cloud Vision API Documentation.* [Online]. Available: https://cloud.google.com/vision/docs/handwriting?hl=de (accessed: May 2024).

[17] Cem Dilmegani, "OCR in 2024: Benchmarking Text Extraction/Capture Accuracy," in *AIMultiple.* Accessed: 04.2024. [Online]. Available: https://research.aimultiple.com/ocr-accuracy/

[18] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2016.352

[19] Ahmet Onur GİRAY and Hatice DOĞAN, "Lane Detection for Autonomous Driving in a Video Game Environment,", *Jitsa*, vol. 6, no. 2, pp. 209–222, 2023, doi: 10.51513/jitsa.1200774.

[20] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A LiDAR Point Cloud Generator," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, New York, NY, USA, 2018. [Online]. Available: http://dx.doi.org/10.1145/3206025.3206080

[21] J. S. Leinar Ramos, *Gartner, Maverick Research: Maverick Research: Forget About Your Real Data — Synthetic Data Is the Future of AI.* [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2022-06-22-is-synthetic-data-the-future-of-ai

[22] S. I. Nikolenko, *Synthetic Data for Deep Learning*. Cham: Springer International Publishing, 2021. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-75178-4

[23] D. Karunakaran, J. S. Berrio Perez, and S. Worrall, "Generating Edge Cases for Testing Autonomous Vehicles Using Real-World Data," *Sensors (Basel, Switzerland)*, vol. 24, no. 1, 2023, doi: 10.3390/s24010108.

[24] "Curve Fitting with Bézier Cubics," *Graphical Models and Image Processing*, vol. 58, no. 3, pp. 223–232, 1996, doi: 10.1006/gmip.1996.0019.

[25] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning,", *J Big Data*, vol. 6, no. 1, pp. 1–48, 2019, doi: 10.1186/s40537-019-0197-0.

[26] *Coco Dataset Documentation: Detection Evaluation.* [Online]. Available: https://cocodataset.org/

[27] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge,", *Int J Comput Vis*, vol. 88, no. 2, pp. 303–338, 2010, doi: 10.1007/s11263-009-0275-4.

[28] "IoU-uniform R-CNN: Breaking through the limitations of RPN," *Pattern Recognition*, vol. 112, p. 107816, 2021, doi: 10.1016/j.patcog.2021.107816.

[29] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, "Unmasking Clever Hans predictors and assessing what machines really learn,", *Nat Commun*, vol. 10, no. 1, pp. 1–8, 2019, doi: 10.1038/s41467-019-08987-4.

[30] P. Oza, V. A. Sindagi, V. V. Sharmini, and V. M. Patel, "Unsupervised Domain Adaptation of Object Detectors: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–24, 2024, doi: 10.1109/tpami.2022.3217046.

[31] "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *0031-3203*, vol. 137, p. 109347, 2023, doi: 10.1016/j.patcog.2023.109347.