Herausgeber

T. LÄNGLE
M. HEIZMANN

FORUM BILDVERARBEITUNG 2024
IMAGE PROCESSING FORUM 2024

KIT Scientific Publishing

T. Längle | M. Heizmann (Hrsg.)

FORUM BILDVERARBEITUNG 2024
IMAGE PROCESSING FORUM 2024

FORUM BILDVERARBEITUNG 2024
IMAGE PROCESSING FORUM 2024

Herausgegeben von
T. Längle und M. Heizmann

KIT Scientific Publishing

Impressum

Scientific Publishing

Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu

Print on Demand 2024 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 2510-7224
ISBN 978-3-7315-1386-5
DOI 10.5445/KSP/1000174496

Vorwort

Bildverarbeitung spielt in vielen Bereichen der Technik und des Alltags eine wichtige Rolle zur Informationserfassung. Sie ist eine etablierte Technologie u. a. in der Mess- und Automatisierungstechnik, der Robotik, der Fahrzeugtechnik und der Unterhaltungselektronik. Wichtige Vorteile von Bildverarbeitungssystemen gegenüber anderen sensorischen Prinzipien bestehen u. a. darin, dass Bilder berührungslos gewonnen werden können und dass Bildsensoren inzwischen vergleichsweise günstig sind. Besonders spannend an der Bildverarbeitung ist aber, dass die Sensorik weitgehend dem menschlichen Leitsinn – dem Sehen – entspricht, aber nicht an die Beschränkungen des Menschen gebunden ist. Dies betrifft etwa die nutzbaren Spektralbereiche, die quantitative Interpretierbarkeit der Bilder, die gleichbleibende Aufmerksamkeit und Reproduzierbarkeit oder die Möglichkeit zur Erfassung hochdynamischer Prozesse. Auch wenn die Bildverarbeitung als Teil mehrerer Fachdisziplinen – u. a. Mess- und Automatisierungstechnik, Systemtheorie, Mathematik, Informatik, Optik, Lichttechnik, Mikrosystemtechnik – eine gewisse Reife erreicht hat, gibt es immer noch spannende neue Erkenntnisse. Gerade die Bildverarbeitung profitiert enorm von neuen Technologien wie z. B. dem maschinellen Lernen oder neuen Sensorprinzipien wie etwa Event-Based Vision.

Das „Forum Bildverarbeitung / Image Processing Forum“ hat das Ziel, über solche aktuellen Trends und neuartige Lösungen in der Bildverarbeitung zu berichten und zum fachlichen Austausch zwischen Wissenschaft und Anwendung beizutragen. Es findet in jedem zweiten Jahr seit 2010 statt und wird inzwischen gemeinsam vom Geschäftsfeld Inspektion und Optronische Systeme des Fraunhofer-Instituts für Optronik, Systemtechnik und Bildverarbeitung IOSB und dem Institut für Industrielle Informationstechnik IIIT des Karlsruher Instituts für Technologie KIT organisiert. Dem Aufruf zur Einreichung von Beiträgen sind erfreulich viele Autoren gefolgt. Aus den Einreichungen konnte der Programmausschuss nach einer eingehenden Begutachtung 24 hochwertige Beiträge auswählen und den Schwerpunkten

- Messtechnische Anwendungen,
- Robotik,
- Bildgewinnung,
- Bildverarbeitung,
- Unsicherheiten bei maschinellem Lernen,
- Wahrnehmung von Personen,
- Künstliche Intelligenz als Mess- und Prüfmittel, sowie
- Fahrzeuge

zuordnen. Zur überwiegenden Zahl der Beiträge wurden Aufsätze erstellt, die im vorliegenden Tagungsband enthalten sind. Wir danken den Autoren für ihre Beiträge, den Mitgliedern des Programmausschusses für die Ansprache von Autoren und ihre wertvolle Expertise bei der Begutachtung der Einreichungen und allen, die durch ihre Anwesenheit zum Gelingen des „Forums Bildverarbeitung / Image Processing Forum“ beitragen. Für die Organisation der Veranstaltung und die technische Unterstützung bei der Erstellung des Tagungsbands bedanken wir uns bei Britta Ost, Felix Lehnerer, Florian Steigleder, Lukas Dippon und Jürgen Hock.

November 2024

Thomas Längle
Michael Heizmann

Tagungsleitung

Prof. Dr.-Ing. M. Heizmann	Karlsruher Institut für Technologie
Prof. Dr.-Ing. T. Längle	Fraunhofer IOSB Karlsruhe

Mitglieder

Prof. Dr. C. Bach	Buchs (Schweiz)
Dr.-Ing. S. Bauer	Boston (Massachusetts, USA)
Dr.-Ing. D. Berndt	Magdeburg
Prof. Dr.-Ing. J. Beyerer	Karlsruhe
Prof. Dr. A. Braun	Düsseldorf
Dr. rer. nat. J. Eggert	Offenbach
Dr. M. Glitzner	München
Dr. T. Haist	Stuttgart
Prof. Dr.-Ing. M. Huber	Stuttgart
Prof. Dr. B. Jähne	Heidelberg
M. Sc. C. Kludt	Karlsruhe
Dipl.-Ing. M. Maurer	Wiesbaden
Dr. J. Meyer	Karlsruhe
Dr. M. Overdick	Waldkirch
Prof. Dr. F. Salazar	Madrid (Spanien)
Dipl.-Ing. M. Stelzl	Mainz
Prof. Dr. R. Stiefelhagen	Karlsruhe
Prof. Dr.-Ing. C. Stiller	Karlsruhe
Prof. Dr.-Ing. R. Tutsch	Braunschweig
Prof. Dr.-Ing. M. Ulrich	Karlsruhe
Prof. Dr.-Ing. S. Werling	Mannheim
Prof. Dr.-Ing. V. Willert	Darmstadt

Herausgeber

T. Längle und M. Heizmann

Inhaltsverzeichnis

Bildgewinnung

Bildverarbeitung

Unsicherheiten bei maschinellem Lernen

Wahrnehmung von Personen

Fahrzeuge

Automated image-based parameter optimization for single-pulse laser drilling

Manuel Klaiber[1,2,3], Mathias Hug[1,2,3], Lukas Schneller[1,3], Ömer Can[1], Andreas Jahn[2], Axel Fehrenbacher[2], Peter Reimann[1,4], and Andreas Michalowski[3]

[1] Graduate School of Excellence advanced Manufacturing Engineering (GSaME), University of Stuttgart, Nobelstraße 12, 70569 Stuttgart
[2] TRUMPF Laser SE, Aichhalder Straße 39, 78713 Schramberg
[3] Institut für Strahlwerkzeuge (IFSW), University of Stuttgart, Pfaffenwaldring 43, 70569 Stuttgart
[4] Institute of Parallel and Distributed Systems (IPVS), University of Stuttgart, Universitätsstraße 28, 70569 Stuttgart

Abstract A significant challenge in laser drilling is the optimization of process parameters and drilling strategies to achieve high-quality holes. This is further complicated by the fact that quality assessment is a manual and time-consuming task. This paper presents a methodology designed to significantly reduce the manual effort required in optimizing parameters for single-pulse laser drilling of 0.3 mm thick stainless steel. The objective is to precisely drill holes with an entry diameter of 70 µm and an exit diameter of 20 µm, achieving high roundness. The features of the drilled holes were extracted automatically from the raw data using a combined approach that utilizes deep learning and image processing techniques. The outcomes were compared against manual measurements. Results indicate that the mean deviations between automated and manual measurements for both inlet and outlet diameters are less than one micrometer. We employed a Bayesian optimization algorithm to efficiently explore the parameter space without the need for incorporating expert knowledge. The approach rapidly identified optimal drilling parameters after only a few iterations, significantly expediting the optimization process and considerably reducing manual labor.

Keywords Laser drilling, semantic segmentation, feature extraction, Bayesian optimization

1 Introduction

The manufacturing industry is constantly searching for advanced methods to improve the precision and efficiency of laser drilling processes [1]. Various strategies have been employed, including traditional methods such as Design of Experiments (DoE) and Response Surface Methodology (RSM), as well as advanced computational techniques. For example, Gupta et al. [2] used DoE and RSM to optimize the hole quality in ms-pulsed laser drilling, while Wang et al. [3] applied artificial neural networks to predict optimal drilling parameters for ns-pulsed laser drilling in stainless steel. Chatterjee et al. [4] used neuro-fuzzy systems and genetic programming to predict drilling outcomes, showing reasonable accuracy. However, these strategies often require extensive experimental setups or training data and cannot efficiently navigate complex parameter spaces to search for optimal drilling parameters.

Recent advances in computational techniques, particularly approaches to Bayesian optimization (BO), provide a promising alternative that can predict multi-dimensional parameters spaces in laser processes with significantly fewer iterations and less manual intervention [5]. Yang et al. [6] applied BO to improve taper and drilling time in spiral drilling of stainless steel, achieving suitable results with few iterations. Bamoto et al. [7] optimized a femtosecond laser micro-drilling process and Menold et al. [8] demonstrated the versatility of BO in optimizing laser cutting, laser welding and laser polishing and showed that less experiments are needed than with traditional approaches.

In addition to the actual optimization of drilling parameters, the extraction of the features required by the optimization approaches represents a significant challenge in process optimization. In previous studies on laser drilling, the quality measurements were predominantly assessed through manual measurements [2]. Feuer et al. [9] propose an automated approach to extract the drilling geometry as features. Approaches to automated feature extraction and quality control for a laser welding process using semantic segmentation are presented by Hartung et al. [10].

This paper presents an approach that incorporates sophisticated feature extraction techniques that employ a combination of deep learning models and conventional image processing methods to accurately ex-

tract quality features of single-pulse drilled holes. Subsequently, this study investigates the potential of BO with the aim of determining optimal laser parameters including pulse power, pulse length, and focus position to ensure high-quality holes in terms of diameter and roundness.

2 Materials and Methods

This section describes the experimental setup for single-pulse laser drilling of thin metal sheets. Furthermore, it gives an overview of the feature extraction and parameter optimization methods utilized. The procedure of the iterative optimization process is shown in Figure 1. For the first n=6 optimization steps, the parameter sets are generated using a sobol sequence to ensure that the points are evenly distributed in the parameter space. Subsequent parameter sets are suggested by the BO.

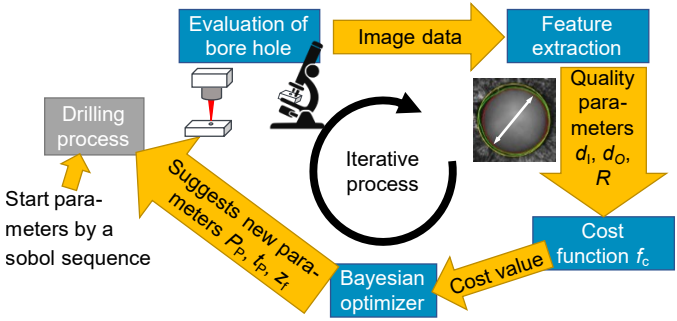


Figure 1: Optimization process with a Bayesian Optimizer.

2.1 Experimental Setup

Figure 2 shows the experimental setup of the single-pulse laser drilling process. In this study, a continuous wave (cw) single mode fiber laser (TRUMPF TruFiber 2000) was used to perform the single-pulse laser drilling experiments. The emission wavelength of the unpolarized laser

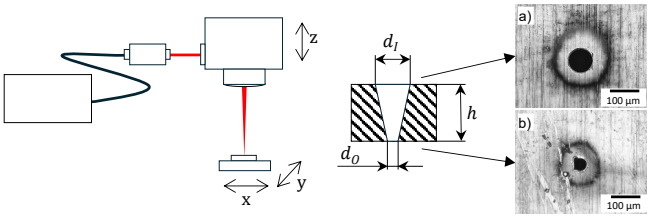


Figure 2: Left: Experimental Setup. Right: Borehole cross section with images of an a) inlet with diameter d_I and b) outlet with diameter d_O.

was specified as 1075 nm in conjunction with a beam propagation factor of M^2<1.2. The laser beam was positioned onto the stainless steel sample with a galvanometer scanner. A telecentric F-Theta lens with a focal length of 163 mm was used, resulting in a focus diameter ($1/e^2$) of d_f= 20 µm. The pulsed operation mode of the laser source enables the generation of pulses with a peak power P_P up to 1400 W. This allows for the adjustment of the pulse duration between values from 1 to 25 µs. The setup was equipped with linear stages (x,y) for the sample and a linear drive (z) for the process optics to adjust the focus position. The focus position can thus be positioned with an accuracy of one micrometer.

The materials used for the experiments are stainless steel (1.4310) substrates. The substrates, with a thickness h of 0.3 mm, were cut to a size of 100 mm × 50 mm. An optical microscope (Zeiss Axio Imager) was used to evaluate the borehole criteria, such as inlet (Figure 2 a) and outlet (Figure 2 b). A 20× magnification was used for optical microscopic observation, where one pixel is equivalent to 0.172 × 0.172 µm^2. The evaluation criteria include the diameter of the inlet d_I and of the outlet d_O, as well as the roundness R of the outlet. We drilled and analyzed i=3 holes per parameter set to reduce the influence of side effects from the inherent noise of laser processing and other uncertainties.

2.2 Feature Extraction

The objective is to automatically extract the features that are required for the parameter optimization directly from the microscope images. The features include the borehole's inlet diameter d_I and outlet diame-

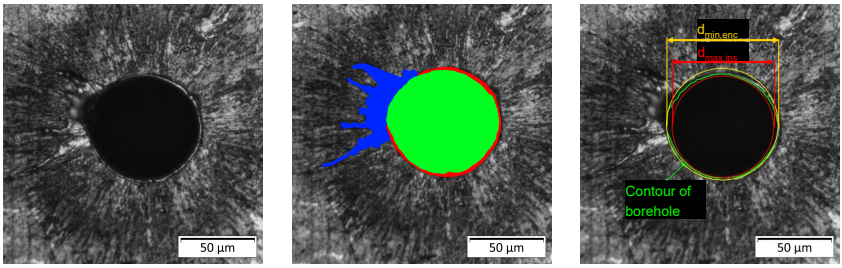


Figure 3: (a) Inlet of a borehole with breakthrough. (b) Segmented classes by the neural network: melt (blue), burr (red) and borehole with breakthrough (green). (c) Contour of the borehole (green) from which the diameter of the maximum inscribing circle $d_{max,ins}$ (red) and the diameter of the minimum enclosing circle $d_{min,enc}$ (yellow) were derived.

ter d_O, the borehole's roundness R, the area of the melt deposits around the borehole, and a classification of whether a breakthrough has occurred. Initial attempts to perform feature extraction based solely on conventional image processing methods have not delivered satisfactory results. Due to the divergent surface properties of the materials to be processed, there is a high degree of variance in the captured images, e.g., due to reflections and mirroring. This variance requires great efforts to manually adjust the algorithm parameters of conventional image processing methods. Deep learning methods represent another viable approach to address natural deviations in images like reflections and mirroring. Nevertheless, a method based exclusively on deep learning that directly determines quality characteristics is intricate and challenging for the operator to comprehend. A combined approach, comprising semantic segmentation models and conventional image processing methods, enables a more robust and understandable extraction of features. In our study, we employ two semantic segmentation models, each with a neural network architecture modified from the SDU-net [11]. These models are used to segment images from the top (inlet) and bottom (outlet) of the borehole. The inlet model classifies the image into the following classes, as partly shown in Figure 3(b): *burr*, *melt*, and one of the classes *borehole with breakthrough* or *borehole*

without breakthrough. The outlet model segments the image into: *background*, *melt*, *borehole with breakthrough*, and *borehole without breakthrough*. To train the inlet model, 68 labeled images were used, while the outlet model was trained with 44 images. The discrepancy in the number of training images is due to the fact that only continuous boreholes are included in the outlet dataset. The models are initialized randomly without any pre-training. Both models use *Categorical Focal Loss* [12] as loss function. The classes segmented by the models are further analyzed using conventional image processing methods. Figure 3(c) shows, how the borehole diameter d_I was calculated using the contour (green) of the segmented borehole class *borehole with breakthrough*. This calculation involves averaging the diameters of two specific circles: the minimum enclosing circle, $d_\mathrm{min,enc}$ (shown in yellow), determined using the method proposed by Welzl et al. [13], and the maximum inscribing circle, $d_\mathrm{max,ins}$ (shown in red), as described by Xia et al. [14].

The roundness R of the borehole is defined by the ratio of the borehole area A_borehole (Figure 3(b) green) to the area of the minimum enclosing circle $A_\mathrm{min,enc}$ [15]. The melt deposition area is calculated as the sum of the segmented burr and melt area classes. In order to ascertain whether breakthrough is present, the areas belonging to the *borehole with breakthrough* and *borehole without breakthrough* classes are compared. The classification of breakthrough is dependent on the class from which the larger area was segmented.

2.3 Bayesian Optimization

To optimize the single-pulse laser drilling process, we used the extracted data in a Bayesian Optimization framework. The goal was to find laser parameters that yield high-quality holes with defined geometries. The AX Service API [16] was used, with a Gaussian Process as a surrogate model [17], and the default squared exponential kernel for the optimization. This approach efficiently explored the parameter space, aiming to optimize the drilling process with minimal experimental effort. As acquisition function *Expected Improvement* [18] was used. More detailed explanations and applications of the BO for other laser processes were given by Menold et al. [8]. Table 1 shows the process parameters that were varied and the quality parameters that result from the feature extraction process described in Section 2.2. The area

of the melt was excluded from the BO to concentrate on enhancing the accuracy of the diameters and roundness.

Table 1: Parameters and variables for the process.

Category	Parameter/Variable	Symbol	Value Range, Target
Process Parameters	Pulse Power	P_P	300 W ... 1400 W
	Pulse Length	t_P	1 µs ... 25 µs
	Focal Position	z_f	-200 µm ... 200 µm
Quality Variables	Inlet Diameter	d_I	$d_\mathrm{I,target} = 70\,\mu\mathrm{m}$
	Outlet Diameter	d_O	$d_\mathrm{O,target} = 20\,\mu\mathrm{m}$
	Roundness	R	0 ... 1, R_target=1

For each parameter set, i=3 holes were drilled and evaluated with an optical microscope. The image data was analyzed by feature extraction to obtain the inlet and outlet diameters $d_\mathrm{I}, d_\mathrm{O}$ and the roundness R of the outlet. The cost function

$$\begin{aligned} C(x) = {} & w_\mathrm{d,I} \cdot |d_\mathrm{I}(x) - d_\mathrm{I,target}| + w_\mathrm{d,O} \cdot (d_\mathrm{O}(x) - d_\mathrm{O,target})^2 \\ & + w_\mathrm{R} \cdot (1 - R(x)) + w_\mathrm{E} \cdot E_\mathrm{P} \end{aligned} \tag{1}$$

with the process parameters x=($P_\mathrm{P}, t_\mathrm{P}, z_\mathrm{f}$) and the weights $w_\mathrm{d,I}$=1 µm^{-1}; $w_\mathrm{d,O}$=4 µm^{-1}; w_R=200; w_E=2 mJ^{-1} calculates the cost C of each borehole, with lower costs indicating higher quality. Determining the appropriate weights w requires domain-specific expertise and is inherently subjective. These weightings are contingent upon the optimization objectives and the relative magnitude of the associated process parameters. Given the significant impact of these weightings on the optimization outcomes, it may be necessary to adjust them prior to initiating the optimization process. $C(x)$ is formulated to achieve a target inlet and outlet diameter with maximum roundness of the outlet. Pulse length and pulse power were summarized as pulse energy E_P=$P_\mathrm{P} \cdot t_\mathrm{P}$, which is to be minimized to encourage a short drilling duration and lower heat input. If no breakthrough occurs, the cost C becomes high due to the quadratic influence of the outlet diameter term $w_\mathrm{d,O} \cdot (d_\mathrm{O}(x) - d_\mathrm{O,target})^2$. In addition, the roundness R is set to zero, which leads to maximum costs of the roundness term $w_\mathrm{R} \cdot (1 - R(x))$=200.

3 Results and Discussion

This section discusses the results obtained from the feature extraction techniques, which are divided into two parts: The evaluation of the training of the segmentation networks and the evaluation of the feature extraction methods based on the segmentation results. Subsequently, we explore the findings from the BO process.

3.1 Results of the Feature Extraction

To evaluate the effectiveness of the feature extraction process, 80 images of borehole openings, captured from 40 laser-drilled boreholes (40 images of inlets and 40 images of outlets), have been labeled by experts and are available for analysis. These images were not included in the training data set. We employ the Intersection over Union (IoU) [19] as evaluation metric to assess the model predictions:

$$\mathrm{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

Where A is the segmentation mask used for training and B is the prediction of the segmentation network. During the evaluation, the inlet model achieved an IoU value of 0.97 for the borehole classes, while the outlet model achieved an IoU value of 0.95 for this class. However, the melt and burr classes exhibit a decline in performance, with each reaching an IoU value of 0.75. This is primarily attributable to the distinctive characteristics of the melt, which also manifests as a maximum IoU value of 0.78 for these two classes during training.

After the image segmentation, the diameters are calculated based on the prediction of the borehole models. To assess the precision of the measurement techniques with respect to representative data, the inlets and outlets of 40 additional boreholes, drilled in identical experimental conditions as illustrated in Figure 2, were evaluated. Figure 4 shows the diameters based on automatic feature extraction $d_{\text{feat. extr.}}$ (x-axis) and manual measurements $d_{\text{man.}}$ (y-axis) of the inlet (blue) and outlet (green). The manual measurements were conducted using an optical microscope. The right side of Figure 4 shows the deviation between the feature extraction diameter and the manual measurement. The mean deviation is -0.5 µm for the inlet and 0.34 µm for the outlet,

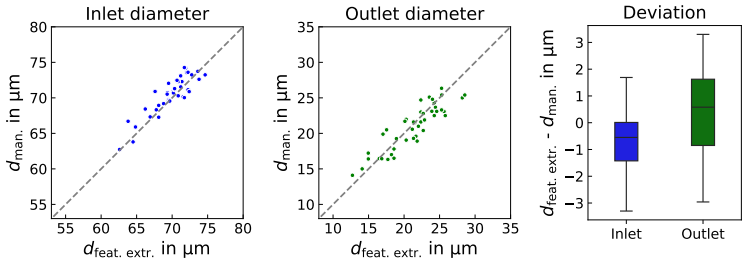


Figure 4: Comparison of the results of automatic feature extraction (x-axis) with manual measurements (y-axis) of inlet and outlet diameters.

which is within the expected accuracy and tolerance limits for borehole measurements. These low deviations, typical between automated and manual measurement techniques, validate the effectiveness of feature extraction in determining inlet and outlet diameters. The methods outlined enable automated borehole measurement, facilitating the use of the extracted features for parameter optimization and significantly reducing manual effort.

3.2 Results of the Bayesian Optimization

Figure 5 shows the evolution of the process parameters (left) and quality variables (right) during the optimization process. During the initialization process (orange) with parameters chosen by the sobol sequence, a wide range of process parameters is covered, resulting in a high cost value (red curve). In the start sequence, three parameter sets n=3, 4, 5 did not lead to through holes, because the pulse duration was too short. In the following optimization steps the BO suggested only one more parameter set at n=26, where no breakthroughs were achieved. In the bottom right of Figure 5 the inlet and outlet of the borehole with the minimum cost value C_{12}=$86.00^{+13.57}_{-20.83}$ after n=12 iterations with process parameters z_f=-79.0 µm, P_L=898 W and t_P=17 µs is shown. This led to an inlet diameter of d_I=$68.0^{+0.5}_{-0.8}$ µm, an outlet diameter of d_O=$18.03^{+0.57}_{-0.43}$ µm and a roundness of R=$0.84^{+0.6}_{-0.4}$ µm which are close to the targeted values.

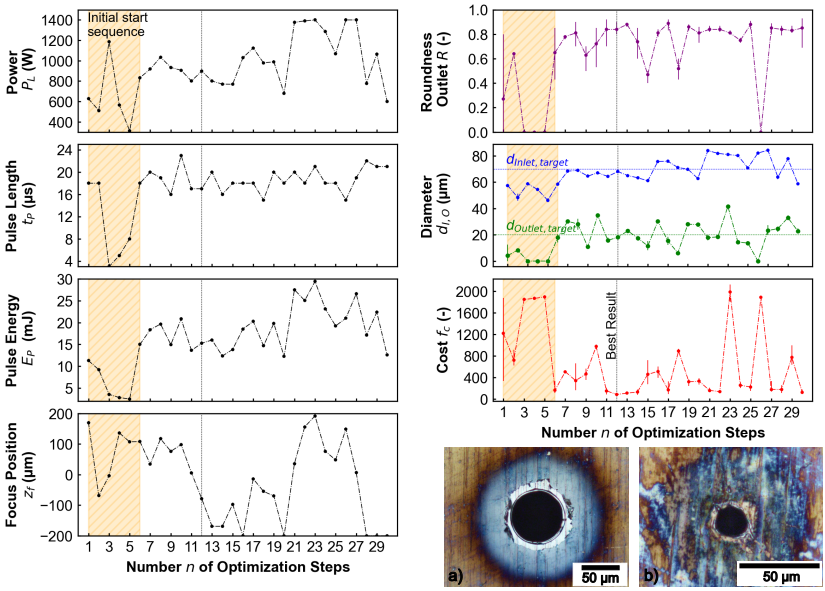


Figure 5: Evolution of the process parameters (left) and quality variables (right) of the drilled holes and the value of the cost function during optimization (red). The error bars are min/max values of three experiments for each parameter set. a) inlet and b) outlet of borehole n=12.

4 Conclusion

The aim of this work was to reduce the manual effort in parameter search for single-pulse laser drilling. By employing a combination of deep learning techniques for the segmentation of microscope images and conventional image processing methods for the measurement of segmentations, it is possible to perform a robust and rapid determination of the quality features of a borehole, particularly in challenging imaging situations, such as those caused by reflections. The results demonstrate that the mean deviations between manual measurements and feature extraction for both inlet and outlet diameters are less than one micrometer. Furthermore, BO has been demonstrated to be an effective approach for achieving target hole characteristics with a min-

imal number of iterations. In an optimization experiment comprising 30 iterations, the parameters conducive to drilling with the desired characteristics were identified after just 12 iterations. This significantly reduces the need for traditional full-factorial experimental designs, simplifying the laser drilling optimization process and increasing efficiency in industrial applications.

References

1. Y. C. Shin, B. Wu, S. Lei, G. J. Cheng, and Y. Lawrence Yao, "Overview of Laser Applications in Manufacturing and Materials Processing in Recent Years," *Journal of Manufacturing Science and Engineering*, vol. 142, no. 11, p. 110818, 10 2020.
2. A. K. Gupta, R. Singh, and D. Marla, "Millisecond pulsed laser microdrilling of stainless steel – optimizing hole quality using response surface methodology," *Journal of Laser Micro/Nanoengineering*, 2023.
3. C.-S. Wang, Y.-H. Hsiao, H.-Y. Chang, and Y.-J. Chang, "Process parameter prediction and modeling of laser percussion drilling by artificial neural networks," *Micromachines*, vol. 13, no. 4, 2022.
4. S. Chatterjee, S. S. Mahapatra, V. Bharadwaj, B. N. Upadhyay, and K. S. Bindra, "Prediction of quality characteristics of laser drilled holes using artificial intelligence techniques," *Engineering with Computers*, vol. 37, no. 2, pp. 1181–1204, 2021.
5. A. Michalowski, A. Ilin, A. Kroschel, S. Karg, P. Stritt, A. Dais, S. Becker, G. Kunz, S. Sonntag, M. Lustfeld, P. Tighineanu, V. Onuseit, M. Haas, T. Graf, and H. Ridderbusch, "Advanced laser processing and its optimization with machine learning," 03 2023, p. 4.
6. J. Yang, J. Niu, L. Chen, K. Cao, T. Jia, and H. Xu, "Tunable simultaneous Bayesian optimization of hole taper and processing time in qcw laser drilling," *Journal of Manufacturing Processes*, vol. 109, pp. 471–480, 2024.
7. K. Bamoto, H. Sakurai, S. Tani, and Y. Kobayashi, "Autonomous parameter optimization for femtosecond laser micro-drilling," *Opt. Express*, vol. 30, no. 1, pp. 243–254, Jan 2022.
8. T. Menold, V. Onuseit, M. Buser, M. Haas, N. Bär, and A. Michalowski, "Laser material processing optimization using bayesian optimization: A generic tool," *Light: Advanced Manufacturing*, 2024.

9. A. Feuer, R. Weber, R. Feuer, D. Brinkmeier, and T. Graf, "High-quality percussion drilling with ultrashort laser pulses," *Applied Physics A*, vol. 127, no. 9, 2021.
10. J. Hartung, A. Jahn, and M. Heizmann, "Quality control of laser welds based on the weld surface and the weld profile."
11. S. Wang, S.-Y. Hu, E. Cheah, X. Wang, J. Wang, L. Chen, M. Baikpour, A. Ozturk, Q. Li, S.-H. Chou, C. D. Lehman, V. Kumar, and A. Samir, "U-net using stacked dilated convolutions for medical image segmentation."
12. M. Yeung, E. Sala, C.-B. Schönlieb, and L. Rundo, "Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation," *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society*, vol. 95, p. 102026, 2022.
13. E. Welzl, "Smallest enclosing disks (balls and ellipsoids)," in *New Results and New Trends in Computer Science*, ser. Lecture Notes in Computer Science, H. Maurer, Ed. Heidelberg: Springer-Verlag, 1991, vol. 555, pp. 359–370.
14. R. Xia, W. Liu, J. Zhao, H. Bian, and F. Xing, "Robust algorithm for detecting the maximum inscribed circle," in *Proc. of the 10th IEEE International Conference on Computer-Aided Design and Computer Graphics*. IEEE, 2007, pp. 230–233.
15. B. Walters, T. Uynuk-Ool, M. Rothdiener, J. Palm, M. L. Hart, J. P. Stegemann, and B. Rolauffs, "Engineering the geometrical shape of mesenchymal stromal cells through defined cyclic stretch regimens," *Scientific reports*, vol. 7, no. 1, p. 6640, 2017.
16. E. Bakshy, L. Dworkin, B. Karrer, K. Kashin, B. Letham, A. Murthy, and S. Singh, "Ae: A domain-agnostic platform for adaptive experimentation," Red Hook, NY, USA, 2018.
17. C. E. Rasmussen, *Gaussian processes for machine learning*, ser. Adaptive computation and machine learning. Cambridge, Mass.: MIT Press, 2006.
18. J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum," *Towards Global Optimization*, vol. 2, no. 117-129, p. 2, 1978.
19. P. Jaccard, "Lois de distribution florale dans la zone alpine," *Bulletin de la Societe Vaudoise des Sciences Naturelles*, vol. 38, no. 144, pp. 69–130, 1902.

Release 4.1 of the EMVA standard 1288: A universal concept to characterize modern image sensors

Bernd Jähne

Interdisciplinary Center for Scientific Computing (IWR)
Heidelberg University
Berliner Straße 43, 69120 Heidelberg

Abstract In order to illustrate the broad application range of the new version of the EMVA standard 1288, its basic concepts will be outlined and illustrated with measurements from standard monochrome industrial cameras, VIS-SWIR cameras and an automotive HDR camera with a dynamic range of 120 dB.

Keywords Image sensors, system theory, standardization, EMVA 1288

1 Introduction

The standard 1288 of the European Machine Vision Association (EMVA) is used worldwide for objective characterization of the quality parameters for industrial cameras [1–5]. It is the oldest standard activity of the EMVA, celebrating its twenties anniversary this year. The standard has been elaborated by a consortium of industry leading sensor and camera manufacturers, distributors, and research institutes.

A first version was published in 2005 [6], release 3.1 went into effect end of 2016 [7] with a standardized summary data sheet. This release still could only be applied to cameras with a linear characteristic curve. Furthermore, no preprocessing was possible which changes the temporal noise, except for simple operations such as binning or time-delayed-integration (TDI).

The next major progress was release 4.0 in 2021 [8], which added an additional general model to be applied to any camera with a defined

exposure time and known pixel size. At first glance it appears that the standard has now split into two variants. This is not the case, because still the same measurements are taken. Subsequent work on release 4.1 — which is still ongoing — made it clear that the addition of the general model put the focus on the parameter which is really important for signal quality, namely the signal-to-noise and the signal-to-nonuniformity. This is what a user should really should look at in first place and not parameters, such as the quantum efficiency and the standard deviation of the noise of the dark signal.

The paper takes this approach. In other words, it asks the simple question which effects limit the quality of the signal of an image sensor. These are

- the temporal noise, which expresses the uncertainty of each measurement,
- the nonuniformity, which says that each pixel of an image sensor responds to the exposure slightly different,
- the dark current, which represents the effect that there is a signal even without light, which is increasing with the exposure time, and
- the saturation capacity, which limits the maximum exposure that can be measured by an image sensor.

A more detailed presentation can be found in the textbook [9, Sec. 4.5].

2 A general system theoretical approach

A general system theoretical concept is the base of the standard 1288. which requires This means that the camera can be regarded as a black box as shown in Fig. 1 and that no measurements from within the camera are required. Only the input/output relation is considered.

The input signal is the mean number of photons μ_p hitting each pixel during the exposure time. In order to obtain the input signal three pieces of information are required. Firstly, the irradiation E at the sensor plane must be measured using an absolutely calibrated photodiode. Secondly, the integration time must be known, which is normally the exposure time t_{exp} set in the camera. Thirdly, the pixel size

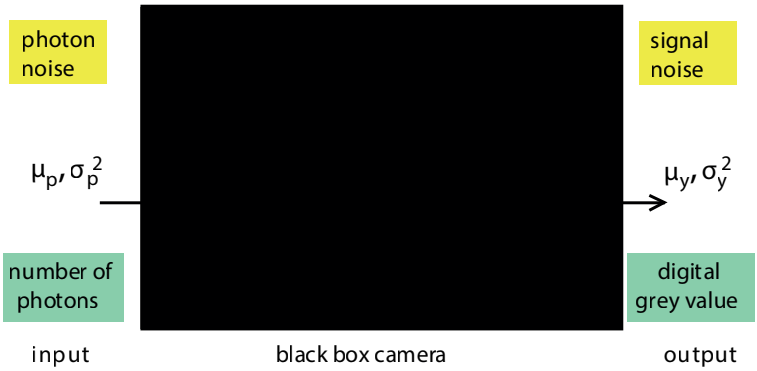


Figure 1: Black box camera model, from [10].

must be known. This is not the light-sensitive area of the pixel, but the whole pixel area A computed by the horizontal and vertical pixel pitch, because the whole pixel is irradiated homogeneously. Thus the pixel-related input signal in units of photons is given by

$$H_{\text{in}} = \mu_p = EAt_{\text{exp}}. \tag{1}$$

It is important to note here that the input signal exhibits temporal noise. Due to the laws of quantum mechanics the input signal follows a Poisson distribution. Therefore the variance of the input signal is equal to its mean:

$$\sigma_p^2 = \mu_p. \tag{2}$$

The output signal is the digital signal y (units DN) with mean μ_y, the temporal variance σ_y^2 and the variance of the spatial nonuniformity s_y^2. The mean of this signal and its variances can be measured for any camera with a digital output. The temporal variance of the output signal includes the variance of the input signal and all further noise sources from the components within the images sensor and signal processing circuits within the camera.

3 Key parameters signal-to-noise ratio and signal-to-nonuniformity (SNR)

According to the discussion in the previous section, the quality of a camera signal is simply given by the relation of the mean output signal versus the standard deviations of the temporal noise and spatial nonuniformity. This results in the signal-to-noise and signal-to-nonuniformity ratios:

$$\mathrm{SNR}_{\mathrm{out}} = \frac{\mu_y - \mu_{y\mathrm{dark}}}{\sigma_y} \quad \text{and} \quad \mathrm{SNR}_{\mathrm{out.nu}} = \frac{\mu_y - \mu_{y\mathrm{dark}}}{s_y} \tag{3}$$

These two rations can be combined to the total SNR

$$\mathrm{SNR}_{\mathrm{out.total}} = \frac{\mu_y - \mu_{y\mathrm{dark}}}{\sqrt{\sigma_y^2 + s_y^2}} == \frac{\mu'_y}{\sqrt{\sigma_y^2 + s_y^2}}. \tag{4}$$

In this way, the SNR can be measured for any camera with a digital output. Only care must be taken that the quantization is not too coarse. Otherwise, the standard deviations would be biased [9, Sec. 5.6.2]. However, one important fact must be considered. This is unusual, because normally only linear systems are considered. In a linear system noise and signal are amplified in the same way. This means that the SNR at the input and the output is the same. The SNR of interest is actually not the output SNR but the input SNR, because the quantity of interest is the measured exposure H. It gives the certainty with which the pixel exposure can be measured. In a non-linear system, it is necessary to differentiate between input and out SNR. Therefore, it is required to find a way to compute the input SNR from the measured output SNR.

Because the characteristic curve $\mu_y(\mu_p)$ is also measured, it is possible to compute the input SNR from the output SNR via inverse error propagation. The two quantities are related to each other by the slope of the characteristic curve:

$$\sigma_y = \frac{\mathrm{d}\mu_y}{\mathrm{d}\mu_p}\sigma_p \quad \rightsquigarrow \quad \mathrm{SNR}_{\mathrm{in}} = \frac{\mu_p}{\sigma_p} = \frac{\mu_p}{\sigma_y}\frac{\mathrm{d}\mu_y}{\mathrm{d}\mu_p} = \frac{\mu_p}{\mu'_y}\frac{\mathrm{d}\mu_y}{\mathrm{d}\mu_p}\mathrm{SNR}_{\mathrm{out}} \tag{5}$$

In this way, the input SNR can be computed from the measured quantities, a) the slope of the characteristic curve, b) the applied mean exposure, μ_p, and c) the measured mean output signal minus the mean dark signal, μ'_y. From Eq. 5, it can also be inferred that input and output SNR are equal only for a linear characteristic curve. It is further important to note that the standard deviation σ_p does not only include the temporal noise of the incoming stream of photons (shot noise) but also all other noise sources within the non-linear camera — back-projected to the input signal.

It is also easy to specify the input SNR for an ideal general sensor with no noise reduction processing and no additional noise sources. Then only the photon noise remains. Therefore the ideal input SNR is given by

$$\mathrm{SNR}_{\mathrm{in.ideal}}(\mu_p) = \sqrt{\mu_p}. \tag{6}$$

From the above considerations, we can draw three important conclusions, which emphasize the importance of the SNR approach for general image sensor and camera quality assessment. Firstly, very different types of cameras can be compared with each other by compareing the input SNR. Secondly, it is possible to specify how much worse a real camera (5) is in comparison with an ideal one (6). Without a more detailed camera model, it is not possible to determine the quantum efficiency[1] of the sensor. However, this is not a significant disadvantage. Derived camera performance parameters really of importance for applications such as the absolute sensitivity threshold, the dynamic range, and the maximum SNR can be derived from the input SNR *without* knowing the quantum efficiency.

4 Discussion of examples

In this section, we show several examples to illustrate the power and usefulness of the discussion in the previous sections. Double logarithmic plots of the SNR are shown, in which derived quality parameters are marked, such as the absolute sensitivity threshold, the saturation capacity, the dynamic range, and the maximal SNR, $\mathrm{SNR}_{\mathrm{max}}$.

[1] The quantum efficiency relative to a maximum response can still be measured by performing measurements over the whole range of wavelengths.

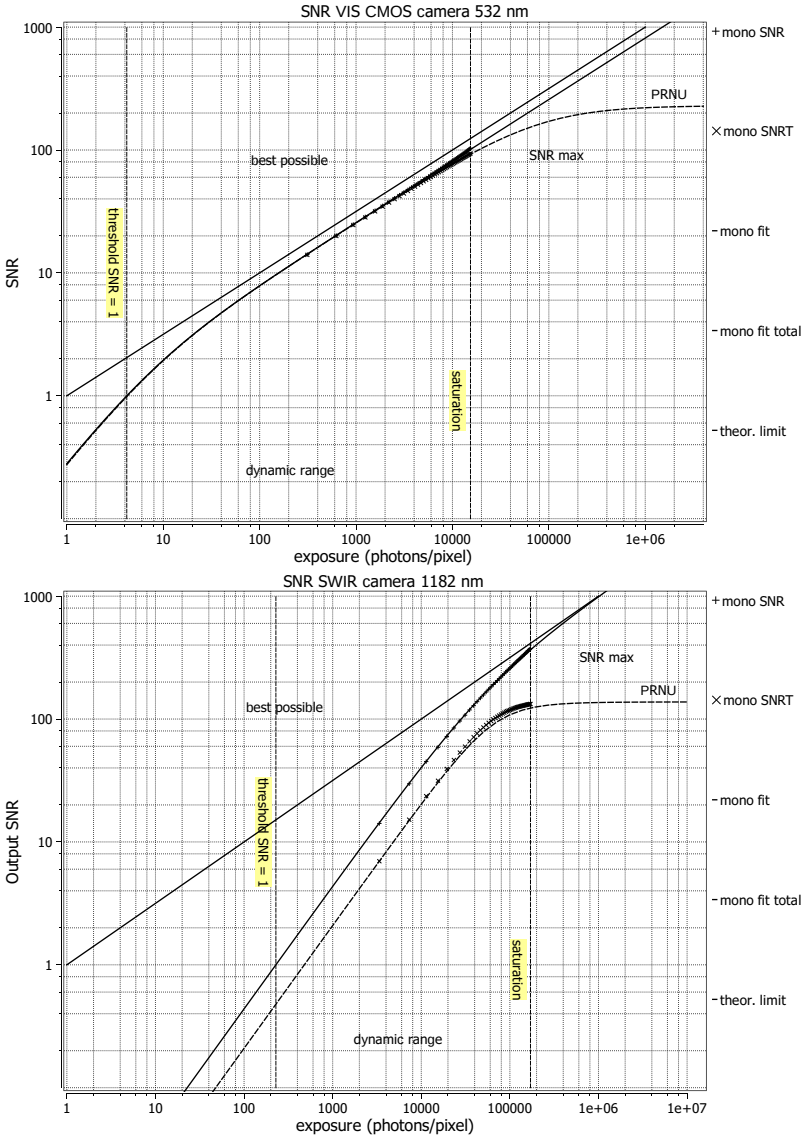


Figure 2: SNR of a typical high-end linear industrial camera in the visible range, measured with a wavelength of 532 nm (top) and a SWIR camera (bottom).

The first case is a typical high-end linear industrial camera with a maximum SNR of about 100 and almost negligible nonuniformity (Fig. 2 top). The camera has also a low temporal noise of the dark signal, because the measured SNR runs parallel to the ideal sensor without any additional noise sources for almost the whole saturation range.

The second example shows the SNR of a typical SWIR camera (Fig. 2 bottom) with quite different properties. The camera has a saturation capacity which is more than a decade higher than the camera in the visible range. Therefore, the maximal SNR is with a value of about 400 four times higher. Two other significant differences are obvious from the direct comparison. Firstly, the quality of the SWIR camera is limited over almost the entire saturation range by the much higher dark noise. Therefore, the absolute sensitivity threshold is also more than 200 photons instead of about 4 for the standard silicon image sensor. Secondly, the nonuniformity is at least as large as the temporal noise. Therfore the total SNR is about a factor of two lower at almost all saturation levels. Close to the maximum saturation level it is even almost a lower by a factor of four.

The last examples shows a linear 24-bit HDR camera (Fig. 3). It illustrates that the standard 1288 is also capable to characterize cameras over a dynamic range of more than 120 dB.

5 Conclusions and outlook

It has been shown, that the EMVA standard 1288 can characterize and compare a wide range of cameras/sensors. Despite the diversity, the central tool is the SNR and total SNR. From the SNR, a minimum set of application-oriented quality parameters can be derived. It is possible to characterize and compare a) simple linear cameras without preprocessing that changes the noise, b) linear cameras with proprocessing, and c) linear and nonlinear HDR cameras. It could also be shown how different the properties of SWIR cameras with a lower band gap are from standard silicon semiconductor cameras.

With the concept of computing the input SNR for nonlinear cameras from the output SNR it will also be possible to apply the analysis to any parameters derived from several channels of a multimodal image sen-

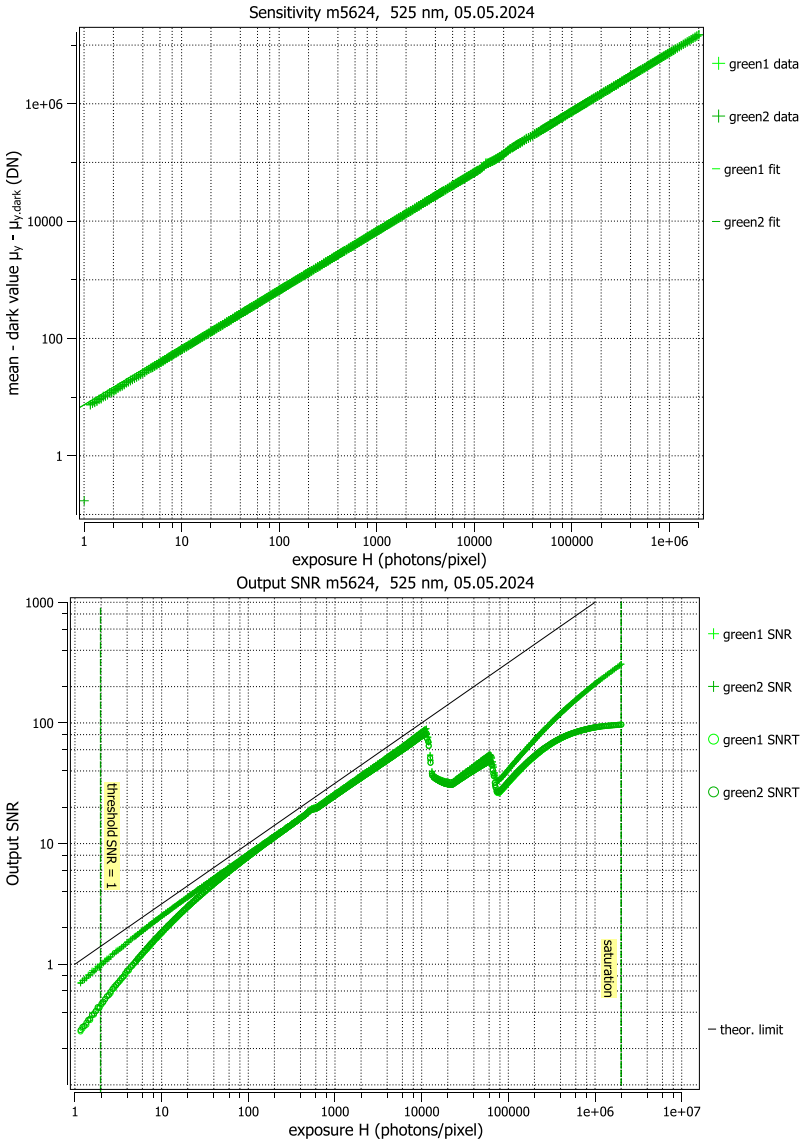


Figure 3: Characteristic curve (top) and SNR (bottom) of a linear 24-bit HDR camera.

sor. Examples include color hue, color saturation, polarization anygle and polarization.

Not yet covered is an entirely different class of image sensors, so-called event-based or neuromorphic sensors. Research to extend the EMVA standard 1288 also for this class of sensors has already started [11], see also the contribution of Manakov et al. in this volume.

Acknowledgments

The author gratefully acknowledges financial support for this research through his senior professorship at Heidelberg University.

References

1. A. Darmont, "Using the EMVA 1288 standard to select an image sensor or camera," in *Sensors, Cameras, and Systems for Industrial/Scientific Applications XI*, ser. Proc. SPIE, E. Bodegom and V. Nguyen, Eds., vol. 7536, 2010, p. 753609.
2. B. Jähne, "EMVA 1288 standard for machine vision – objective specification of vital camera data," *Optik & Photonik*, vol. 5, pp. 53–54, 2010.
3. A. Darmont, J. Chahiba, J. F. Lemaitre, M. Pirson, and D. Dethier, "Implementing and using the EMVA1288 standard," in *Sensors, Cameras, and Systems for Industrial/Scientific Applications XIII*, ser. Proc. SPIE, R. Widenhorn, V. Nguyen, and A. Dupret, Eds., vol. 8298, 2012, p. 82980H.
4. M. Rosenberger, C. Zhang, P. Votyakov, M. Preißler, R. Celestre, and G. Notni, "EMVA 1288 camera characterisation and the influences of radiometric camera characteristics on geometric measurements," *Acta IMEKO*, vol. 5, pp. 81–87, 2016.
5. A. Darmont, *High Dynamic Range Imaging: Sensors and Architectures*, 2nd ed. SPIE, 2019.
6. EMVA 1288 Working Group, "EMVA Standard 1288 - standard for characterization of image sensors and cameras, release A1.00," European Machine Vision Association, open standard, 2005.
7. ——, "EMVA Standard 1288 - standard for characterization of image sensors and cameras, release 3.1," European Machine Vision Association, open standard, 2016.

8. ——, "EMVA Standard 1288 - standard for characterization of image sensors and cameras, release 4.0 linear," European Machine Vision Association, open standard, 2021. [Online]. Available: https://www.emva.org/standards-technology/emva-1288/emva-standard-1288-downloads-2/
9. B. Jähne, *Digitale Bildverarbeitung und Bildgewinnung*, 8th ed. Springer Vieweg, 2024.
10. EMVA 1288 Working Group, "EMVA Standard 1288 - standard for characterization of image sensors and cameras, release 4.0 general," European Machine Vision Association, open standard, 2021. [Online]. Available: https://www.emva.org/standards-technology/emva-1288/emva-standard-1288-downloads-2/
11. A. Manakov and B. Jähne, "Characterization of event-based image sensors in extent of the EMVA 1288 standard," in *Forum Bildverarbeitung*, M. Heizmann and T. Längle, Eds. KIT Scientific Publishing, 2020, pp. 1–11.

Beitrag zur robusten Parameterschätzung
Iteratively reweighted least squares revisited

Bastian Erdnüß

Fraunhofer-IOSB (Institut für Optronik, Systemtechnik und Bildverarbeitung)
Fraunhoferstr. 1, D-76131 Karlsruhe

Zusammenfassung Die Kleinste-Quadrate-Schätzung ist optimal für normalverteilte Messfehler, jedoch anfällig gegenüber groben Messfehlern. M-Schätzer können eine endlastigere Fehlerverteilung berücksichtigen, was sie robuster gegenüber groben Messfehlern macht. In diesem Beitrag wird eine in der Notation einfachere Beschreibung der klassischen Theorie der robusten M-Schätzer vorgestellt und für den Fall von gleichverteilten Ausreißer durchgesprochen. Darüber hinaus wird eine Familie bekannter robuster Verlustfunktionen in diese Notation übersetzt und Verbindungen zu einer Kernel-Lifting-Methode aufgezeigt, die als Alternative zum üblichen IRLS-Algorithmus zur Berechnung von M-Schätzern verwendet werden kann.

Schlüsselwörter Kleinste Quadrate, Robuste Schätzung, IRLS

Abstract The least squares estimator is optimal for normally distributed measurement errors, but it can break down under gross measurement errors. M-estimators can take fat-tailed error distribution into account, which makes them more robust to gross measurement errors. In this paper, a simpler description of the classical theory of robust M-estimators is presented and used to describe M-estimators for uniformly distributed outliers. In addition, a family of well known robust loss functions is presented in this notation and connections to a kernel lifting method are shown, which can be used as an alternative to the usual IRLS algorithm for calculating the M-estimators.

Keywords Least squares, robust estimation, IRLS

1 Der Kleinste-Quadrate-Schätzer

Der Kleinste-Quadrate-Schätzer ergibt sich als Maximum-Likelihood-Schätzer der Normalverteilung. Sind Beobachtungen $\boldsymbol{y} \in \mathbb{R}^N$ N-dimensional normalverteilt mit Erwartungswert $\boldsymbol{\mu} \in \mathbb{R}^N$ und Kovarianzmatrix $\Sigma \in \mathbb{R}^{N\times N}$ (geschrieben: $\boldsymbol{y} \sim \mathcal{N}_N(\boldsymbol{\mu}, \Sigma)$), so entspricht die *Likelihood* gegeben der Beobachtungen $\boldsymbol{y}$ der Wahrscheinlichkeitsdichte von $\boldsymbol{y}$:

$$l(\boldsymbol{\mu}, \Sigma \mid \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp(-\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\mu}\|^2_{\Sigma^{-1}}) . \tag{1}$$

Hierbei steht $|\cdot|$ für die Determinante und $\|\boldsymbol{x}\|_A = \sqrt{\boldsymbol{x}^\mathsf{T} A \boldsymbol{x}}$ für die Norm bzgl. einer Metrik A. Die Likelihood ist maximal, wenn die negative Log-Likelihood

$$\lambda(\boldsymbol{\mu}, \Sigma \mid \boldsymbol{y}) = -\log\left(l(\boldsymbol{\mu}, \Sigma \mid \boldsymbol{y})\right) = \frac{1}{2}\log(|2\pi\Sigma|) + \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\mu}\|^2_{\Sigma^{-1}} \tag{2}$$

minimal ist.

Ein paar Spezialfälle werden nun genauer betrachtet. Liegen N unabhängig identisch eindimensional normalverteilte Beobachtungen $y_i \sim \mathcal{N}(\mu, \sigma^2)$ vor (also $\boldsymbol{\mu} = \mu\mathbf{1}$ der mit dem Faktor $\mu \in \mathbb{R}$ skalierte Konstant-1-Vektor und $\Sigma = \sigma^2 I$ die mit dem Faktor σ^2 skalierte Einheitsmatrix), so ist

$$\lambda(\mu, \sigma^2 \mid \boldsymbol{y}) = C + \frac{N}{2}\log(\sigma^2) + \frac{1}{2\sigma^2}\sum_i (y_i - \mu)^2 \tag{3}$$

mit der Konstante $C = \frac{N}{2}\log(2\pi)$. Unabhängig von der Wahl von σ^2 ist $\sum_i (y_i - \mu)^2$ minimal, wenn $\mu = \overline{y} = \frac{1}{N}\sum_i y_i$ der Mittelwert der Beobachtungen y_i ist. Dies ist der Maximum-Likelihood-Schätzer des Erwartungswerts μ der Beobachtungen y_i. Da $\mu = \overline{y}$ die Summe der Quadrate $\sum_i (y_i - \mu)^2$ minimiert, wird er auch als *Kleinste-Quadrate-Schätzer* bezeichnet.

Im zweiten betrachteten Fall ist $\boldsymbol{y} \sim \mathcal{N}_N(\boldsymbol{\mu}(\boldsymbol{x}), s^2 Q)$ mit linearem $\boldsymbol{\mu}(\boldsymbol{x}) = \boldsymbol{a} + A\boldsymbol{x}$, sowie $\boldsymbol{a} \in \mathbb{R}^N$, $\boldsymbol{x} \in \mathbb{R}^M$, $A \in \mathbb{R}^{M\times N}$, $Q \in \mathbb{R}^{N\times N}$ symmetrisch positiv definit und $s^2 > 0$. $\boldsymbol{x}$ ist ein zu schätzender Parametervektor und $\boldsymbol{a}, A, Q$ beschreiben das als bekannt vorausgesetzte

stochastische Modell der Beobachtungen. In dem Fall ist

$$\lambda(\boldsymbol{x}, s^2 \mid \boldsymbol{y}) = C + \frac{N}{2}\log(s^2) + \frac{1}{2s^2}\|\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{x})\|^2_{Q^{-1}} \tag{4}$$

mit der Konstante $C = \frac{1}{2}\log(|2\pi Q|)$, wobei $\|\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{x})\|^2_{Q^{-1}}$ unabhängig von s^2 minimal wird, wenn

$$\boldsymbol{x} = (A^\mathsf{T} Q^{-1} A)^{-1} A^\mathsf{T} Q^{-1}(\boldsymbol{y} - \boldsymbol{a}) \tag{5}$$

ist. (5) ist der Schätzer des linearen Gauß-Markoff-Modells, vgl. [1, Gl. (4.41)], und stellt gewissermaßen die Basis der Ausgleichsrechnung dar.

Schließlich wird noch ein dritter Fall betrachtet, in dem die Beobachtungen $\boldsymbol{y} = (\boldsymbol{y}_i)_i$ in n-dimensonale stochastisch unabhängige Beobachtungsgruppen $\boldsymbol{y}_i \sim \mathcal{N}_n(\boldsymbol{\mu}_i(\boldsymbol{x}), s^2 Q_i)$ mit $\boldsymbol{\mu}_i(\boldsymbol{x}) = \boldsymbol{a}_i + A_i\boldsymbol{x}$ zerfallen. In dem Fall ist

$$\lambda(\boldsymbol{x}, s^2 \mid \boldsymbol{y}) = C + \frac{N}{2}\log(s^2) + \frac{1}{2s^2}\sum_i \|\boldsymbol{y}_i - \boldsymbol{\mu}_i(\boldsymbol{x})\|^2_{Q_i^{-1}} \tag{6}$$

mit der Konstante $C = \frac{1}{2}\sum_i \log(|2\pi Q_i|)$, wobei N die Gesamtdimension aller Beobachtungen ist. $\sum_i \|\boldsymbol{y}_i - \boldsymbol{\mu}_i(\boldsymbol{x})\|^2_{Q_i^{-1}}$ wird dann unabhängig von s^2 minimal, wenn

$$\boldsymbol{x} = \Big(\sum_i A_i^\mathsf{T} Q_i A_i\Big)^{-1} \sum_i A_i^\mathsf{T} Q_i(\boldsymbol{y}_i - \boldsymbol{a}_i) \tag{7}$$

ist. Häufig wird (5) intern mit (7) berechnet, um die üblicherweise vorhandene Block-Diagonal-Struktur vor Q algorithmisch effizient zu nutzen. Im Folgenden wird sich diese Darstellung jedoch auch methodisch als sinnvoll erweisen.

2 Robuste Schätzung

Problematisch am Kleinste-Quadrate-Schätzer ist seine Anfälligkeit ggü. Ausreißern. Ein einziger grob falscher Messwert kann den Mittelwert beliebig weit verschieben. Huber hat daher in [2] vorgeschlagen, statt wie in (3) die Summe $\sum_i r_i^2$ der Quadrate der Residuen

$r_i = y_i - \mu$, die Summe $\sum_i \rho(r_i)$ anderer Verlustfunktionen ρ der Residuen zu minimieren, um dadurch robustere Schätzer zu erhalten, die er als M-Schätzer bezeichnet. Bspw. führt die Minimierung der Summe der Absolutresiduen mit $\rho(r) = |r|$ auf den bekanntermaßen robusten Medianschätzer $\mu = \text{med}_i(y_i)$. Huber basiert viele seiner Untersuchungen auf die Einflussfunktion $\psi(r) = \rho'(r)$. Passend zu den Ergebnissen von Huber gibt es den IRLS-Algorithmus (iteratively reweighted least squares, vermutlich auf unveröffentlichte Arbeiten von Tukey zurückgehend, vgl. [3]), der mithilfe der Gewichtsfunktion $w(r) = \psi(r)/r$ Summen vom Typ $\sum_i \rho(r_i)$ mit oft nur wenigen Iterationen minimieren kann. Eine derartige Darstellung der Theorie findet sich z. B. in [1, Kap. 4.7.4].

In diesem Artikel wird eine andere Darstellung näher an [4, Kap. 3.3] präsentiert, bei der ρ statt in den Residuen r_i in den halben quadrierten Residuen $\Omega_i = \frac{1}{2} r_i^2$ parametrisiert wird. Dies erlaubt eine einfachere Darstellung der Theorie bei mehrdimensionalen Beobachtungsgruppen.

2.1 Die Verteilung $\mathcal{W}_{w,n}$

In dieser Darstellung werden Verteilungen mit einem gewissen Grad an Symmetrie um ihr Zentrum $\boldsymbol{\mu}$ betrachtet. Diese Verteilungen sollen durch die halbe quadratische Mahalanobisdistanz

$$\Omega_{\boldsymbol{\mu},\Sigma}(\boldsymbol{y}) = \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{\mu}\|^2_{\Sigma^{-1}} \tag{8}$$

faktorisieren. Dazu wird eine n-dimensionale Verteilung $\mathcal{W}_{w,n}(\boldsymbol{\mu}, \Sigma)$ mit Lageparameter $\boldsymbol{\mu} \in \mathbb{R}^n$ und Skalenparameter $\Sigma \in \mathbb{R}^{n \times n}$ auf Basis einer integrierbaren Gewichtsfunktion $w : \mathbb{R}_{\geq 0} \to \overline{\mathbb{R}}_{\geq 0}$ definiert. Diese soll die Wahrscheinlichkeitsdichte

$$p_{\mathcal{W}_{w,n}(\boldsymbol{\mu},\Sigma)}(\boldsymbol{y}) = \frac{\Gamma(\frac{n}{2})}{\sqrt{|2\pi\Sigma|}} \cdot p_{w,n}\big(\Omega_{\boldsymbol{\mu},\Sigma}(\boldsymbol{y})\big) \tag{9}$$

mit der Eulerschen Gammafunktion $\Gamma(k) = \int_0^\infty \exp(-t)\, t^{k-1}\, \mathrm{d}t$ und

$$p_{w,n}(\Omega) = \frac{\exp\big(-\rho_w(\Omega)\big)}{\Gamma_w(\frac{n}{2})} \quad \text{mit} \quad \rho_w(\Omega) = \int_0^\Omega w(\omega)\, \mathrm{d}\omega \tag{10}$$

sowie der Normierungskonstante

$$\Gamma_w(k) = \int_0^\infty \exp\left(-\rho_w(t)\right)\, t^{k-1}\,\mathrm{d}t \tag{11}$$

haben. Dieser Zugang unterscheidet sich von [4, Kap. 3.3] nur dahingehend, dass ρ in $\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{\mu}\|^2_{\Sigma^{-1}}$ parametrisiert ist statt in $\|\boldsymbol{y}-\boldsymbol{\mu}\|^2_{\Sigma^{-1}}$ (ohne dem Vorfaktor $\frac{1}{2}$). Dieses Vorgehen bring deutliche Vorteile in der Notation mit sich, u. a. den auch in [5, Gl. (2)] angedeuteten einfachen Zusammengang $\rho'_w(\Omega) = w(\Omega)$ zwischen Verlustfunktion ρ_w und zugehöriger Gewichtsfunktion w.

Nicht für alle Gewichtsfunktionen w ist $\mathcal{W}_{w,n}(\boldsymbol{\mu},\Sigma)$ eine Wahrscheinlichkeitsverteilung. $w(\Omega)\cdot\Omega$ ist die *gewichtete* halbe quadratische Mahalanobisdistanz, an deren Grenzwert $\Omega_w = \lim_{\Omega\to\infty} w(\Omega)\cdot\Omega$ sich ablesen lässt, ob das Integral $\Gamma_w(n/2)$ konvergiert und $\mathcal{W}_{w,n}(\boldsymbol{\mu},\Sigma)$ damit zu einer echten Wahrscheinlichkeitsverteilung macht. Existiert der Grenzwert Ω_w und gilt für diesen $\Omega_w > n/2$ (inkl. $\Omega_w = \infty$), so konvergiert $\Gamma_w(n/2)$. Gilt für den Grenzwert dagegen $\Omega_w < n/2$, so divergiert $\Gamma_w(n/2)$. Für $\Omega_w = n/2$ oder falls Ω_w nicht existiert, ist eine genauere Untersuchung notwendig. Divergiert $\Gamma_w(n/2)$, kann mit der uneigentlichen Wahrscheinlichkeitsdichte $h\exp\left(-\rho_w\left(\Omega_{\boldsymbol{\mu},\Sigma}(\boldsymbol{x})\right)\right)$ mit unbestimmtem Skalierungsfaktor h gearbeitet werden oder mit der auf $\Omega_{\boldsymbol{\mu},\Sigma}(\boldsymbol{x}) \le \Omega_{\max}$ eingeschränkten Wahrscheinlichkeitsdichte, die entsteht, in dem man die Gewichtsfunktion formal mit $w(\Omega) = \infty$ für $\Omega > \Omega_{\max}$ anpasst, wodurch $\rho_w(\Omega) = \infty$ und damit $p_{w,n}(\Omega) = 0$ für $\Omega > \Omega_{\max}$ werden. Der Normierungsfaktor $\Gamma_w(n/2)$ in $p_{w,n}$ kann dann auch durch das unvollständige Integral $\gamma_w(n/2,\Omega_{\max})$ mit der ursprünglichen Gewichtsfunktion w ohne Anpassung ab $\Omega_{\max}$ ersetzt werden, für das gilt:

$$\gamma_w(k,T) = \int_0^T \exp\left(-\rho_w(t)\right)\, t^{k-1}\,\mathrm{d}t\,. \tag{12}$$

Ist $\Omega_w > (n+1)/2$ existiert der Erwartungswert der $\mathcal{W}_{w,n}(\boldsymbol{\mu},\Sigma)$-Verteilung und ist $\boldsymbol{\mu}$. Ist $\Omega_w > (n+2)/2$ existiert auch die Kovarianzmatrix der Verteilung und ist $s^2_{w,n}\Sigma$ mit dem Skalierungsfaktor

$$s^2_{w,n} = \frac{\Gamma_w\left(\frac{n}{2}+1\right)}{\frac{n}{2}\,\Gamma_w\left(\frac{n}{2}\right)}\,, \qquad \text{bzw.} \qquad s^2_{w,n} = \frac{\gamma_w\left(\frac{n}{2}+1,\Omega_{\max}\right)}{\frac{n}{2}\,\gamma_w\left(\frac{n}{2},\Omega_{\max}\right)}\,. \tag{13}$$

Ist $\Omega_w > (n+3)/2$ existieren die 3. Zentralmomente der Verteilung und verschwinden, d. h. die Verteilung ist *symmetrisch*. Ist Ω_w exakt $(n+1)/2$, $(n+2)/2$ oder $(n+3)/2$ ist jeweils eine genauere Untersuchung notwendig, ob die entsprechenden Momente existieren. Ist Ω_w dagegen kleiner, existieren sie nicht.

Die Verteilungen $\mathcal{W}_{w,n}(\boldsymbol{\mu}, \Sigma)$ haben ihren Modus in $\boldsymbol{\mu}$ und fallen um diesen herum symmetrisch ab. Dadurch ist zu erwarten, dass die Maximum-Likelihood-Methode auf diesem Verteilungstyp sinnvolle Ergebnisse liefert, da sich Fehler in alle Richtungen symmetrisch ausgleichen können.

Für die konstante Gewichtsfunktion $w(\Omega) = s^{-2}$ ist $\Gamma_w(\frac{n}{2}) = s^n\,\Gamma(\frac{n}{2})$ und $\mathcal{W}_{w,n}(\boldsymbol{\mu}, \Sigma) = \mathcal{N}_n(\boldsymbol{\mu}, s^2\,\Sigma)$ die n-dimensionale Normalverteilung mit Erwartungswert $\boldsymbol{\mu}$ und Kovarianzmatrix $s^2\,\Sigma$. Insbesondere ist $\mathcal{W}_{w,n}(\boldsymbol{\mu}, \Sigma) = \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ für die konstante Gewichtsfunktion $w(\Omega) = 1$. Für die konstante Gewichtsfunktion $w(\Omega) = 0$ ergibt sich die uneigentliche Gleichverteilung oder die Gleichverteilung auf $\Omega < \Omega_{\max}$, wenn sie mit $w(\Omega) = \infty$ für $\Omega > \Omega_{\max}$ bei $\Omega_{\max}$ abgeschnitten wird.

2.2 Iteratively Reweighted Least Squares (IRLS)

Mit $\boldsymbol{\mu} = \boldsymbol{\mu}(\boldsymbol{x})$ und $\Sigma = s^2 Q$ ist die negative Log-Likelihood von (9)

$$\lambda_{\mathcal{W}_{w,n}(\boldsymbol{\mu}(\boldsymbol{x}),s^2Q)}(\boldsymbol{x}, s^2 \mid \boldsymbol{y}) = C + \frac{n}{2}\log(s^2) + \rho_w\big(s^{-2}\,\Omega_{\boldsymbol{\mu}(\boldsymbol{x}),Q}(\boldsymbol{y})\big) \quad (14)$$

mit $C = \log\left(\frac{\Gamma_w(n/2)}{\Gamma(n/2)}\sqrt{|2\pi Q|}\right)$. Für die n-dimensionalen Beobachtungsgruppen $\boldsymbol{y}_i \sim \mathcal{W}_{w,n}(\boldsymbol{\mu}_i(\boldsymbol{x}), s^2 Q_i)$ ergibt sich analog zu (6) die zu minimierende negative Log-Likelihood

$$\lambda(\boldsymbol{x}, s^2 \mid \boldsymbol{y}) = C + \frac{N}{2}\log(s^2) + \sum_i \rho_w(s^{-2}\,\Omega_i) \quad (15)$$

mit $C = \sum_i \log\left(\frac{\Gamma_w(n/2)}{\Gamma(n/2)}\sqrt{|2\pi Q_i|}\right)$ und $\Omega_i = \Omega_{\boldsymbol{\mu}_i(\boldsymbol{x}),Q_i}(\boldsymbol{y}_i)$. Die Beobachtungsgruppen $\boldsymbol{y}_i$ könnten auch unterschiedliche Dimensionen n_i und unterschiedliche Gewichtsfunktionen w_i haben, z. B. wenn unterschiedliche Beobachtungstypen wie 2D-Featurepunkte und 3D-GNSS-Messungen miteinander kombiniert werden, oder wenn die Beobachtungen gegen qualitativ unteschiedliche Ausreißer anfällig sind. In

dem Fall ist $N = \sum_i n_i$ in Formel (15) die Gesamtdimension aller Beobachtungen.

Damit (15) minimal in $\boldsymbol{x}$ ist, muss

$$0 = \frac{\partial}{\partial \boldsymbol{x}} \lambda(\boldsymbol{x}, s^2 \,|\, \boldsymbol{y}) = s^{-2} \sum_i \rho'_w(s^{-2}\,\Omega_i)\, \frac{\partial}{\partial \boldsymbol{x}} \Omega_i \tag{16}$$

$$= s^{-2} \sum_i w(s^{-2}\,\Omega_i)(\boldsymbol{y}_i - \boldsymbol{\mu}_i(\boldsymbol{x}))^{\mathsf{T}} Q_i^{-1}\, \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{\mu}_i(\boldsymbol{x}) \tag{17}$$

sein, und mit $\Omega_i = \Omega_{\boldsymbol{\mu}_i(\boldsymbol{x}),Q_i}(\boldsymbol{y}_i)$ und $w_i = w(\Omega_i/s^2)$ ist das im linearen Gauß-Markoff-Modell $\boldsymbol{\mu}_i(\boldsymbol{x}) = \boldsymbol{a}_i + A_i\boldsymbol{x}$ erfüllt, wenn

$$\boldsymbol{x} = \big(\sum_i w_i\, A_i^{\mathsf{T}} Q_i^{-1} A_i\big)^{-1} \sum_i w_i\, A_i^{\mathsf{T}} Q_i^{-1} (\boldsymbol{y}_i - \boldsymbol{a}_i) \tag{18}$$

ist. Abgesehen von den Gewichten w_i entspricht diese Formel gerade (7). Jedoch ist zu beachten, dass hier die w_i selbst sowohl von $\boldsymbol{x}$ als auch von s^2 abhängen. Dennoch können startend von Näherungswerten $\boldsymbol{x}$ und s^2 iterativ die Gewichte w_i berechnet werden und damit eine verbesserte Lösung für $\boldsymbol{x}$ berechnet werden. Dies ist der IRLS-Algorithmus.

Um s^2 robust zu schätzen, gibt es mehrere Möglichkeiten, z. B. [1, Kap. 4.7.3] oder den mit leichtem Bias versehenen Maximum-Likelihood-Schätzer, der entsteht, wenn (15) nach s^2 abgeleitet und dessen Nullstelle berechnet wird. Das führt auf

$$s^2 = \frac{2}{N} \sum_i w_i\, \Omega_i \tag{19}$$

wobei zu beachten ist, dass $w_i = w(\Omega_i/s^2)$ selbst von s^2 abhängt und (19) aufgefasst als Fixpunktgleichung $v = f(v) = \frac{2}{N} \sum_i w(\Omega_i/v)\, \Omega_i$ mit $v = s^2$ nicht zwingend konvergieren muss.

2.3 Bekannte Gewichtsfunktionen

Barron [6] hat eine Funktionsfamilie aufgezeigt, die viele der in der Literatun bekannten Schätzer umfasst. In der hier gewählten Darstellung hat sie die Form

$$w_{\beta,s^2,k}(\Omega) = \frac{k}{s^2}\, w_{\beta,1,1}\left(\frac{\Omega}{s^2}\right) \quad \text{mit} \quad w_{\beta,1,1}(\Omega) = \left(1 + \frac{\Omega}{\beta}\right)^{-\beta} \tag{20}$$

für $0 < \beta < \infty$ mit den Grenzwerten $w_{0,1,1}(\Omega) = 1$ und $w_{\infty,1,1}(\Omega) = \exp(-\Omega)$. Die Familie umfasst die Normalverteilungen $w_{0,1,s^{-2}}$, den geglätteten Huber-Schätzer $w_{1/2,s^2,s^2}$, die n-dimensionale Cauchy-Verteilung $w_{1,s^2/2,(n+1)/2}$, den Geman-McClure-Schätzer $w_{2,s^2,1}$ und den Welsch-Schätzer w_{∞,s^2,s^2}. Es gilt

$$\rho_{w_{\beta,s^2,k}}(\Omega) = k\,\rho_{w_{\beta,1,1}}\left(\frac{\Omega}{s^2}\right) \tag{21}$$

$$\rho_{w_{\beta,1,1}}(\Omega) = \frac{\beta}{1-\beta}\left(\left(1+\frac{\Omega}{\beta}\right)^{1-\beta} - 1\right) \tag{22}$$

mit den Grenzwerten $\rho_{w_{0,1,1}}(\Omega) = \Omega$ und $\rho_{w_{\infty,1,1}}(\Omega) = 1-\exp(-\Omega)$ und dem Sonderfall $\rho_{w_{1,1,1}}(\Omega) = \log(1+\Omega)$.

$w_{\beta,s^2,k}$ führt für $\beta < 1$ auf eine Wahrscheinlichkeitsverteilung, zu der alle Momente existieren. Für $\beta > 1$ lässt sich die entstehende Verteilung dagegen nicht normieren und nur als uneigentliche oder abgeschnittene Wahrscheinlichkeitsverteilung verwenden. Für $\beta = 1$ hängt die Situation von dem Wert von k ab. Für $k > \frac{n}{2}$ entsteht eine n-dimensionale Wahrscheinlichkeitsverteilung zu der nur genau die Momente kleiner als $2k-n$ existieren. Für $k \leq \frac{n}{2}$ lässt sich die entstehende n-dimensionale Verteilung dagegen wieder nicht normieren.

[6] schlägt vor, startend von $\beta = 0$ schrittweise $\beta \to \infty$ laufen zu lassen, wodurch Ausreißer zunehmend abgewertet werden. In der hier gewählten Darstellung zeigt sich ein auffälliger Zusammenhang der Funktionsfamilie (20) zur bekannten Approximation $\exp(x) = \lim_{n\to\infty}(1+\frac{x}{n})^n$ der Exponentialfunktion in $w_{\beta,1,1}$.

2.4 Mischungen von $\mathcal{W}_{w_i,n}$

Die Mischung zweier Zufallsvariablen $\boldsymbol{y}_0$ und $\boldsymbol{y}_1$ ist die Zufallsvariable $\boldsymbol{y} = \boldsymbol{y}_I$ mit dem zufälligen Index $I \sim \mathcal{B}(\varepsilon)$, der mit Wahrscheinlichkeit $\varepsilon \in [0,1]$ den Wert 1 annimmt und mit Wahrscheinlichkeit $1-\varepsilon$ den Wert 0. Die Wahrscheinlichkeitsdichte $p_{\boldsymbol{y}}$ von $\boldsymbol{y}$ ist

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = (1-\varepsilon)\,p_{\boldsymbol{y}_0}(\boldsymbol{y}) + \varepsilon\,p_{\boldsymbol{y}_1}(\boldsymbol{y})\,, \tag{23}$$

wobei $p_{\boldsymbol{y}_i}$ für $i = 0,1$ jeweils die Wahrscheinlichkeitsdichte von $\boldsymbol{y}_i$ ist. Sind $\boldsymbol{y}_i \sim \mathcal{W}_{w_i,n}(\boldsymbol{\mu},\Sigma)$, so ist auch $\boldsymbol{y} \sim \mathcal{W}_{w,n}(\boldsymbol{\mu},\Sigma)$ mit

$$p_{w,n}(\Omega) = (1-\varepsilon)\;p_{w_0,n}(\Omega) + \varepsilon\;p_{w_1,n}(\Omega)\,. \tag{24}$$

Aus (10) folgt $p'_{w,n}(\Omega) = -p_{w,n}(\Omega) \cdot w(\Omega)$ und leitet man (24) nach Ω ab, erhält man nach Multiplikation mit -1

$$p_{w,n}(\Omega)\ w(\Omega) = (1-\varepsilon)\ p_{w_0,n}(\Omega)\ w_0(\Omega) + \varepsilon\ p_{w_1,n}(\Omega)\ w_1(\Omega)\ . \quad (25)$$

Löst man das nach $w(\Omega)$ auf und ersetzt darin $p_{w,n}(\Omega)$ mit (24), so folgt

$$w(\Omega) = \frac{(1-\varepsilon)\ p_{w_0,n}(\Omega)\ w_0(\Omega) + \varepsilon\ p_{w_1,n}(\Omega)\ w_1(\Omega)}{(1-\varepsilon)\ p_{w_0,n}(\Omega) + \varepsilon\ p_{w_1,n}(\Omega)}\ . \quad (26)$$

Folgt $\boldsymbol{y}_0$ der Wahrscheinlichkeitsverteilung der Inlier und $\boldsymbol{y}_1$ der Wahrscheinlichkeitsverteilung der Ausreißer, so folgt $\boldsymbol{y}$ der Wahrscheinlichkeitsverteilung, die entsteht, wenn man Inlier mit einem Anteil von ε an Ausreißern kontaminiert.

2.5 Gleichverteilte Ausreißer

Es wird angenommen, dass die Inlier einer Normalverteilung mit Erwartungswert $\boldsymbol{\mu}$ und Kovarianzmatrix Σ folgen, dass also konstant $w_0(\Omega) = 1$ ist. [4, Kap. 3.3] und [1, Fig. 4.13] schlagen beide die Gleichverteilung für die Ausreißer vor, geben jedoch nicht die dafür notwendige Gewichtsfunktion

$$w(\Omega) = \left(1 + k \exp(\Omega)\right)^{-1} \quad (27)$$

an. Diese lässt sich aus (26) mit $w_0(\Omega) = 1$ und $w_1(\Omega) = 0$ durch kürzen des Zählers ermitteln, wobei sich $p_{w_1,n}(\Omega) = 1/\gamma_{w_1}(\frac{n}{2}, \Omega_{\max}) = \frac{n}{2}\Omega_{\max}^{-n/2}$ ergibt und $k = \varepsilon/(1-\varepsilon) \cdot \Gamma(\frac{n}{2}+1) \cdot \Omega_{\max}^{-n/2}$ substituiert wurde.

Für große k kann die Gewichtsfunktion (27) durch $w_{\infty,1,1/k}(\Omega) = \exp(-\Omega)/k$ angenähert werden, wobei k insbesondere dann groß wird, wenn der Ausreißeranteil ε groß ist. Diese Annäherung ist proportional zur Gewichtsfunktion des Welsch-Schätzers und liefert zu vorgegebenem s^2 daher im Grenzwert dieselben Ergebnisse.

Für (27) ergibt sich durch dividieren von (25) durch $w(\Omega)$ wegen $w_0(\Omega) = 1$ und $w_1(\Omega) = 0$

$$p_{w,n}(\Omega) = \frac{(1-\varepsilon)p_{w_0,n}(\Omega)}{w(\Omega)} = \frac{1-\varepsilon}{w(\Omega)} \cdot \frac{\exp(-\Omega)}{\Gamma(\frac{n}{2})}\ . \quad (28)$$

Ist I_i das Ereignis, dass es sich bei der i. Beobachtung um einen Inlier handelt, mit a-priori Wahrscheinlichkeit $P(I_i) = 1 - \varepsilon$, so ist mit (28) die a-posteriori Wahrscheinlichkeit zu gegebenem $\boldsymbol{y}_i$ nach Bayes,

$$P(I_i \mid \boldsymbol{y}_i) = \frac{P(\boldsymbol{y}_i \mid I_i)P(I_i)}{P(\boldsymbol{y}_i)} = \frac{p_{w_0,n}(\Omega_i)(1-\varepsilon)}{p_{w,n}(\Omega_i)} = w(\Omega_i) \tag{29}$$

mit $\Omega_i = \Omega_{\boldsymbol{\mu},\Sigma}(\boldsymbol{y}_i)$. Dadurch lassen sich die Summen der Form $\sum_i w_i T_i$ in (18) und (19) als empirische Erwartungswerte über die mit den Inlierwahrscheinlichkeiten $w_i = P(I_i|\boldsymbol{y}_i)$ gewichteten Beobachtungen $\boldsymbol{y}_i$ auffassen. Diese Interpretation ist nur für die Gewichtsfunktion (27) möglich, denn für andere Gewichtsfunktionen gilt im Allgemeinen $P(I_i \mid \boldsymbol{y}_i) \neq w(\Omega_i)$.

$W = \sum_i P(I_i \mid \boldsymbol{y}_i) = \sum_i w_i$ ist die a-posteriori zu erwartende Anzahl an Inliern, gegeben der Beobachtungen $\boldsymbol{y}_i$. Da diese mit der Anzahl M der n-dimensionalen Beobachtungen $\boldsymbol{y}_i$ a-priori erwartungsgemäß $E[W] = (1-\varepsilon)M$ ist, ist bei gleichverteilten Ausreißern zu vorgegebenem k

$$1 - \varepsilon = \frac{1}{M}\sum_i w_i = \overline{w} \tag{30}$$

ein Schätzer für den Inlieranteil.

Analog zu dem Vorgehen in [6] bietet es sich an, k startend von 0 schrittweise zu erhöhen, wodurch Ausreißer zunehmend abgewichtet werden. Mit (30) in (27) ist deren negative Log-Likelihood

$$\lambda(k \mid \boldsymbol{y}) = C - M\log(\overline{w}) + \sum_i \log(w_i) \tag{31}$$

mit von k unabhängigem C minimal in k, wenn

$$\frac{1}{M}\sum_i \log(w_i) - \log(\overline{w}) = \overline{\log(w)} - \log(\overline{w}) \tag{32}$$

minimal in k ist. Grundsätzlich lässt sich k durch Ableiten von (32) nach k und berechnen der Nullstellen bestimmen, jedoch führt das auf komplizierte Formeln. Stattdessen ist es einfacher, k wachsen zu lassen, solange (32) fällt und aufzuhören, sobald es zu steigen beginnt.

3 Zusammenhang zur Lifting Methode

In [7, Kap. 3.4] weist Zach auf eine Lifting-Methode hin, die mit einer geeigneten Kernelfunktion dieselben Ergebnisse wie IRLS liefern kann, jedoch teilweise einen größeren Konvergenzbereich aufweisen soll. In etwas angepasster Notation wird dazu

$$\min_{\boldsymbol{x}} \sum_i \rho_w(\Omega_i) = \min_{\boldsymbol{x},\boldsymbol{w}} \sum_i \big(w_i\Omega_i + \varphi_\Omega(w_i)\big) \tag{33}$$

mit $\Omega_i = \Omega_{\boldsymbol{\mu}_i(\boldsymbol{x}),\Sigma}(\boldsymbol{y}_i)$ minimiert, wobei die linke Seite hier für die Lösung des IRLS-Algorithmus steht (ohne Berücksichtigung eines Skalenparameters s^2 oder Parameter der Gewichtsfunktion w) und die rechte Seite die ersatzweise zu minimierende Funktion der Kernelmethode darstellt. Damit die beiden Lösungen übereinstimmen muss die Kernelfunktion φ_Ω zur Gewichtsfunktion w auf der linken Seite passen. Für monoton fallende w ist dies in der hier gewählten Darstellung besonders einfach, denn mit der Umkehrfunktion $\Omega(w)$ von $w(\Omega)$ und $w_0 = w(0) = \max_\Omega w(\Omega)$ ist

$$\varphi_\Omega(w) = \int_w^{w_0} \Omega(\omega)\,\mathrm{d}\omega\ . \tag{34}$$

Für gleichverteilte Ausreißer ist bspw. $\Omega(w) = \log\left(\frac{1-w}{kw}\right)$ durch auflösen von (27) nach Ω und durch integrieren ergibt sich hierfür

$$\varphi_\Omega(w) = (1-w)\log\left(\frac{1-w}{kw}\right) + \log(w)\ . \tag{35}$$

Analog lassen sich die Gewichtsfunktionen (20) nach Ω auflösen und integrieren, was mit elementaren Mitteln machbar ist, allerdings zu etwas sperrigen Ausdrücken führt.

4 Zusammenfassung

In diesem Artikel wurden die auf Huber [2] zurückgehenden M-Schätzer und der IRLS-Algorithmus (iteratively reweighted least squares) zu deren Berechnung betrachtet. Es wurde ein alternativer Zugang dazu gegeben, der die Verlustfunktionen und Gewichtsfunktionen im halben quadratischen Fehler parametrisiert. Dadurch entfallen viele der sonst notwendigen Zwischenschritte und die Darstellung

wird schlanker. Auch werden Querverbindungen sichtbar, die in der üblichen Darstellung verborgen bleiben. Diese ist zum einen ein Zusammenhang zur eulerschen Gammafunktion über (11), der auch z. B. in (13) bemerkbar wird; zum anderen ein Zusammenhang einer Familie bekannter robuster Gewichtsfunktionen [6] zur Approximation der Exponentialfunktion über (20); und schließlich ein Zusammenhang einer Kernel-Lifting-Methode [7], in dem die Verlustfunktion ρ_w des IRLS-Algorithmus auf symmetrische Weise mit der Kernelfunktion φ_Ω der Lifting-Methode über die Umkehrfunktion $\Omega(w)$ der Gewichtsfunktion $w(\Omega)$ zusammenhängt.

Des Weiteren wurde mit gleichverteilten Ausreißern eine sehr gut interpretierbare Gewichtsfunktion (27) durchgesprochen, die zwar an mehreren Stellen insbesondere zur anschaulichen Argumentation angeschnitten wird, aber deren Eigenschaften scheinbar nirgends ausführlich behandelt werden.

Literatur

1. W. Förstner and B. Wrobel, *Photogrammetric Computer Vision: Geometry, Orientation and Reconstruction*, ser. Geometry and Computing. Springer International Publishing, 2016.
2. P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.
3. P. W. Holland, "Weighted ridge regression: Combining ridge and robust regression methods," National Bureau of Economic Research, Tech. Rep., 1973.
4. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *International Workshop on Vision Algorithms*, 2000, pp. 298–372.
5. C. Zach and G. Bourmaud, "Descending, lifting or smoothing: Secrets of robust cost optimization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 547–562.
6. J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4331–4339.
7. C. Zach, "Robust bundle adjustment revisited," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 772–787.

The future of machine vision: AI software designed with users in mind

Markus Schatzl[1], Jonas Meier[1], Henning Frechen[2], and Katrin Götzer[3]

[1] senswork GmbH Innovation Lab,
Friedenstr. 18, 81671 München,
[2] Fraunhofer-Institut für Integrierte Schaltungen IIS,
Nordostpark 84, 90411 Nürnberg
[3] eresult GmbH
Friedrichstraße 3-4, 37073 Göttingen

Abstract Machine learning by means of neural networks has developed an indispensable method to solve intricate challenges in optical quality control for manufacturing. When the technology became usable for inline inspection tasks first, neural network architectures themselves were at focus. However, it has become increasingly obvious that the degree of success in implementing vision AI systems is highly dependent on a well-structured and reliable infrastructure. These aspects are commonly summarised under the terms of machine learning operations (MLOps) and human centered design (HCD). Our experiments are conducted using the industrial AI software Neuralyze®, which has served as a basis for several research projects starting in 2019 to test new approaches to machine learning in manufacturing. In our research, we introduce approaches on how to ideally integrate those methods into AI software concepts to derive an optimum benefit. It is a key goal to retain standardized handling semantics despite the variety of model architectures and use cases.

Keywords Machine vision, industrial imaging, image processing, image analysis, machine learning, deep learning, machine learning operations, user centered design, human centered design, context analysis, quality assurance

1 Introduction

As the applied counterpart to computer vision, machine vision has been putting academic knowledge of image processing into practice almost continuously for over 40 years. This is also the case for machine learning based on convolutional neural networks (CNNs), which emerged as a novel approach to analyze camera data in actual applications about a decade ago. The discipline, usually referred as DL/ML (Deep Learning / Machine Learning) developed into a major research topic since then.

The related transfer into practical application however was subject to significant obstacles in its early phase - it's primary reason being the lack of computing power to be of any value in industrial production use cases. This barrier successively lowered by the fast development of GPU hardware and the related increase of GPU power. It also benefited from the growing interest of the scientific community, going along with the implementation of libraries offering abstractions for fundamental mathematical operations as well as transparent access to computing resources from high level programming languages, like Keras [1].

2017 marks a change with the release of Keras 2.0 as a hugely improved toolset to enable easy access for experimentation with CNNs. The top-level-library TensorFlow [2] added additional capabilities to the point of automated image set downloads. It further decreased the threshold to access deep learning technology, also for non-computer scientists. This also marks about the point where desktop GPU power had developed accordingly to enable first machine vision applications, yet still on very small image sizes.

Since then, major model architectures have evolved which focus on image analysis [3]. Their constant development has lead to a number of core applications that have emerged in the process. The central categories comprise classification, object detection and semantic segmentation, with a number of distinctive forms such as anomaly detection or combinations as in instance segmentation. In the majority of cases, the descendants of these architectures are capable of solving even the most complex machine vision problems when used appropriately.

This suggests that in terms of technical feasibility, as of 2024 almost any conventional image analysis task can be solved fast enough for inline processing in industry. This holds even more true as machine

vision also encompasses the entire vertical design of the image acquisition and processing pipeline, and thus also has control over data generation.

However, experience in industry-grade development of those systems shows that the exclusive focus on the technical solution leaves key aspects of the deployment unaddressed. This can lead to poorly performing and and unsustainable vision AI solutions in the field. From a practice-oriented point of view, it becomes apparent that a consistent and well-structured development environment has a crucial impact on the operational success of machine learning systems. Our research seeks to explore ways to standardize these methods and make them more accessible.

2 Related work

Several prior works have already addressed the importance of a unified approach to cope with the complexity of AI applications. In general, the accuracy and performance of ML systems and in particular Vision AI systems depends on three main factors

1. the chosen model type
2. the model implementation
3. as well as the quality of the input data,

which implies a high complexity of these systems [4]. The interdependence of these three factors requires a high level of care already in the development phase with respect to versioning and reproducibility of the entire ML pipeline. During operations (MLOps), the data quality and quality of the models must also be continuously monitored in order to detect malfunctions at an early stage.

One particular challenge is the large landscape of tools for specific tasks, often developed on a small scale by start-ups or communities. The widely different operating paradigms encountered turned out to be an obstacle in constructing seamless workflows. In addition, the market for MLOps software is currently very dynamic due to the permanent release of new solutions. As one example, "Tensorflow Extended" offers a generic platform for the development of ML systems

that maps the complete ML lifecycle ”end-to-end” [5]. However, specially trained personnel such as data scientists, ML engineers and infrastructure teams are required to set up and operate such platforms.

In order to simplify access to ML systems for domain experts without AI expertise, first technical steps are already being taken by developing explanation methods to understand ML model predictions. Yet these methods are still mainly aimed at data scientists. In addition, ”best practices” from classical software development are increasingly being adopted and adapted to increase confidence in the development process [6] [7].

Due to the often probabilistic nature of ML systems, a key factor of good usability, expectation conformance according to ISO 9241-110:2020, is not given. This means that a system does not always behave similarly, and in particular predictably, even in repeated, identical interactions [8]. This complicates user experience (UX) design in the context of AI systems, since different misbehavior in particular cannot be predicted before model implementation is complete. Here, the use-case-specific development of ”AI playbooks” for designers and developers, which collect typical errors in the operation of ML systems can provide a remedy [8]. In addition, a comprehensive meta-study on guidelines for the development and design of AI systems has already summarised initial guidelines for the design of human-AI interaction. The derived 18 core design principles for human-centered design of AI systems are bundled in the Microsoft HAX Toolkit [9]. However, both of the described guidelines have been evaluated only on publicly available AI products for end users, but not yet on ”critical” applications as found in industry.

Finally, to further democratize ML, recent research suggests the notion of ”human-centric machine learning.” AI systems are now conceived as a symbiosis between humans and machines, and a shift in perspective from ”human-in-the-loop” to ”ML-in-the-loop” is called for [10] [11] [12].

3 Methodology

The initial ideas of the methods we target do not originally arise from a machine learning context. They evolved from good practice in ad-

jacent fields, like Development Operations (DevOps) as a practice to unify and streamline all processes that are necessary to manage and build software code. Based on this ideal, MLOps emerged to achieve something analogous for the development of machine learning applications. Human Centered Design (HCD) originated from experimental psychology in the first half of the last century, expanding to a large range of fields since then. [13].

3.1 Machine Learning Operations (MLOps)

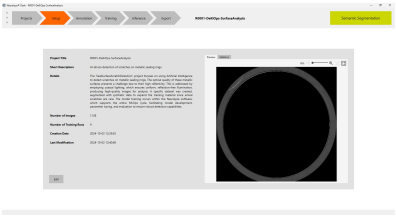
Organizations are confronted with many obstacles when optimizing machine learning systems within their technical lifecycles. Versioning of models, result repeatability, and preserving constant performance across different environments are among the key operational concerns [14]. Cross-functional cooperation, handling a variety of tools, and incorporating ML workflows with current procedures are organizational hurdles [14]. Issues with data quality, resource constraints, and model deployment challenges are the main concerns in industrial settings. We have devised a general method to address these problems in the beforementioned sectors.

We illustrate our efforts with Neuralyze®, a software framework developed by senswork for AI-based image analysis, which puts the above tasks into practice. Figure 1(a) shows the general overview of a project in Neuralyze®. It serves as an informational entry point to provide any user in cross-functional collaboration projects with insights into the development process. This is of high importance for all involved personas of a vision AI project.

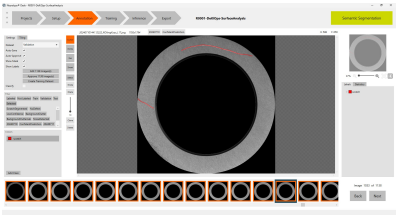
Figure 1(b) shows the data management step in the annotation tab. Users are provided with tools to handle data, which includes data cleaning, sorting, and tagging, gaining insights on the metadata, labeling the data, and finally creating datasets.

The subsequent step in an MLOps cycle is model development. The availability of ready-to-use datasets on sites like Kaggle [15] causes a significant change in focus toward model development. Many academic articles similarly emphasize getting high performance scores on benchmark datasets, with the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) being one of the most well-known examples [16]. Industrial experience has however shown that a more balanced strategy

that incorporates data-centric strategies frequently produces superior long-term results.

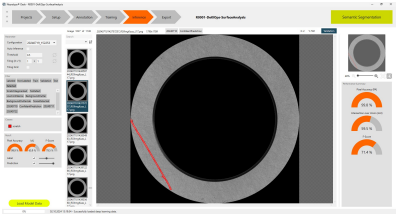


(a) The setup tab gives an overview of a complete AI project.



(b) The annotation tab provides tools to work on data.

(c) The training tab offers facilities to train computer vision models.



(d) The inference tab enables model testing and evaluation based on metrics.

Figure 1: Human centered interface of Neuralyze® Desk.

Neuralyze® follows the paradigm of data-centric AI (DC-AI). Data-centric AI is an emerging paradigm that emphasizes enhancing data quality and quantity to improve AI systems, complementing the traditional model-centric approach [17, 18].

Figure 1(c) shows the training tab in Neuralyze®. In this working area, users can develop a machine vision model based on the dataset created before. Users have the option to select predefined model architectures like ResNet [19]. The focus of this selection is put on model architectures that have proven effective in the field. Combined with a data-centric AI approach, this allows for the efficient and rapid development of models ready for production. Furthermore, the most important hyper-parameters, with default values based on empirical

experience from industrial practice, are accessible to the user. Among them are the input size, batch size, number of epochs, learning rate, and a predefined selection of loss functions.

Figure 1(d) shows how models are evaluated in Neuralyze®. Various performance indicators and error metrics have been proposed for both regression and classification algorithms in engineering and sciences [20]. These metrics are often dependent on the dataset and the specific application of the model [21]. In Neuralyze® we have implemented the most important metrics for this task. These metrics are visualized in Neuralyze® so that users can easily evaluate the trained models based on them.

3.2 Human Centered Design (HCD)

Human-Centered Design (HCD) is an interdisciplinary approach that focuses on optimizing products towards user-friendlyness. As the design and implementation of engineering software is generally profound, and these systems are often highly complex, they require seamless interaction between humans and technology. We will outline the critical role of HCD in creating effective, efficient and satisfying engineering software solutions in relation to the scope of our work.

HCD places users at the center of the design process by iteratively involving them through prototyping, testing, and feedback collection at every stage of development [22]. In engineering software, usability issues can lead to reduced productivity or costly mistakes in high-stakes environments like aerospace, healthcare, and manufacturing [23].

In order to create engineering software with optimum usability, it is necessary to align the design with user needs. This requires knowledge of their characteristics, goals, tasks, environment and resources. The findings are collected by means of a user context analysis. The examination of these findings leads to requirements for the information architecture, system design and interaction design.

The industry-grade systems investigated in the research project represent processes that involve both manual activities, e.g. in production, and pure information work. This results in a wide range of potential requirements. Their identification requires the participation of various groups of stakeholders. Stakeholders in industry (quality assurance, production, technology deployment planning) as well as domain ex-

perts in relation to machine learning and AI applications must be involved.

Both in-depth and contextual interviews were used to collect data that served as the basis for the creation of proto-personas and task models. These easily understandable and communicable artefacts have been iteratively discussed and adapted with the respective stakeholders. For the DeKIOps project, 12 interview partners from both the industrial context and machine vision experts were surveyed.

It went apparent that tasks within the field of data exploration and feature engineering, belonging to the domain of data experimentation are difficult to define and cannot be fully mapped by engineering software. It is likely that these tasks have to be further performed by human experts in the future, using auxiliary tools that are closely tailored to the tasks. For machine learning on image data (vision AI), tasks relating to the creation and continuous, iterative improvement of neural models (training, retraining, monitoring) have been identified as key topics.

4 Discussion

MLOps frameworks are becoming more and more necessary as the complexity of implementing machine learning models in industrial systems increases. This is particularly important in machine vision, where productivity expectations require tasks like segmentation, classification, and object recognition to be improved. In this work, we utilized a prototype platform to show how MLOps concepts, such as automated monitoring and continuous integration/discovery, might simplify model construction for users who are technically inclined but may not be machine learning experts. This technique covers critical difficulties including model versioning, scalability, and performance monitoring [4].

4.1 Findings and Interpretation

Academically trained data scientists have historically been key roles in industrial AI model development. Our prototype, however, seeks to transfer this accountability to users who have received technical voca-

tional training. The system enables such users to iteratively update and upgrade models as new product features or flaws develop. Though it was intended to someday be usable by non-academic users, those with an academic background now dominate the platform.

The platform's user-friendly interface effectively promotes communication and interaction among the various stakeholders in an organization, such as technicians, project managers, sales staff, and even non-technical personnel. This significantly increases the number of people who can enhance, manage, and optimize AI systems without requiring traditional AI development experts like data scientists. This type of cross-disciplinary collaboration is absolutely necessary to ensure that AI systems function reliably under various constraints, such as real-time processing, strict compliance with safety regulations, and scalability for large-scale operations [5].

The integration of Human-Centered Design (HCD) principles ensures that users without deep ML knowledge can interact effectively with the platform via simplified interfaces. This inclusion of HCD ensures that the system not only performs technically but is also usable and efficient for the end-users [22].

4.2 Limitations and Future Work

Even while the platform makes model construction easier, data scientists and machine learning experts are still needed for specialized solutions when dealing with demanding tasks. Furthermore, it is still difficult to define a terminology that unites ML experts and non-experts. Subsequent investigations will concentrate on creating a common lexicon and verifying how successfully non-technical users can utilize the system, finding difficulties they encounter.

An interesting question is whether the methods and processes of HCD can be applied for machine learning and AI systems on a general basis. Analysing the context of use in the DeKIOps project, it became clear that some HCD methods pose new challenges. The extension of HCD towards the scope of machine learning is a field of research to be further explored in the future.

5 Conclusion

In conclusion, scalable and dependable machine learning model deployment in industrial contexts requires the integration of MLOps frameworks, as the Neuralyze® platform demonstrates. Through the prioritization of data-centric AI and the facilitation of cross-functional cooperation, MLOps guarantees that models are resilient, replicable, and condition-adaptive. Simultaneously, an adoption of Human-Centered Design principles improves the platform's usability, making it accessible to both AI professionals and non-experts. The successful use of MLOps and HCD in difficult operational situations will be crucial for industrial AI systems as they develop further.

Acknowledgement

This research is accomplished within the project DeKIOps (AKZ DIK0451). We acknowledge the financial support for the project by the Bavarian Joint Research Programme (BayVFP) of the Free State of Bavaria.

References

1. "Introducing Keras 1.0," 2016, accessed on September 29, 2024. [Online]. Available: https://blog.keras.io/introducing-keras-10.html
2. "Announcing TensorFlow 1.0," 2017, accessed on September 29, 2024. [Online]. Available: https://developers.googleblog.com/en/announcing-tensorflow-10/
3. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Mar 2021.
4. D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Advances in Neural Information Processing Systems. Curran Associates, Inc., 2015.

5. A. N. Modi, C. Y. Koo, C. Y. Foo, C. Mewald, D. M. Baylor, E. Breck, H.-T. Cheng, J. Wilkiewicz, L. Koc, L. Lew, M. A. Zinkevich, M. Wicke, M. Ispir, N. Polyzotis, N. Fiedel, S. E. Haykal, S. Whang, S. Roy, S. Ramesh, V. Jain, X. Zhang, and Z. Haque, "Tfx: A tensorflow-based production-scale machine learning platform," in *KDD 2017*. KDD 2017, 2017.
6. A. Serban, K. van der Blom, H. Hoos, and J. Visser, "Practices for engineering trustworthy machine learning applications," in *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*, 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN). IEEE, May 2021.
7. S. R. Kaminwar, J. Goschenhofer, J. Thomas, I. Thon, and B. Bischl, "Structured verification of machine learning models in industrial settings," *Big Data*, Dec. 2021.
8. M. K. Hong, A. Fourney, D. DeBellis, and S. Amershi, "Planning for natural language failures with the ai playbook," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. ACM, May 2021.
9. S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, and E. Horvitz, "Guidelines for human-ai interaction," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. ACM, May 2019.
10. Y. Liu and H. Shen, "Human-centric machine learning - a human-machine collaboration perspective," München, Tech. Rep., 2021.
11. B. Shneiderman, "Human-centered artificial intelligence: Three fresh ideas," *AIS Transactions on Human-Computer Interaction*, p. 109–124, 2020.
12. M. K. Lee, N. Grgić-Hlača, M. C. Tschantz, R. Binns, A. Weller, M. Carney, and K. Inkpen, "Human-centered approaches to fair and responsible ai," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems. ACM, Apr. 2020.
13. J. Auernhammer, M. Zallio, L. Domingo, and L. Leifer, "Facets of Human-Centered Design: The Evolution of Designing by, with, and for People," in *Design Thinking Research*, ser. Understanding Innovation, C. Meinel and L. Leifer, Eds. Springer, November 2022, pp. 227–245.
14. A. Singla, "Machine Learning Operations (MLOps): Challenges and Strategies," *Journal of Knowledge Learning and Science Technology ISSN:*

2959-6386 (online), vol. 2, no. 3, pp. 333–340, Aug. 2023. [Online]. Available: https://jklst.org/index.php/home/article/view/107

15. "Kaggle documentation," 2024, accessed on September 29, 2024. [Online]. Available: https://www.kaggle.com/docs
16. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
17. J. Jakubik, M. Vössing, N. Kühl, J. Walk, and G. Satzger, "Data-Centric Artificial Intelligence," Jan. 2024, arXiv:2212.11854 [cs]. [Online]. Available: http://arxiv.org/abs/2212.11854
18. A. Majeed and S. O. Hwang, "Towards Unlocking the Hidden Potentials of the Data-Centric AI Paradigm in the Modern Era," *Applied System Innovation*, vol. 7, no. 4, p. 54, Jun. 2024. [Online]. Available: https://www.mdpi.com/2571-5577/7/4/54
19. S. Targ, D. Almeida, and K. Lyman, "Resnet in Resnet: Generalizing Residual Architectures," Mar. 2016, arXiv:1603.08029 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1603.08029
20. M. Z. Naser and A. H. Alavi, "Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences," *Architecture, Structures and Construction*, vol. 3, no. 4, pp. 499–517, Dec. 2023. [Online]. Available: https://link.springer.com/10.1007/s44150-021-00015-8
21. R. S. Tiwari, "Model Evaluation," in *Fundamentals and Methods of Machine and Deep Learning*, 1st ed., P. Singh, Ed. Wiley, Feb. 2022, pp. 33–100. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/9781119821908.ch3
22. J. Hehn and D. Mendez, "Combining design thinking and software requirements engineering to create human-centered software-intensive systems," *arXiv preprint arXiv:2112.05549*, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2112.05549
23. A. Seffah, J. Gulliksen, and M. C. Desmarais, *Human-Centered Software Engineering: Software Engineering Models, Patterns and Architectures for HCI*. Dordrecht: Springer, 2005.

Constrained hand-multiple-eyes calibration

Msuega Jnr. Iorpenda and Volker Willert

Center for Robotics (CERI). Technical University of Applied Sciences,
Würzburg-Schweinfurt,
Konrad-Geiger-Strasse 2, 97421 Schweinfurt.

Abstract This paper addresses the problem of calibrating multiple visual sensors mounted on a robotic manipulator, a task critical for accurate robot perception and interaction. We present a novel approach to hand-multiple-eyes calibration that incorporates closed-loop constraints to ensure consistency between the sensors' poses. Unlike traditional hand-eye calibration methods that handle individual sensor pairs independently, our method leverages a unified optimization framework that simultaneously optimizes the relative poses of all sensors while enforcing a loop closure constraint to each pose triplet. The core of our approach is a least squares approach to solve multiple hand-eye matrix equations of the form $\mathbf{AX} = \mathbf{XB}$, further enhanced with the method of Lagrangian multipliers to account for loop-closure constraints. We apply this idea to a minimal setup involving one hand and two eyes and demonstrate its effectiveness in improving the accuracy of pose estimation for various levels of noisy measurements.

Keywords Hand-eye calibration, multi-sensor-robot calibration, pose-graph optimization, constrained optimization

1 Introduction

The spatial relationship between a robot's end-effector (hand) and its visual sensor (eye) is critical for achieving synchronization in task execution [1]. In certain robotic applications, the use of multiple visual sensors is necessary for robust and reliable estimations, often requiring precise calibration.

A hand-eye calibration outputs the relative pose $\mathbf{X} \in SE(3)$ between a sensor (eye) and the robot end-effector (hand) comprising rotation $\mathbf{R}$ and translation $\mathbf{t}$. Using N pairs of measured pose changes $\{\mathbf{A}_i, \mathbf{B}_i\}_{i=1}^{N}$ and minimizing the nonlinear least squares loss $\mathcal{L}(\mathbf{X}) = \frac{1}{2N}\sum_{i=1}^{N} \|\mathbf{A}_i\mathbf{X} - \mathbf{X}\mathbf{B}_i\|^2$ subject to $\mathbf{X}$ via a gradient descent approach results in a very accurate estimate for the unknown pose $\mathbf{X}$ (see also Fig. 1) outperforming non-iterative classical methods [2]. This idea has been extended to multi-sensor setups comprising K sensors by several authors [3–5] that all share the same basic idea: Simply optimizing the overall loss $\mathcal{L}_K = \sum_{j=1}^{K} \mathcal{L}(\mathbf{X}_j)$, whereas $\{\mathbf{X}_j\}_{j=1}^{K}$ are the fixed relative poses between each possible pair of sensors. This is equivalent to optimizing each pair of sensors separately because the constrained geometric relations between the relative poses are not taken into account. In [4] the constraint of equivalent rotations of the measured poses $\mathbf{A}_i$ and $\mathbf{B}_i$ are considered but no loop-closure constraint on the estimates $\mathbf{X}_j$. There are also solutions that increase accuracy for the special case of hand-cameras calibration by directly optimizing the reprojection error and considering the uncertainties of the different measures using a Gaussian-Helmert model [4, 6].

2 Proposed Method

Our work follows the idea of gradient based nonlinear least squares optimization but includes additional closed loop pose-graph constraints to fulfill physical world reality for the estimates of all relative sensor poses. Here, we explore the minimal multi-sensory setup consisting of two sensors S1 and S2 rigidly attached to the end-effector EE of a serial manipulator, as illustrated in Figure 1. Here, $\{\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i\}_{i=1}^{N}$ are all measured pose changes of the two sensors and the end-effector acquired by moving the robot arm accordingly. This sensory setup results in three unknown relative poses $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$. These three relative poses form a closed pose-graph loop at any time. Hence, we can formulate an additional least squares loop closure constraint $\mathcal{L}_c = \frac{1}{2}\|\mathbf{X}\mathbf{Y} - \mathbf{Z}\|^2 \stackrel{!}{=} 0$ and add it to the hand-multiple-eyes loss $\mathcal{L}_3$ for $K = 3$ frames via a

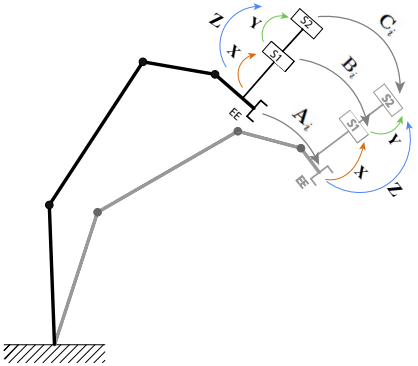


Figure 1: Schematic representation of the configuration of the robot end-effector (EE) and sensors (S1 and S2) for robot-multiple-eyes calibration. The figure illustrates two states of the robot's motion (represented by black and gray outlines) used to measure pose changes $\{\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i\}$. Additionally, the unknown relative poses $\{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$ between the sensors and the end-effector need to form a closed pose-graph loop, where the constraint: $\mathbf{XY} = \mathbf{Z}$ holds in the physical real world.

Lagrangian Multiplier λ as follows: $\mathcal{L} = \mathcal{L}_3 + \lambda \mathcal{L}_c$, which reads

$$\begin{aligned}\mathcal{L} =& \frac{1}{2N} \sum_{i=1}^{N} \left(\|\mathbf{A}_i\mathbf{X} - \mathbf{X}\mathbf{B}_i\|^2 + \|\mathbf{B}_i\mathbf{Y} - \mathbf{Y}\mathbf{C}_i\|^2 + \|\mathbf{A}_i\mathbf{Z} - \mathbf{Z}\mathbf{C}_i\|^2 \right) \\ &+ \frac{1}{2}\lambda \|\mathbf{X}\mathbf{Y} - \mathbf{Z}\|^2, \qquad i \in [1, ..., N]\,. \end{aligned} \tag{1}$$

This objective is optimized with a gradient descent approach applying constrained differential optimization [7] and the angle-axis representation for rotations.

2.1 Optimization of Rotations

We propose an optimization for estimating rotations and translations separately by decoupling the poses of the measurements and estimates from Eq. (1) into rotation matrices and translation vector components. Additionally, we incorporate the closed-loop constraint into the objective function using a Lagrange multiplier, as shown in Eqs. (3), where $\mathcal{L}_{RX}$, $\mathcal{L}_{RY}$, $\mathcal{L}_{RZ}$, and $\mathcal{L}_{RC}$ represent the rotational components of the

objective function corresponding to the transformations **X**, **Y**, **Z**, and the constraint, respectively. The rotation matrices are parameterized in terms of their rotation axes $\mathbf{s} = [s_0, s_1, s_2]^\top$ and denoted as $\mathbf{R}_X(\mathbf{s}_X)$, $\mathbf{R}_Y(\mathbf{s}_Y)$, and $\mathbf{R}_Z(\mathbf{s}_Z)$. The rotation objective reads

$$\mathcal{L}_R(\mathbf{s}_X, \mathbf{s}_Y, \mathbf{s}_Z, \lambda) = \mathcal{L}_{RX}(\mathbf{s}_X) + \mathcal{L}_{RY}(\mathbf{s}_Y) + \mathcal{L}_{RZ}(\mathbf{s}_Z) + \lambda \mathcal{L}_{RC}(\mathbf{s}_X, \mathbf{s}_Y, \mathbf{s}_Z)\,, \tag{2}$$

$$\begin{aligned}\mathcal{L}_R =& \frac{1}{2N} \sum_{i=1}^{N} \Big(\|\mathbf{R}_{Ai}\mathbf{R}_X - \mathbf{R}_X\mathbf{R}_{Bi}\|^2 + \|\mathbf{R}_{Bi}\mathbf{R}_Y - \mathbf{R}_Y\mathbf{R}_{Ci}\|^2 \\ &+ \|\mathbf{R}_{Ai}\mathbf{R}_Z - \mathbf{R}_Z\mathbf{R}_{Ci}\|^2 \Big) + \frac{1}{2}\lambda \|\mathbf{R}_X\mathbf{R}_Y - \mathbf{R}_Z\|^2 \,. \end{aligned} \tag{3}$$

We derive the gradients of the rotational objective function with respect to the axis parameters and the Lagrange multiplier, as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_R}{\partial s_{Xk}} &= \frac{\partial \mathcal{L}_{RX}(\mathbf{R}_X(\mathbf{s}_X))}{\partial s_{Xk}} + \frac{\partial \mathcal{L}_{RC}(\mathbf{R}_X(\mathbf{s}_X), \lambda)}{\partial s_{Xk}} \\ &= \frac{1}{N} \sum_i \left\{ \operatorname{tr} \left(\left[2\mathbf{R}_X - \mathbf{R}_{Ai}^\top \mathbf{R}_X \mathbf{R}_{Bi} - \mathbf{R}_{Ai} \mathbf{R}_X \mathbf{R}_{Bi}^\top \right]^\top \frac{\partial \mathbf{R}_X(\mathbf{s}_X)}{\partial s_{Xk}} \right) \right\} \\ &\quad + \lambda \operatorname{tr} \left(\left[\mathbf{R}_X - \mathbf{R}_Z \mathbf{R}_Y^\top \right]^\top \frac{\partial \mathbf{R}_X(\mathbf{s}_X)}{\partial s_{Xk}} \right) , \end{aligned} \tag{4}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_R}{\partial s_{Yk}} &= \frac{\partial \mathcal{L}_{RY}(\mathbf{R}_Y(\mathbf{s}_Y))}{\partial s_{Yk}} + \frac{\partial \mathcal{L}_{RC}(\mathbf{R}_Y(\mathbf{s}_Y), \lambda)}{\partial s_{Yk}} \\ &= \frac{1}{N} \sum_i \left\{ \operatorname{tr} \left(\left[2\mathbf{R}_Y - \mathbf{R}_{Bi}^\top \mathbf{R}_Y \mathbf{R}_{Ci} - \mathbf{R}_{Bi} \mathbf{R}_Y \mathbf{R}_{Ci}^\top \right]^\top \frac{\partial \mathbf{R}_Y(\mathbf{s}_Y)}{\partial s_{Yk}} \right) \right\} \\ &\quad + \lambda \operatorname{tr} \left(\left[\mathbf{R}_Y - \mathbf{R}_X^\top \mathbf{R}_Z \right]^\top \frac{\partial \mathbf{R}_Y(\mathbf{s}_Y)}{\partial s_{Yk}} \right) , \end{aligned} \tag{5}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_R}{\partial s_{Zk}} &= \frac{\partial \mathcal{L}_{RZ}(\mathbf{R}_Z(\mathbf{s}_Z))}{\partial s_{Zk}} + \frac{\partial \mathcal{L}_{RC}(\mathbf{R}_Z(\mathbf{s}_Z), \lambda)}{\partial s_{Zk}} \\ &= \frac{1}{N}\sum_i \left\{ \operatorname{tr}\left(\left[2\mathbf{R}_Z - \mathbf{R}_{Ai}^\top \mathbf{R}_Z \mathbf{R}_{Ci} - \mathbf{R}_{Ai}\mathbf{R}_Z\mathbf{R}_{Ci}^\top \right]^\top \frac{\partial \mathbf{R}_Z(\mathbf{s}_Z)}{\partial s_{Zk}} \right) \right\} \\ &+ \lambda \operatorname{tr}\left(\left[\mathbf{R}_Z - \mathbf{R}_X\mathbf{R}_Y \right]^\top \frac{\partial \mathbf{R}_Z(\mathbf{s}_Z)}{\partial s_{Zk}} \right), \end{aligned} \tag{6}$$

$$\frac{\partial \mathcal{L}_R}{\partial \lambda} = \frac{1}{2}\|\mathbf{R}_X\mathbf{R}_Y - \mathbf{R}_Z\|^2 = \mathcal{L}_{RC}\,. \tag{7}$$

Compact formulas for partial derivatives of 3D rotation matrices in exponential coordinates can be found in [8]. The gradients were instrumental in the gradient descent optimization, where the update rules for the optimization parameters - rotation axes elements s_{Xk}, s_{Yk} and s_{Zk} for $k \in \{0,1,2\}$ and the Lagrange multiplier λ —are given like follows:

$$s_{Ek}^{i+1} = s_{Ek}^{i} - \alpha \frac{\partial \mathcal{L}_R}{\partial s_{Ek}^{i}} \quad , E \in \{X,Y,Z\}, \quad k \in \{0,1,2\}\,. \tag{8}$$

$$\lambda^{i+1} = \lambda^i + \beta \frac{\partial \mathcal{L}_R}{\partial \lambda^i} = \lambda^i + \beta \frac{1}{2}\|\mathbf{R}_X(\mathbf{s}_X^i)\mathbf{R}_Y(\mathbf{s}_Y^i) - \mathbf{R}_Z(\mathbf{s}_Z^i)\|^2\,. \tag{9}$$

Here, α and β are the step sizes and i representing the iteration index. It should be noted that a gradient descent is performed to find the optimum rotation parameters s_{Ek}, whereas a gradient ascent is performed to find the optimum λ [7].

2.2 Optimization of Translations

Next, we optimize Eq. (1) for the translation vectors $\mathbf{t}_X$, $\mathbf{t}_Y$ and $\mathbf{t}_Z$ assuming the rotations $\mathbf{R}_X$, $\mathbf{R}_Y$ and $\mathbf{R}_Z$ to already been optimal. This leads to the following objective:

$$\mathcal{L}_t = \mathcal{L}_{tX} + \mathcal{L}_{tY} + \mathcal{L}_{tZ} + \lambda_t \mathcal{L}_{tC}\,, \tag{10}$$

$$= \frac{1}{2N}\sum_{i=1}^{N}\Big(\|((\mathbf{R}_{Ai}-\mathbf{I})\mathbf{t}_X-\mathbf{R}_X\mathbf{t}_{Bi}+\mathbf{t}_{Ai})\|^2+\|(\mathbf{R}_{Bi}-\mathbf{I})\mathbf{t}_Y-\mathbf{R}_Y\mathbf{t}_{Ci}+\mathbf{t}_{Bi}\|^2 + \|(\mathbf{R}_{Ai}-\mathbf{I})\mathbf{t}_Z-\mathbf{R}_Z\mathbf{t}_{Ci}+\mathbf{t}_{Ai}\|^2\Big)+\frac{1}{2}\lambda_t\|\mathbf{R}_X\mathbf{t}_Y+\mathbf{t}_X-\mathbf{t}_Z\|^2 . \tag{11}$$

The gradients for the different translation vectors read

$$\frac{\partial\mathcal{L}_{tX}}{\partial\mathbf{t}_X} = \frac{1}{N}\sum_i\Big([\mathbf{R}_{Ai}-\mathbf{I}]^\top[(\mathbf{R}_{Ai}-\mathbf{I})\mathbf{t}_X-\mathbf{R}_X\mathbf{t}_{Bi}+\mathbf{t}_{Ai}]\Big)+\lambda_t(\mathbf{t}_X+\mathbf{R}_X\mathbf{t}_Y-\mathbf{t}_Z), \tag{12}$$

$$\frac{\partial\mathcal{L}_{tY}}{\partial\mathbf{t}_Y} = \frac{1}{N}\sum_i\Big([\mathbf{R}_{Bi}-\mathbf{I})]^\top[(\mathbf{R}_{Bi}-\mathbf{I})\mathbf{t}_Y-\mathbf{R}_Y\mathbf{t}_{Ci}+\mathbf{t}_{Bi}]\Big)+\lambda_t\Big(\mathbf{t}_Y+\mathbf{R}_X^\top[\mathbf{t}_X-\mathbf{t}_Z]\Big), \tag{13}$$

$$\frac{\partial\mathcal{L}_{tZ}}{\partial\mathbf{t}_Z} = \frac{1}{N}\sum_i\Big((\mathbf{R}_{A_i}-\mathbf{I})^\top[(\mathbf{R}_{A_i}-\mathbf{I})\mathbf{t}_Z-\mathbf{R}_Z\mathbf{t}_{Ci}+\mathbf{t}_{Ai}]\Big)+\lambda_t(\mathbf{t}_Z-\mathbf{t}_X-\mathbf{R}_X\mathbf{t}_Y) . \tag{14}$$

These gradients lead to the update rules for the translation parameters with step sizes γ and δ as follows

$$\mathbf{t}_E^{i+1} = \mathbf{t}_E^i - \gamma\frac{\partial\mathcal{L}_t}{\partial\mathbf{t}_E^i}, \quad E\in\{X,Y,Z\}, \tag{15}$$

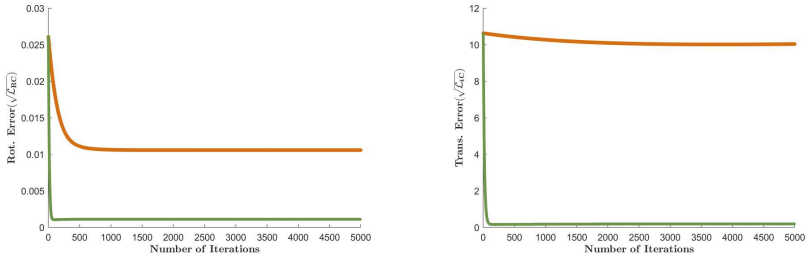
$$\lambda_t^{i+1} = \lambda_t^i + \delta\frac{\partial\mathcal{L}_t}{\partial\lambda_t^i} = \lambda_t^i + \delta\frac{1}{2}\left\|\mathbf{R}_X\mathbf{t}_Y^i+\mathbf{t}_X^i-\mathbf{t}_Z^i\right\|^2 . \tag{16}$$

3 Evaluation

The gradient descent approach derived in Section 2 was implemented in Matlab and tested using synthetic data. Robot and sensor poses were generated using RoboDK [9] and then perturbed with Gaussian noise to simulate real-world conditions. For all experiments the number of measurements is set to $N = 20$, the step sizes are fixed to $\alpha = 0.8$, $\beta = 0.01$, $\gamma = 0.3$ and $\delta = 10^{-7}$ and the number of iterations is $i = 1, \ldots, 5000$. Each optimization run is initialized using the Tsai and Lenz method [10].

3.1 Pose-Graph Loop Closure

Our method enforces a solution for hand-multiple-eyes calibration, achieving a closed-loop pose graph by pushing the closed-loop constraints $\mathcal{L}_{RC}$ and $\mathcal{L}_{\mathbf{t}C}$ close to zero, as shown in Figure 2 (a) and (b) (green lines). In contrast, the unconstrained optimization fails to meet the loop closure constraint (orange lines). The inclusion of constraints also enhances the convergence rate for both rotational and translational errors (green lines), resulting in more precise relative pose estimates (see Fig. 3) that stabilizes at a lower error level.

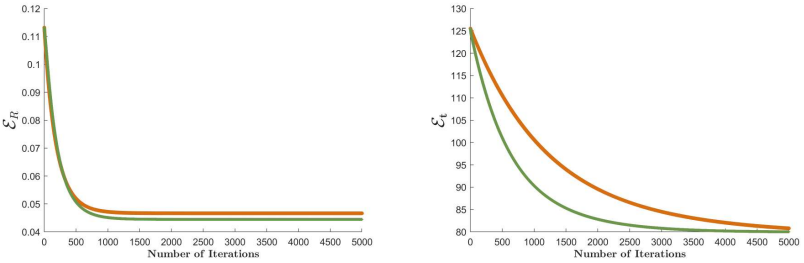


(a) (b)

Figure 2: Rotational and translational errors for constrained (green) and unconstrained (orange) optimization. (a) Error $\sqrt{\mathcal{L}_{RC}}$ when optimizing $\mathcal{L}_R$ including (green) or excluding $\mathcal{L}_{RC}$ (orange). (b) Error $\sqrt{\mathcal{L}_{tC}}$ when optimizing $\mathcal{L}_t$ including (green) or excluding $\mathcal{L}_{tC}$ (orange).

3.2 Improved Rotation and Translation Estimates

We compared the rotation and translation estimates against the ground truth using the metrics $\mathcal{E}_{\mathbf{R}} = \|\mathbf{R}_{est} - \mathbf{R}_{GT}\|$ for rotational deviation and $\mathcal{E}_{\mathbf{t}} = \|\mathbf{t}_{est} - \mathbf{t}_{GT}\|$ for translational deviation. Figure 3 demonstrates the impact of applying the loop-closure constraints during the optimization process by comparing the total rotational and translational errors against ground truth with and without constraints. As can be seen the inclusion of loop-closure constraints not only enforces pose estimates that form a closed pose loop but also enhances the overall calibration accuracy, resulting in a better sensor alignment that is geometrically consistent. In contrast, the unconstrained method achieves less accurate results and does not provide a fully closed pose loop (see also Fig. 2). Additionally, we analyzed the evolution of the accuracy of the relative poses of the sensors during optimization: Sensor 1 with respect to the end effector (**X**), sensor 2 with respect to sensor 1 (**Y**), and sensor 2 with respect to the end effector (**Z**), as shown in Figure 4. Improvements relative to the ground truth were observed across all system components but the improvements vary between the sensors.

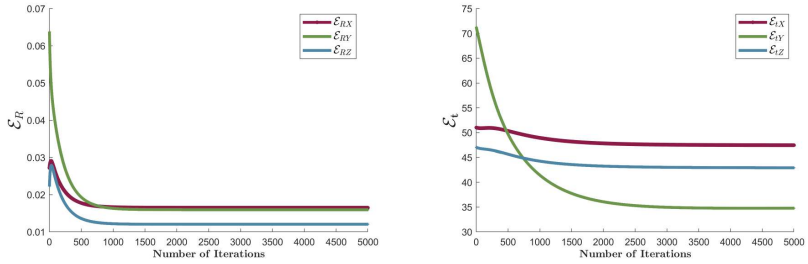


(a) (b)

Figure 3: Rotation and translation estimates versus ground truth during unconstrained (orange) and constraint (green) optimization. (a) Overall rotation errors $\mathcal{E}_{\mathbf{R}}$. (b) Overall translation errors $\mathcal{E}_{\mathbf{t}}$.

3.3 Effect of Noise

We conducted 100 simulation runs per noise level, following [11], with noise sampled from a Gaussian distribution. The histograms in Fig-



(a) (b)

Figure 4: Individual rotation and translation estimates versus ground truth during constraint optimization. (a) Rotation errors $\mathcal{E}_{\mathbf{RX}}, \mathcal{E}_{\mathbf{RY}}$ and $\mathcal{E}_{\mathbf{RZ}}$. (b) Translation errors $\mathcal{E}_{\mathbf{tX}}, \mathcal{E}_{\mathbf{tY}}$ and $\mathcal{E}_{\mathbf{tZ}}$.

ure 5 present the results for four different noise levels (NL1 to NL4). In the rotation error histograms (top row), the noise follows a Gaussian distribution with standard deviations from 0.5 to 2.0 degrees, while in the translation error histograms (bottom row), the noise has standard deviations from 1 to 4 mm, as per [12]. As the noise levels increase (NL1 to NL4), the spread of both rotation and translation errors broadens, indicated by the larger standard deviations (green dashed lines). These results demonstrate the system's sensitivity to increasing noise in both rotation and translation estimates.

4 Summary

We present a new extension to hand-multiple-eyes calibration by adding closed-loop constraints to ensure geometrical consistency between the poses of multiple visual sensors mounted on a robotic manipulator. Unlike traditional hand-eye calibration methods that address sensor pairs independently, our approach simultaneously optimizes the relative poses of all sensors.

First experimental results indicate that the inclusion of closed loop pose-graph constraints in the optimization process leads to estimates that form closed pose loops and each of these estimates are more accurate than if the optimization is done without adding the loop closure constraint. We have experienced that the results and the convergence

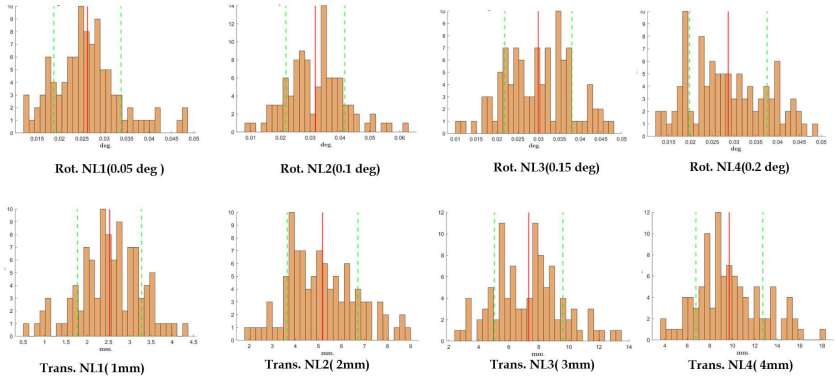


Figure 5: Histograms of rotation errors ($\mathcal{E}_{\mathbf{R}}$, top row) and translation errors ($\mathcal{E}_{\mathbf{t}}$, bottom row) for four different noise levels (NL1 to NL4). The orange bars show the error distribution from 100 simulation runs per noise level. The red vertical lines represent the mean errors, while the green dashed lines mark the standard deviations.

properties strongly depend on the choice of suitable step sizes. Next, an adaptive step size control should be added to take this problem into account. Further on, the dependency on the number of measurements and imbalances in the noise levels between the sensors need to be evaluated.

The Langrangian Multiplier method allows a straight forward extension to a calibration of more than three relative poses. Also a direct optimization of the reprojection error for multiple-camera setups including the loop-closure constraints is some future work to do.

Acknowledgements

This research is co-funded by the Petroleum Technology Development Fund (PTDF) through the OSS Postgraduate Scholarship Scheme. Reference No: PTDF/ED/OSS/PHD/MI/1486/19 - 19PHD058. PTDF Towers, Plot 1058 Memorial Drive, Central Business District, Abuja, Nigeria. Email: info@ptdf.gov.ng.

References

1. B. Grossmann and V. Kruger, "Continuous hand-eye calibration using 3D points," in *15th International Conference on Industrial Informatics (INDIN)*. IEEE, 2017, pp. 311–318.
2. K. M. A. Y. Amy Tabb, "Solving the robot-world hand-eye(s) calibration problem with iterative methods," vol. 28, no. 5, pp. 569–590, 2017.
3. Q. V. Le and A. Y. Ng, "Joint calibration of multiple sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3651–3658.
4. K. Huang and C. Stachniss, "Extrinsic multi-sensor calibration for mobile robots using the gauss-helmert model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1490–1496.
5. D. Zuñiga-Noël, J.-R. Ruiz-Sarmiento, R. Gomez-Ojeda, and J. Gonzalez-Jimenez, "Automatic multi-sensor extrinsic calibration for mobile robots," vol. 4, no. 3, pp. 2862–2869, 2019.
6. M. Ulrich and M. Hillemann, "Uncertainty-aware hand-eye calibration," *IEEE Transactions on Robotics*, 2023.
7. J. C. Platt and A. H. Barr, "Constrained differential optimization," in *Neural Information Processing Systems, Denver, Colorado, USA*, 1987, pp. 612–621.
8. A. Y. Guillermo Gallego, "A compact formula for the derivative of a 3-d rotation in exponential coordinates," *Journal of Mathematical Imaging and Vision*, vol. 51, pp. 378–384, 2015.
9. RoboDK, "Robodk: Robot simulation and offline programming software, version 5.4.3," https://robodk.com, 2023.
10. R. Y. Tsai, R. K. Lenz *et al.*, "A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 345–358, 1989.
11. K. Koide and E. Menegatti, "General hand–eye calibration based on reprojection error minimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1021–1028, 2019.
12. I. Enebuse, B. K. K. Ibrahim, M. Foo, R. S. Matharu, and H. Ahmed, "Accuracy evaluation of hand-eye calibration techniques for vision-guided robots," *Plos one*, vol. 17, no. 10, p. e0273261, 2022.

Machine learning-based battery electrode foil inspection

Ines Müller[2,3], Wenjing Song[1,3], Sebastian Georgi[3], Timo Eckhard[3], and Alexander Korff[2]

[1] University of Applied Sciences Bremerhaven, Embedded Systems Design, Fritz-Erler-Straße 13, 27578 Bremerhaven
[2] University of Applied Sciences Lübeck, Department of Electrical Engineering and Computer Science, Mönkhofer Weg 239, 23562 Lübeck
[3] Chromasens GmbH, Max-Strohmeyer-Str. 116, 78467 Konstanz

Abstract This paper presents an analysis of various autoencoder methods for automated anomaly detection. Prototype image datasets of battery foils, used as anode (copper foil) and cathode (aluminum foil) in lithium-ion batteries, are generated using a line-scan camera system with different illumination setups. The objective is to design and evaluate unsupervised learning methods for surface inspection of the foils. Additionally, the impact of different illumination geometries on the classification performance of the implemented models and their inference times is investigated and analyzed. Another objective is to accelerate model inference by integrating a DPU-based architecture, focusing on optimizing runtime performance for real-time anomaly detection. Using the DPU, an approach achieved a speedup by a factor of 40 compared to computations on the CPU.

Keywords Autoencoder, unsupervised machine learning, anomaly detection, DPU acceleration, hardware acceleration

1 Introduction

The detection of defects in industrial production is crucial as product anomalies can lead to increased costs, delays, and quality issues. In recent years, the production of lithium-ion batteries has significantly expanded due to the rising demand for electronic devices and electric

vehicles. Quality assurance plays a vital role in ensuring that the produced batteries meet performance standards. This includes the quality of the battery electrode foils, the anode, and the cathode, which are later used in batteries. Early detection of anomalies in these foils is essential to identify potential production errors or quality problems. This is where anomaly detection using machine learning methods, such as the autoencoder, comes into play. The autoencoder is a special type of neural network that can be used for unsupervised or semi-supervised anomaly detection [1]. Unsupervised methods are particularly suitable for industrial anomaly detection because labeled defect data are often scarce, expensive, or difficult to obtain.
For this reason, the following study investigates various methods for automated anomaly detection in the context of anode and cathode battery foils. To ensure a comprehensive analysis of autoencoder methods, these will be compared with methods based on similarities between data points extracted from pre-trained neural networks. Furthermore, the implemented methods will be compared with state-of-the-art approaches in industrial anomaly detection, like *Patchcore* [2] and *PaDim* [3].

Another objective includes examining the impact of different lighting conditions on the application-specific properties of the foils. For this purpose, datasets will be created under various lighting conditions, including both defect-free training data and defect anomaly data. The goal is to find a suitable lighting geometry and a method appropriate for the respective applications of the cathode and anode.

As the complexity of machine learning models increases, the demand for computational resources becomes more stringent. Traditional CPU and GPU implementations may struggle to meet the strict real-time processing requirements of industrial applications. Therefore, achieving real-time anomaly detection in industrial environments requires not only effective detection methods but also optimized inference speed to meet operational demands. To address this issue, the study also explores accelerating model inference using DPU-based hardware architectures. By deploying anomaly detection models on a DPU, inference speed and runtime performance can be significantly enhanced, enabling the real-time deployment of complex machine learning models and bridging the gap between advanced detection techniques and practical industrial applications.

2 Materials and Methods

2.1 Data Acquisition and Preprocessing

A line-scan-system (Figure 1) is used to create the datasets for anode and cathode. The foils are illuminated using different geometries: one brightfield and two darkfields. A bright field lighting technique makes reflecting surfaces appear bright since the angle at which the light is incident and the angle at which the camera is aimed are equal. Conversely, dark field illumination involves observing the light that has been dispersed or refracted from the sample. The goal is to identify different types of anomalies.

The line-scan system records image information line by line (8192 pixels per line), and the foils movement over a roller enables the assembly of these lines into a complete surface image. Each line is captured three times, with the different lighting geometries each time. This setup allows capturing the same area under varied lighting conditions, and through a line shift called deinterlacing, the images are separated into three distinct ones for further processing.

Initially, images of undamaged foils are captured to serve as the baseline for training sets. Subsequently, anomalies such as dust, scratches, and moisture are introduced to create test datasets. The influence of the three lighting setups is demonstrated in Figure 2.

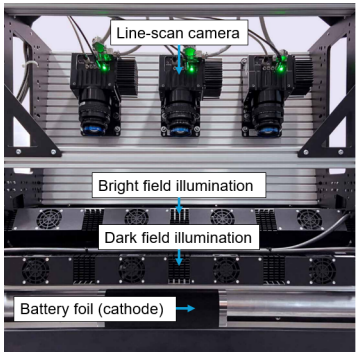
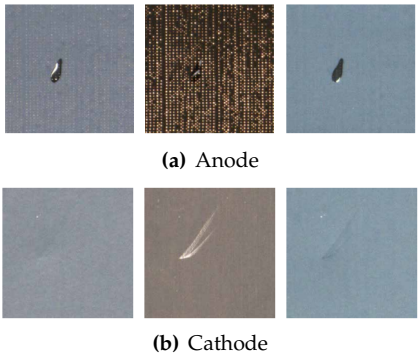


Figure 1: Line-scan-vision platform while scanning the cathode.



(a) Anode

(b) Cathode

Figure 2: Images of same sample material under different illumination geometries (dark field back, bright field, dark field front).

The preprocessing strategy is based on the assumption that different defects are visible under different lighting conditions. Images from each lighting condition are split into patches (256x256x3), transformed into grayscale (256x256x1), and combined into a *multi-flash* image (256x256x3). This combination stores relevant information from each lighting condition in separate color channels, facilitating the recognition of various anomaly types in a single image.

2.2 Solution Approach 1: Reconstruction-Based Methods

To classify the anomalies in the generated datasets, two autoencoder methods were initially tested: Convolutional Autoencoder (CAE) [4] and Variational Autoencoder (VAE) [1]. Autoencoders learn to reconstruct an image from *error-free* data that closely resembles the original. During inference, images with errors are reconstructed by the model as if they had no anomalies. By comparing the original input image with the reconstruction, such as using the *Mean Squared Error* (MSE), anomalies can be classified. For the reconstruction-based methods, *Mean Squared Error* (MSE) and *Structural Similarity Index* (SSIM) are used as classification metrics. *MSE* is widely used and quickly computed, making it suitable for high-speed applications. However, it can be sensitive to noise. *SSIM*, on the other hand, considers bright-

ness, contrast, and structure, providing robustness against noise [5] . Both metrics help determine anomaly scores and set thresholds for binary classification based on F1-score (harmonic mean of presision and recall) maximization.

2.3 Solution Approach 2: Similarity-Based Embedding Methods

This approach involves using pre-trained neural networks (backbones) to extract features from *error-free* training data, forming embeddings that are then reduced using *Principal Component Analysis* (PCA). Classification methods such as *k-Nearest Neighbors* (kNN) and *Kernel Density Estimation* (KDE) compare the similarity of these embeddings to detect anomalies.

ResNet-50 and *MobileNet* are chosen as backbones. *ResNet-50* is suited for extracting complex features and structures in image data, making it ideal for patterned surfaces like the anode foil. *MobileNet* is selected for its efficiency and suitability for resource-constrained environments.

The extracted feature embeddings of the error-free images are stored as vectors after dimensionality reduction. During inference, the Euclidean distance to the *k*-nearest neighbors in the embeddings is calculated. The features of anomalous images are further away from the stored features of error-free images. The mean of the calculated distances then forms the anomaly score for this method.

The choice of these classifiers is motivated by the need for efficiency and the generally low complexity of the image structures involved. These approaches are based on *state-of-the-art* methods such as *Patchcore* [2], *PaDim* [3], and a method from the TKH Group (*TKH-AD*) [4], which are also compared in the evaluation. Figure 3 shows the process of the *Similarity-Based Embeddings* approach. All approaches were implemented in Python using TensorFlow and Keras.

2.4 DPU acceleration Solution

To meet the demanding processing speeds required in the battery foil industry, the Xilinx Deep Learning Processing Unit (DPU) [6] IP core was selected for hardware acceleration. The DPU, integrated into the

[4] `https://www.tkhgroup.com/`

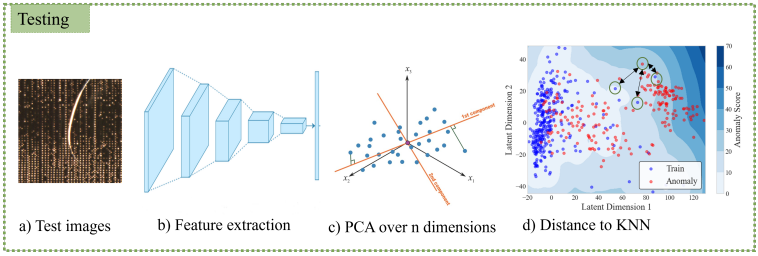


Figure 3: Testing or inference procedure of the similarity-based KNN method.

Xilinx ZCU102 FPGA platform, is designed to accelerate Convolutional Neural Network (CNN) computations using dedicated hardware optimized for parallel processing and high throughput. The architecture is configurable, supporting up to four cores [7], with a maximum of three cores used on the ZCU102 due to resource constraints. Each core independently handles deep learning tasks, maximizing resource utilization through multi-core and multi-threaded processing.

The DPU's specialized instruction set efficiently manages CNN operations such as convolutions and activation functions, making it suitable for real-time applications. Models must be quantized and compiled using Xilinx's Vitis AI tools to optimize them for the DPU, with unsupported operations offloaded to the ARM CPU. This study focused on deploying a quantized Convolutional Autoencoder (CAE) model on different DPU configurations, analyzing the impact of multi-threading on inference speed and how quantization affects the model's accuracy, using the cathode dataset.

3 Results and discussion

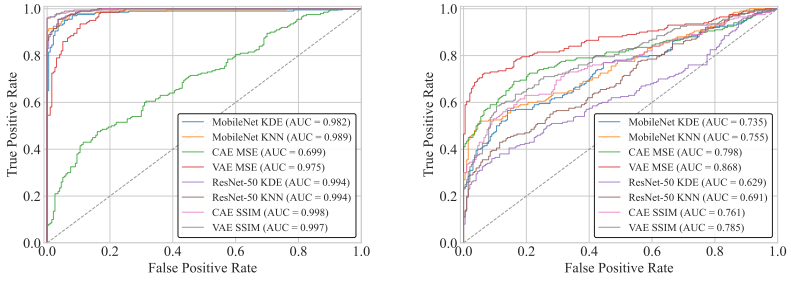
3.1 Performance Evaluation under Different Illumination Geometries

This section presents the evaluation of model performance by analyzing their *Receiver Operating Characteristic* (ROC) curves and *Area Under the Curve* (AUC) values. The *ROC* curve illustrates the true positive rate against the false positive rate at various threshold settings, while the *AUC* value provides a single measure of overall model performance by

quantifying the area under the *ROC* curve. The models were trained with 300 good samples per dataset, and the evaluation metrics were calculated on a test set with 200 good and 200 bad samples. Initially, we evaluated the implemented approaches using combined illumination geometries (*Multi-Flash*).

Figure 4 show the *ROC* curves for the cathode and anode, respectively, under *Multi-Flash* illumination. The *ROC* curves for the anode are significantly lower than those for the cathode. To examine the impact of the individual illuminations, the best models for each illumination category were tested and summarized in Figure 5. The results indicate that the combined illumination (*Multi-Flash*) does not enhance performance for the anode, with the best performance achieved using dark field back illumination alone, where MobileNet with KNN classifier reached 97% *AUC*. For the cathode, multiple approaches achieved 99% *AUC* under both *Multi-Flash* and bright field illumination.

For a comprehensive comparison with *state-of-the-art* methods, Figure 6 presents the AUC values and F1 scores for the anode data under dark field back illumination. With this Dataset the MobileNet KNN approach achieved a slightly higher AUC (97%) compared to *Patchcore* and *PaDim* (both 94%). However, *Patchcore* achieved the best performance under *Multi-Flash* illumination with 88% AUC, while *PaDim* performed best under dark field front illumination with 97% AUC for the anode data.



(a) Cathode **(b)** Anode

Figure 4: ROC-Curves with combined *Multi-Flash* images.

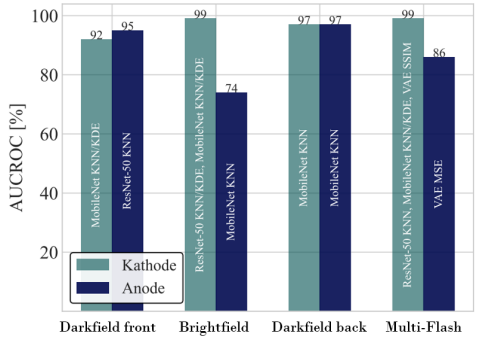


Figure 5: Comparison of the best approaches per illumination geometry.

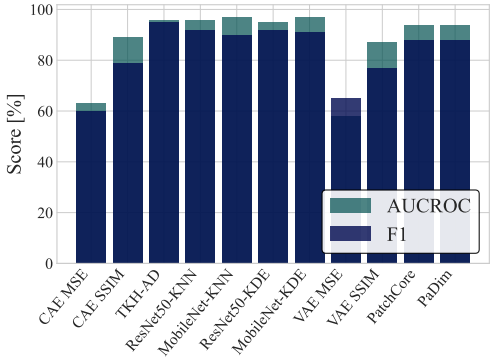


Figure 6: Comparison of the classification performance for the anode in the dark field back with *state-of-the-art* methods.

3.2 Speed evaluation

The inference speed (time for predicting, if one patch is normal or anormal) of the implemented methods is measured across the entire test dataset. The measurements were taken on a *NVIDIA GeForce RTX 3090* GPU and averaged over all test samples to relate speed to classification performance (AUC). The best time was achieved by the CAE and corresponds to a line rate of 0.076 kHz. The measurements in-

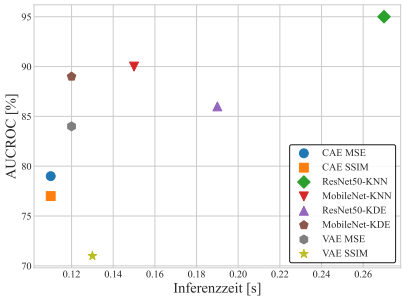


Figure 7: Inference speed of the implemented models at the anode in the darkfield front.

clude the predictions or reconstructions made by the autoencoder and the feature extraction from the pre-trained networks (calculated on the GPU), as well as the computation of the post-processing on the CPU (Intel Core i7).

3.3 Performation Evaluation of Hardware acceleration

Figure 8 is a performance comparison line chart that illustrates the time efficiency of three different configurations, labeled as *1DPU*, *2DPU*, and *3DPU*, across various thread counts while processing a single frame. In studying the impact of DPU core count and thread count on acceleration performance, it was found that performance improvements are not linear as the number of DPU cores and threads increases. When the thread count reaches a certain level, the performance of a single DPU core tends to saturate, and adding more threads may actually lead to a decline in performance. In multi-core configurations, although increasing the number of cores can enhance performance, the complexity of coordinating multiple cores and resource contention limits the extent of these improvements.

For the CAE model used in this study, the optimal configuration, identified through optimization analysis, is a combination of two DPU cores with four threads. Under this configuration, model inference achieved a line rate of 2.65 KHz (32 Patches). This result demonstrates the significant performance improvement in model inference within a

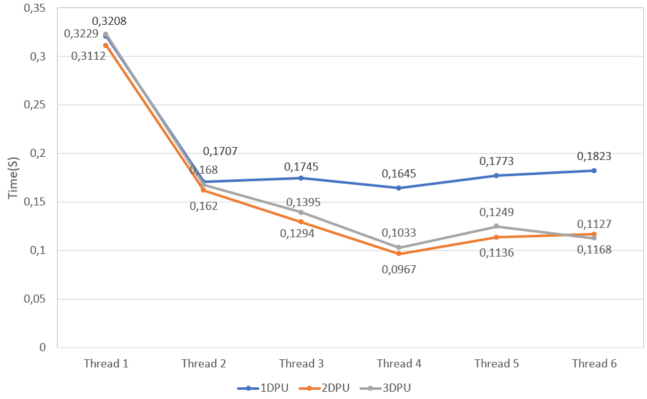


Figure 8: Thread-based performance analysis of single and multi-DPU Configurations.

DPU-accelerated environment.

Quantization was found to degrade the reconstruction accuracy of the models, particularly for MSE-based approaches due to their sensitivity to pixel-level variations, whereas SSIM-based models maintained greater robustness, demonstrating better tolerance to the effects of reduced precision.

4 Conclusions and outlook

This work developed and evaluated unsupervised machine learning methods for detecting anomalies in battery foils under various lighting conditions. Additionally, by using Xilinx's DPU IP core and Vitis AI tools for hardware acceleration, we achieved significant improvements in the model's speed and efficiency. This highlights the benefits of FPGA-based solutions for industrial applications that require fast and power-efficient performance. The investigations showed that combining different illumination geometries into a single image proved effective for the cathode, while the anode did not benefit from this approach. The best results for the anode were achieved using only the dark field back illumination. Here, the best approach (*MobileNet-KNN*) delivered slightly better results compared to established *state-of-the-art*

methods such as *PatchCore* and *PaDim*. For datasets containing more structure (anode, multi-flash), *PatchCore* achieved higher results compared to MobileNet-KNN.

To obtain the most realistic results from the models, annotation by an expert would be necessary. Furthermore, saturation of *AUC* values was observed for several approaches in the cathode datasets. A future approach would be to generate and annotate datasets with even more subtle anomalies to better compare the approaches.

The findings have also demonstrated the significant improvements in processing speed and efficiency afforded by DPU acceleration, making these systems suitable for scenarios where rapid data analysis is critical.

For future advancements, focusing on further reducing latency in data processing and optimizing the entire computational pipeline will be crucial. This includes not only enhancing the model inference stages but also streamlining data input/output operations, preprocessing, and postprocessing. Real-time applications often involve continuous data streams, necessitating systems that can maintain high processing speeds without bottlenecks.

The concept of *Whole Application Acceleration*(WAA) is particularly promising. Considering the substantial improvements in processing times and efficiency achieved through DPU acceleration in this study, future research could further expand the scope of acceleration. By employing FPGA or *High-Level Synthesis* (HLS) not only for model inference but also for preprocessing and postprocessing, the entire computational pipeline, from data acquisition to final output, could benefit from hardware acceleration. Implementing WAA would lead to a more comprehensive utilization of FPGA capabilities, minimizing CPU dependencies and alleviating the bottlenecks observed in the current setup.

Acknowledgements

We acknowledge Chromasens GmbH for providing measurement equipment and technical support.

References

1. D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
2. K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," 2022.
3. T. Defard, A. Setkov, A. Loesch, and R. Audigier, "Padim: a patch distribution modeling framework for anomaly detection and localization," 2020.
4. J. K. Chow, Z. Su, J. Wu, P. S. Tan, X. Mao, and Y. H. Wang, "Anomaly detection of defects on concrete structures with the convolutional autoencoder," *Advanced Engineering Informatics*, vol. 45, p. 101105, 2020.
5. P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," pp. 372–380, 2019. [Online]. Available: http://arxiv.org/pdf/1807.02011v3
6. Xilinx, "Dpu ip details and system integration," https://xilinx.github.io/Vitis-AI/3.0/html/docs/workflow-system-integration.html?highlight=thread, 2020, last Accessed: 2023-12-03.
7. Xilinx, "Dpuczdx8g introduction," https://docs.xilinx.com/r/4.0-English/pg338-dpu/Introduction?tocId=Bd4R4bhnWgMYE6wUISXDLw, June 2022, last Accessed: 2024-01-10.

Deep learning-based localisation of combine harvester components in thermal images

Hanna Senke[1,2], Dennis Sprute[2], Ulrich Büker[3], and Holger Flatt[2]

[1] University of Applied Sciences and Arts Ostwestfalen-Lippe
Campusallee 12, 32657 Lemgo, Germany
[2] Fraunhofer IOSB, Industrial Automation Branch (IOSB-INA),
Campusallee 1, 32657 Lemgo, Germany
[3] University of Applied Sciences and Arts Ostwestfalen-Lippe,
Institute Industrial IT (inIT),
Campusallee 6, 32657 Lemgo, Germany

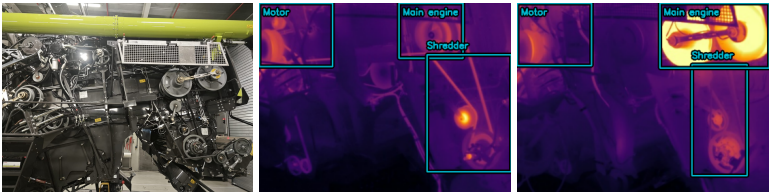
Abstract It is crucial to identify defective machine components in production to ensure quality. Some components generate heat when defective, so automating the inspection process with a thermal imaging camera can provide qualitative measurements. This work aims to use computer vision methods to locate these components in thermal images. Since there is currently no comparison of object detection and semantic segmentation algorithms for this use case, this study compares different architectures with the goal of localising these components for further defect inspection. Moreover, as there are currently no datasets for this use case, this study contributes a novel annotated dataset of thermal images of combine harvester components. The different algorithms are evaluated based on the quality of their predictions and their suitability for further defect inspection. As semantic segmentation and object detection cannot be directly compared with each other, custom weighted metrics are used. The architectures evaluated include RetinaNet, YOLOV8 Detector, DeepLabV3+, and SegFormer. Based on the experimental results, semantic segmentation outperforms object detection regarding the use case, and the SegFormer architecture achieves the best results with a weighted MeanIOU of 0.853.

Keywords Thermal images, object localisation, deep learning architectures, industrial quality assurance

1 Introduction

Identifying defective components in production is crucial for quality management. Some components generate heat when defective and can be identified by this. Currently, this is either done manually or not done at all. An automatic inspection using a thermal imaging camera that captures temperature in a 2D image could enable objective and reproducible measurements, improving quality and supporting workers. To achieve this, the location of each component in the thermal image must be known. A naive approach is to use fixed areas. However, in modern production lines, there are often different machine variants with changing layouts, and some components can be very close to each other or even overlap. Thus, this simple approach does not provide the accuracy needed to evaluate components separately. To address this issue, the components have to be localised in each image individually based on computer vision algorithms, such as those from the fields of object detection and semantic segmentation.

Therefore, in this work, different object detection and semantic segmentation architectures are compared in an industrial production use case, specifically the localisation of combine harvester components during assembly as illustrated in Fig. 1. This use case is chosen due to the high number of product variants and component layouts. A main contribution of this work is a novel annotated dataset of thermal images of combine harvester components intended for object detection and semantic segmentation tasks. Moreover, this work provides a compre-



(a) Color image **(b)** Thermal image of a non-defective machine **(c)** Thermal image of a defective main engine

Figure 1: Images of combine harvesters in side view in different variants with the relevant components motor, main engine and shredder.

hensive performance evaluation of different object detection and semantic segmentation architectures on this novel dataset with the objective of localising the components for further defect inspection.

The remainder of this paper is structured as follows: First, existing approaches related to the topic and the architectures and backbones used for this study are presented. Then, the image acquisition and dataset generation process, the custom metric and the experimental setup are covered. Finally, the results of the comparison are discussed.

2 Related Work

There are already concepts for defect detection on thermal images [1], [2], [3]. However, most of these approaches localise the defect based on thermal information instead of localising the objects first. To ensure that each component is inspected separately, it is necessary to localise the components first. For object localisation on thermal images using object detection or semantic segmentation, there are already existing approaches, e.g. Mukherjee et al. [4] compare different versions of YOLO to localise humans and objects in disaster scenes. Ulhaq et al. [5] optimize YOLO to detect small objects for animal detection in thermal images. Moreover, Ippalapally et al. [6] detect objects for autonomous vehicles in the FLIR dataset, and Li et al. [7] propose a new architecture for semantic segmentation on thermal images. However, none of these approaches match this specific use case involving machine components of combine harvesters, which is especially challenging due the wide range of variants and component layouts.

There are also approaches that localise objects with the intention of further inspection. For example, Gong et al. [8] use YOLO as a base to localise electrical equipment and detect rotation to create a fitting area. Madura Meenakshi et al. [9] localise the eye region using YOLOV2 on infrared thermal images, while Kakileti et al. [10] use convolutional and deconvolutional neural networks on greyscale thermal images to segment areas for breast cancer detection. However, these works do not compare different object detection or semantic segmentation methods on thermal images with a focus on a subsequent inspection, which is necessary to select an optimal neural network architecture for component localisation.

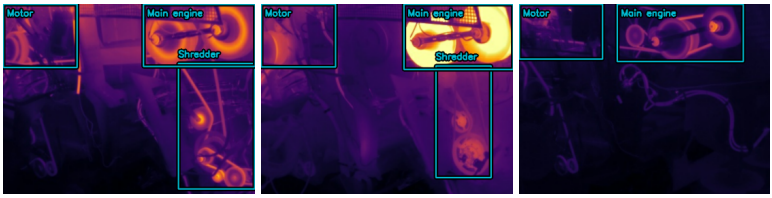
3 Architectures and Backbones

To compare different object detection and semantic segmentation methods, four state-of-the-art neural network architectures are selected. SegFormer [11] is a transformer-based architecture for semantic segmentation that uses mix transformer (MiT) backbones. It can be compared to the DeepLabV3+ [12] architecture, which is also designed for semantic segmentation. DeepLabV3+ is a deep convolutional neural network (DCNN) and will be evaluated with common backbones, namely MobileNetV3, EfficientNetV2, ResNet, ResNetV2, DenseNet, and the YOLOV8 backbone. YOLO models are commonly used in object detection applications. For this study, the YOLOV8 Detector [13] is used and combined with the YOLOV8 backbones. It will be compared with RetinaNet [14], a popular one-stage object detection architecture that uses the same backbones as DeepLabV3+.

4 Image Acquisition, Preprocessing and Dataset

For the dataset of combine harvester components, thermal images were collected over 49 production days. The thermal camera captured an image every 10 to 20 seconds with a resolution of 382 by 288 pixels. Per day, there are five to ten measurement cycles, with each cycle testing one combine harvester. The data contains both defective and non-defective machines, with non-defective machines being more prevalent. The relevant components captured are the motor in the top left corner, the main engine in the top right and the shredder on the right side as illustrated Fig. 2 . The temperature ranges from 30 °C to 60 °C for non-defective combine harvesters (see Fig. 2(a)) and from 30 °C to 125 °C for defective machines (see Fig. 2(b)). Due to the large number of variants, there are also machines without a shredder as depicted in Fig. 2(c). In total, this dataset comprises 19 different machine variants.

The acquired thermal images are first converted to RGB images to utilize common neural network architectures designed for colour images and pre-trained weights of large-scale image datasets. The temperature is clipped at 55 °C to account for inconsistent colouring due to higher temperatures in defective machines. To create the dataset, the measured data is split into measurement cycles. Then, three images per



(a) non-defective machine **(b)** defective main engine **(c)** without shredder

Figure 2: Thermal images of different combine harvester variants.

cycle with a temperature above 45 °C are randomly chosen. This ensures that each machine variant is present in the dataset and different temperatures are represented. As there are significantly fewer images of defective combine harvesters, images with a temperature of 70 °C or above are added to the dataset. Segmentation masks and bounding boxes are then manually added to the data. The final dataset consists of 1200 images, including 253 images of defective machines and 69 images of machines with only two components. This dataset is split into 720 training images and 240 validation and test images each.

The thermal images of defective combine harvesters have some differences in colouring and contrast. For the machines with two components, there is only a small number of images. To identify weaknesses in the models and highlight missing training data, additional datasets are needed. Separate test datasets for defective machines (DM) and non-defective machines (NDM), each containing 250 images, and a test dataset for machines with two components (TCM), containing 69 images, are created from images of the original dataset.

5 Custom Metric

For comparison of the performance of object detection models among each other and semantic segmentation models among each other, the Mean Intersection over Union (MeanIOU) is used. However, for object detection, MeanIOU does not provide a direct comparison for our specific use case, as the bounding boxes cannot accurately represent the components. To address this issue, the ground truth segmentation

masks are used for evaluation. However, bounding box predictions naturally include pixels that do not belong to the component. To ensure a fair comparison, non-heat generating parts of the machine, which do not pose a problem for defect inspection, need to be weighted differently compared to heat-generating parts.

For this purpose, a temperature threshold value $\tau = 70\ ^{\circ}\mathrm{C}$ is defined, which is specific to the components in this use case. The number of false negative pixels is denoted as FN. The false positive pixels are split into two groups using the temperature threshold τ. The number of pixels falsely classified as belonging to the component and with a temperature above the threshold τ is denoted as $FP_{t\geq\tau}$. The number of pixels falsely classified as belonging to the component with a temperature below the threshold τ is denoted as $FP_{t<\tau}$. Each group is given a separate weight. For one component, the weighted absolute error is defined as follows:

$$wAE = \lambda_1 \cdot FP_{t<\tau} + \lambda_2 \cdot FP_{t\geq\tau} + \lambda_3 \cdot FN \tag{1}$$

As mentioned before, non-heat-generating parts of the machine do not pose a problem but are often included in the prediction of object detection models. Therefore, λ_1 should be much smaller than the other weights. For this study, the false positives under the temperature threshold will be weighted with $\lambda_1 = 0.1$. Since the false positives over the temperature threshold τ and the false negatives influence the results of the final defect inspection, they will be weighted with $\lambda_2 = 1$ and $\lambda_3 = 1$.

The intersection over union (IOU) is a common metric to evaluate object detection and semantic segmentation, but it is not well suited for comparing predicted bounding boxes with ground truth segmentation masks. As the wAE evaluates the FP and FN, a weighted union wU can be calculated as the sum of the intersection and the wAE. For the weighted MeanIOU, the arithmetic mean over all components is calculated, with n representing the number of components. We define the weighted IOU and the weighted MeanIOU as follows:

$$wIOU = \frac{I}{wU} = \frac{TP}{TP + wAE} \tag{2}$$

$$wMeanIOU = \frac{1}{n} \sum_{i=1}^{n} wIOU_i \tag{3}$$

The precision can be weighted using the same concept as the other metrics, grouping the FP based on their temperature. Therefore, it is defined as follows:

$$wPrecision = \frac{TP}{TP + \lambda_1 \cdot FP_{t<\tau} + \lambda_2 \cdot FP_{t\geq\tau}} \tag{4}$$

Since $\lambda_3 = 1$, the regular recall does not need to be modified. As with the previous metrics, the arithmetic mean over all components is calculated for recall and weighted precision.

6 Experimental Setup

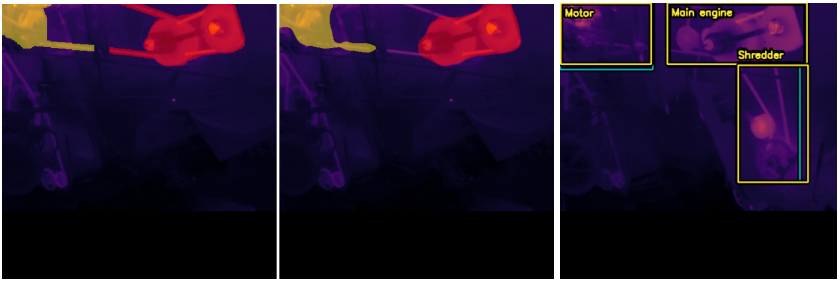
The experiments are conducted on a computer with Windows 11, equipped with a 12th Gen Intel Core i7 processor running at a base speed of 3.60 GHz, 32 GB of RAM, and an NVIDIA GeForce RTX 3060 graphics card with 12 GB of VRAM. The implementation is based on TensorFlow (2.16.1), Keras (3.0.5) and KerasCV (0.8.2).

All models are trained for 150 epochs using pre-trained weights from ImageNet or COCO dataset. At the end, the best weights based on the validation dataset are restored. The stochastic gradient descent (SGD) optimizer is used, with a global clipnorm of 10 and an exponential decay learning rate scheduler starting with a learning rate of 0.001. For each epoch, the model is trained on 360 images, which is half of the training dataset, and evaluated on the validation dataset. From each of the mentioned backbone types for DeepLabV3+ and RetinaNet in Sect. 3, one backbone, preferably of medium size, is selected. The YOLOV8 Detector and SegFormer are trained on all feasible backbones.

After training, the models are evaluated on the test dataset. Since the final goal is to classify objects as defective or non-defective, the most important metrics for the comparison are the weighted MeanIOU, weighted Recall, and weighted Precision. The MeanIOU is very suitable for comparing the performance of object detection and semantic segmentation models among each other.

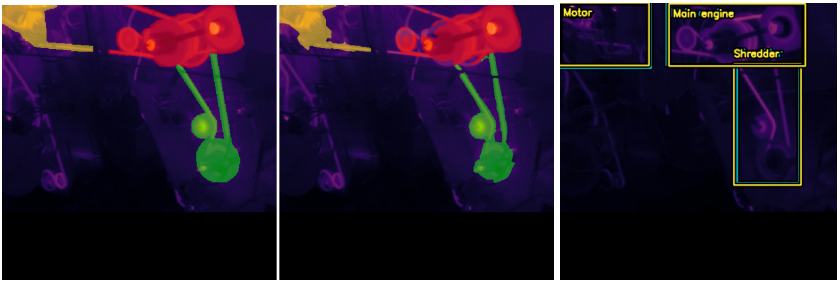
7 Results

The results of the three models with the best MeanIOU from each architecture are presented in Tab. 1, and an example prediction of the best model from each architecture is shown in Fig. 3. The DeepLabV3+



(a) SegFormer,MeanIOU = 0.805, wMeanIOU = 0.870

(b) RetinaNet, MeanIOU = 0.915, wMeanIOU = 0.56



(c) DeepLabV3+,MeanIOU = 0.805, wMeanIOU = 0.850

(d) YOLOV8 Detector, MeanIOU = 0.944, wMeanIOU = 0.589

Figure 3: Example predictions of the best models from each architecture. Ground truth on the left and prediction on the right for (a) and (c). Blue and yellow boxes represent the ground truth and model predictions, respectively, and the model's confidence score is visualized next to the bounding boxes for (b) and (d).

architecture achieves the best MeanIOU with the YOLOV8 M backbone and the second-best MeanIOU with the large MobileNetV3 backbone. For RetinaNet, the YOLOV8 M backbone achieves the best re-

sults, and the large MobileNetV3 reaches the second-best results. For the YOLOV8 Detector models, the medium-sized backbone achieves the best results.

Table 1: Results of the three best models from each architecture on the test dataset.

Architecture	Backbone	Pre-trained weights	Mean-IOU	wMean-IOU	Re-call	wPre-cision
SegFormer	**MiT B0**	ImageNet	**0.8**	**0.853**	**0.861**	**0.988**
DeepLabV3+	DenseNet169	ImageNet	0.755	0.804	0.812	0.971
	MobileNetV3 large	ImageNet	0.76	0.807	0.815	0.962
	YOLOV8 M	COCO	**0.803**	**0.838**	**0.844**	**0.99**
RetinaNet	DenseNet169	ImageNet	0.879	0.448	0.498	0.472
	MobileNetV3 large	ImageNet	0.901	0.513	0.579	0.534
	YOLOV8 M	COCO	**0.914**	**0.682**	**0.766**	**0.704**
YOLOV8 D.	YOLOV8 XL	COCO	0.936	0.631	0.707	0.651
	YOLOV8 XS	COCO	0.939	0.644	0.726	0.662
	YOLOV8 M	COCO	**0.942**	**0.686**	**0.771**	**0.704**

Compared by the MeanIOU, DeepLabV3+ with the YOLOV8 M backbone pre-trained on COCO performs the best for semantic segmentation. The second best performs SegFormer with the MIT-B0 backbone pre-trained on ImageNet. For the weighted MeanIOU, the SegFormer model performs better than the DeepLabV3 model. As the MeanIOU and the weighted MeanIOU are similar, there seems to be no significant difference between a transformer-based architecture and a DCNN for this use case. It is noticeable that colder parts of the component are not always detected, resulting in missing parts of predicted components. For object detection, compared by the MeanIOU, the YOLOV8 Detector performs better than the RetinaNet architecture. For the weighted MeanIOU both models perform similar. Both models use the YOLOV8 M backbone, pre-trained on the COCO dataset. The architectures sometimes miss components, especially in images with only two components.

Compared with the semantic segmentation models, the object detection models perform better for the MeanIOU. However, the semantic segmentation models perform better for the weighted MeanIOU.

Table 2: Models with the best MeanIOU from each architecture tested on the additional test datasets: non-defective machines (NDM), defective machines (DM) and machines with two components (TCM).

Architecture	Backbone	weighted MeanIOU			
		All	**NDM**	**DM**	**TCM**
RetinaNet	YOLOV8 M	0.682	0.65	0.529	0.793
YOLOV8 D.	YOLOV8 M	0.686	0.721	0.778	0.639
DeepLabV3+	YOLOV8 M	0.838	**0.898**	**0.904**	0.753
SegFormer	MiT B0	**0.853**	0.888	0.864	**0.871**

The semantic segmentation model with the best weighted MeanIOU, SegFormer with the MiT-B0 backbone, achieves a weighted MeanIOU of 0.853, while the best object detection model, the YOLOV8 Detector with the YOLOV8 M backbone, only reaches a weighted MeanIOU of 0.686. It is interesting to note that the best models from DeepLabV3+, YOLOV8 Detector, and RetinaNet all use the YOLOV8 M backbone pre-trained on the COCO dataset.

The results of the best model from each architecture on the additional test datasets can be seen in Tab. 2. For the additional test datasets, the YOLOV8 Detector and DeepLabV3+ perform worse on images with only two components than on images of defective machines. In contrast, RetinaNet and SegFormer perform better on images with two components than on images of defective machines.

8 Conclusions and Future Work

This study aimed to compare different object detection and semantic segmentation models with the objective of localising machine components in thermal images for further defect inspection. For this purpose, the specific use case of combine harvester components coming in a wide range of variants and layouts was selected. Based on the evaluation, semantic segmentation models provide the best results for the weighted MeanIOU, and the SegFormer architecture with MiT-B0 backbone achieves the best results. For the object detection architectures, the YOLOV8 M backbone performed best.

Additionally, the results show that the novel dataset presented challenges for the models. For images of defective machines, the colouring

differs from that of non-defective machines, resulting in less accurate predictions on these images. Additionally, images of machines with only two of the three components posed a problem. Both groups require better representation in the training dataset to address this issue. Overall, the dataset is quite small with 1200 images and could benefit from more data from additional measurement cycles. To overcome this problem, additional images could be artificially generated. This could be a promising area for future research, such as using stable diffusion techniques. For further investigation, it would be valuable to assess the performance of the models on data from non-harvesting machinery with different characteristics. Another potential study could explore the use of thermal data directly as a one-channel image with a modified architecture instead of converting it to an RGB image. Overall, this topic has a lot of potential for application in industrial production and quality assurance.

Acknowledgements

This research work is based on "Datenfabrik.NRW", a flagship project by "KI.NRW", funded by the Ministry for Economics, Innovation, Digitisation and Energy of the State of North Rhine-Westphalia (MWIDE). We like to thank CLAAS for supporting the image data acquisition.

References

1. R. Ali and Y.-J. Cha, "Subsurface damage detection of a steel bridge using deep learning and uncooled micro-bolometer," *Construction and Building Materials*, vol. 226, pp. 376–387, November 2019.
2. J. Hu, W. Xu, B. Gao, G. Y. Tian, Y. Wang, Y. Wu, Y. Yin, and J. Chen, "Pattern deep region learning for crack detection in thermography diagnosis system," *Metals*, vol. 8, no. 8, p. 612, June 2018.
3. Y. Laib dit Leksir, M. Mansour, and A. Moussaoui, "Localization of thermal anomalies in electrical equipment using infrared thermography and support vector machine," *Infrared Physics & Technology*, vol. 89, pp. 120–128, March 2018.
4. S. Mukherjee, O. Coudert, and C. Beard, "UNIMODAL: UAV-aided infrared imaging based object detection and localization for search and dis-

aster recovery," in *2022 Virtual IEEE International Symposium on Technologies for Homeland Security*, 2022, pp. 1–6.

5. A. Ulhaq, P. Adams, T. E. Cox, A. Khan, T. Low, and M. Paul, "Automated detection of animals in low-resolution airborne thermal imagery," *Remote Sensing*, vol. 13, no. 16, p. 3276, June 2021.
6. R. Ippalapally, S. H. Mudumba, M. Adkay, and N. V. H. R., "Object detection using thermal imaging," in *2020 IEEE 17th India Council International Conference (INDICON)*, 2020, pp. 1–6.
7. C. Li, W. Xia, Y. Yan, B. Luo, and J. Tang, "Segmenting objects in day and night: Edge-conditioned cnn for thermal image semantic segmentation," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 7, pp. 3069–3082, July 2021.
8. X. Gong, Q. Yao, M. Wang, and Y. Lin, "A deep learning approach for oriented electrical equipment detection in thermal images," *IEEE Access*, vol. 6, pp. 41 590–41 597, July 2018.
9. R. Madura Meenakshi, N. Padmapriya, N. Venkateswaran, R. Ravikumar, and C. Ramya, "Localization of eye region in infrared thermal images using deep neural network," in *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2021, pp. 446–450.
10. S. T. Kakileti, G. Manjunath, and H. J. Madhu, "Cascaded CNN for view independent breast segmentation in thermal images," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, vol. 2019, pp. 6294–6297, 2019.
11. E. Xie *et al.*, "SegFormer: Simple and efficient design for semantic segmentation with transformers," May 2021. [Online]. Available: http://arxiv.org/pdf/2105.15203
12. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," February 2018. [Online]. Available: http://arxiv.org/pdf/1802.02611
13. "YOLOv8: A new state-of-the-art computer vision model," 24.05.2024. [Online]. Available: https://yolov8.com/
14. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 2, pp. 318–327, February 2020.

Noise analysis of a synthetically rendered scene in sensor-realistic image simulation

Christian Kludt[1], Frederik Seiler[2], Verena Eichinger[2], Johannes Meyer[1], Ira Effenberger[2], Thomas Längle[1], and Jürgen Beyerer[1]

[1] Fraunhofer IOSB, Fraunhoferstraße 1, 76131 Karlsruhe
[2] Fraunhofer IPA, Nobelstraße 12, 70569 Stuttgart

Abstract This paper investigates how the noise characteristics of synthetically generated camera images correspond to those of a real camera. We determine the photon transfer curve from a set of rendered images of a static scene. Furthermore, we present a method to identify the regions with high temporal noise, i.e., rendering noise, in synthetically generated data from a single rendered image. Finally, we present a strategy on how a parameterization of the rendering can be achieved that minimizes the noise while also minimizing the rendering time.

Keywords Synthetic data generation, sensor-realistic simulation, noise analysis, EMVA 1288

1 Introduction

The advances in detecting and classifying defects that we might expect from machine learning (ML) approaches have often been stymied by lack of data. To train the AI models, they would need to be fed with a large number of examples of good products, but also supplied with precisely labeled bad ones. There are simply not enough of those, if any at all, which is why we turned the focus of our efforts on synthetic image generation as it has been performed in various visual inspection applications [1–8]. The idea is to simulate the entire testing and inspection environment – specimen geometry, material properties, lighting, sensor technology – to produce images that are synthetic, but still sufficiently realistic. And then we can use data of defects that we have

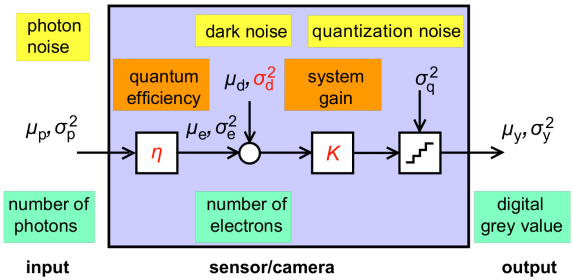


Figure 1: Mathematical camera model of a single pixel (source: EMVA Standard 1288 [9]).

gathered in the past to add synthetic defects as well and vary them in various ways. This might help us solve the "chicken or egg" problem. ML-based reproduction of images with defects requires that there are at least some images available, so it still depends on the quantity and quality of the input data. We can also build in any kind of defect we want, and, of course, the synthetic images created in this way are always labeled perfectly.

In the following, we analyze the noise characteristics of such a synthetic scene and how it resembles the linear camera model according to the standard EMVA 1288 [9]. Furthermore, we are developing strategies on how the noise characteristics can be improved so that they more closely resemble those of a real camera.

2 Fundamentals

2.1 Image Formation

We assume the transmission system to be a linear, shift invariant system. A standard digital industrial camera provides a linear photo response characteristic: the digital signal increases linearly with the number of photons received. These assumptions describe the properties of an ideal camera or sensor as described by the EMVA Standard 1288 (cf. Fig. 1) [9,10].

When a mean number of photons μ_p reaches the pixel area during exposure time, the fraction η is absorbed and creates a mean number

of photo electrons

$$\mu_e = \eta \mu_p. \tag{1}$$

The dark current μ_d is the mean number of electrons present without light. It is added to the mean number of electrons μ_e. Together they form a charge, which is converted by a capacitor to a voltage and amplified by the system gain K. Then the voltage is digitized resulting in a digital gray value μ_y:

$$\mu_y = K(\mu_e + \mu_d) = K\mu_e + \mu_{y.dark}. \tag{2}$$

The mean photon flux fluctuates randomly according to the Poisson probability distribution [11]. Therefore, the variance of the electron noise is equal to the mean number of electrons:

$$\sigma_e^2 = \mu_e. \tag{3}$$

All noise sources related to the sensor read out and amplifier circuits can be described by a signal independent normally distributed noise source with variance σ_d^2. The final analog-to-digital conversion adds another noise source that is uniformly distributed with variance $\sigma_q^2 = 1/12$. Because the variances of all noise sources add up linearly, the total temporal variance of the digital signal μ_y is given according to the laws of error propagation by

$$\sigma_y^2 = K^2(\sigma_d^2 + \sigma_e^2) + \sigma_y. \tag{4}$$

After plugging (3) into (4), we get

$$\sigma_y^2 = K^2(\sigma_d^2 + \mu_e) + \sigma_y. \tag{5}$$

The mean number of photo electrons μ_e cannot be measured. From (2) we get

$$\mu_e = (\mu_y - \mu_{y.dark})/K. \tag{6}$$

Now plugging (6) into (5) yields

$$\sigma_y^2 = K^2\sigma_d^2\sigma_q^2 + K(\mu_y - \mu_{y.dark}). \tag{7}$$

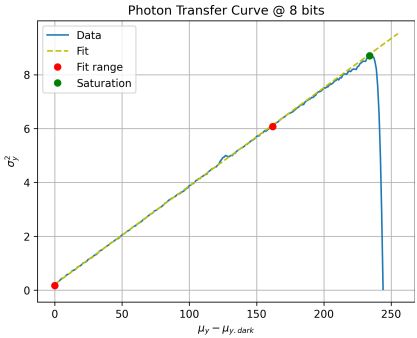


Figure 2: Photon transfer function of a real camera. The graph draws the measured variance σ_y^2 versus the mean photo-induced gray values $\mu_y - \mu_{y.dark}$ and the linear regression line used to determine the overall system gain K. The red dots mark the 0—70% range of saturation that is used for the linear regression.

Now the unknown parameters from Fig. 1 (red color) can be determined using the so called photon transfer method [12]: The system gain K is determined from the slope of (7), and the dark noise variance σ_d^2 from its offset.

So in summary, for a linear camera, the temporal noise with variance σ_y^2 shows a linear dependence on the mean signal μ_y. In order to verify whether a camera or a (synthetically generated) dataset exhibits this linear characteristic, one simply has to apply the photon transfer method and analyze the linearity of the graph. A real camera has the characteristics as shown in Fig. 2.

2.2 Ray Tracing

Ray tracing is a rendering technique that simulates the behavior of light to create realistic images using geometric optics. At its core, the process begins by sending rays from a virtual camera into a scene. When a ray encounters an object, the algorithm evaluates how light interacts with the surface at that point. This involves calculating surface normals, material properties, and the angle of incidence, which inform how much light is reflected, refracted, or absorbed by the material. Finally, after processing all rays for each pixel, the results are combined to form the final image. The accumulated energy values, influenced by

lighting and surface properties, create a 2D image representation of the scene. During the rendering, the render equation is approximately solved using a monte carlo approach. The render equation

$$L_\mathrm{o}(\mathbf{p},\boldsymbol{\omega}_\mathrm{o}) = L_\mathrm{e}(\mathbf{p},\boldsymbol{\omega}_\mathrm{o}) + \int_\Omega \mathrm{BRDF}(\mathbf{p},\boldsymbol{\omega}_\mathrm{i},\boldsymbol{\omega}_\mathrm{o}) L_\mathrm{i}(\mathbf{p},\boldsymbol{\omega}_\mathrm{i})(\boldsymbol{\omega}_\mathrm{i}^\intercal \mathbf{n})\,\mathrm{d}\,\boldsymbol{\omega}_\mathrm{i} \quad (8)$$

describes the propagation of light through the scene with $L_\mathrm{o}(\mathbf{p},\boldsymbol{\omega}_\mathrm{o})$ representing the outgoing radiance from point $\mathbf{p}$ in the direction $\boldsymbol{\omega}_\mathrm{o}$, $L_\mathrm{e}(\mathbf{p},\boldsymbol{\omega}_\mathrm{o})$ being the emitted radiance from point $\mathbf{p}$ in the direction $\boldsymbol{\omega}_\mathrm{o}$, $\mathrm{BRDF}(\mathbf{p},\boldsymbol{\omega}_\mathrm{i},\boldsymbol{\omega}_\mathrm{o})$ denoting the bidirectional reflectance distribution function (BRDF), which indicates how light is reflected from the direction $\boldsymbol{\omega}_\mathrm{i}$ to the direction $\boldsymbol{\omega}_\mathrm{o}$ at point $\mathbf{p}$, $L_\mathrm{i}(\mathbf{p},\boldsymbol{\omega}_\mathrm{i})$ representing the incoming radiance to point $\mathbf{p}$ from the direction $\boldsymbol{\omega}_\mathrm{i}$, the cosine $\boldsymbol{\omega}_\mathrm{i}^\intercal \mathbf{n}$ of the angle of incidence between the incoming light direction $\boldsymbol{\omega}_\mathrm{i}$ and the surface normal $\mathbf{n}$ and the positive hemisphere Ω above point $\mathbf{p}$.

3 Setup

The pipeline for image synthesis consists of several steps, which are described subsequently. First, the setup for the real world image acquisition is virtually recreated using 3D models. The open source 3D software Blender [13] makes it possible to either model the required objects manually or import existing models from CAD data and other sources. In Blender the visual inspection system setup is recreated in detail and the positions of sensors and lighting can be defined in the 3D scene. To generate images from the 3D scene, ray tracing is used as a rendering method. Ray tracing physically simulates light rays to create photo-realistic images by following the path of light rays and analyzing their interaction with surfaces to calculate effects such as shadows and reflections. Mitsuba 3 [14] is used as the rendering engine. To render the scene created in Blender with Mitsuba, all objects in the scene are exported and the Mitsuba Blender Add-On is used to generate an XML scene description. This scene description contains a textual description of the recreation of the measuring setup including all the information required for rendering with Mitsuba 3. The rendering is followed by a denoising process using Intel Open Image Denoise [15]. This open

source library offers high-quality and high-performance denoising filters. The rendering and denoising process is automated through scripting. In the process of rendering an image using the Mitsuba rendering tool, each generated pixel value represents the energy received at that pixel. This energy is linearly assigned to the pixel values. Similar to the saturation in CCD sensors, the pixel values are clipped at a maximum value of one. Before saving the images, a gamma correction of 2.2 is applied in accordance with the sRGB color space, and the images are quantized into 8-bit formats.

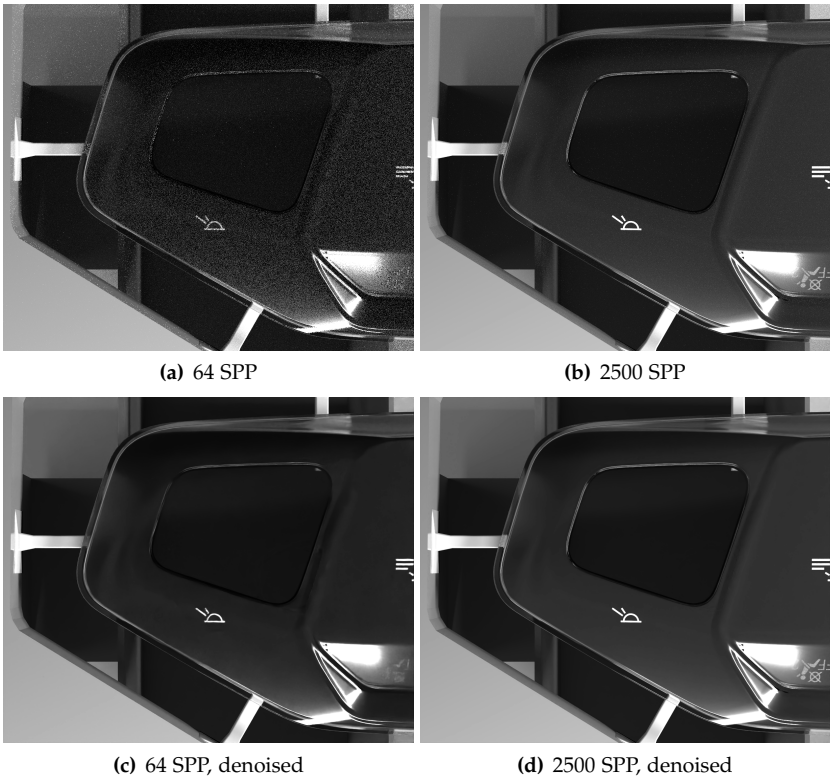
The image generation in ray tracing algorithms is dependent on random variables. Therefore, the seed of the random number generator is changed to produce statistically independent images.

The stochastic nature of the rendering process leads to local deviations from the perfect scene, which can be interpreted as spatial noise. Less noise can be achieved by increasing the rendering parameter samples per pixel (SPP) but at the cost of higher rendering times. In practice, the maximum allowed time to render an image sets an upper boundary for the maximum SPP.

Especially in dark field setups the noise is very strong. Fig. 3 depicts the test object as seen by the virtual camera, with the effects of different SPP and the denoiser turned off or on.

4 Experiments

The scene is a dark field setup and consists of several components, with the base being a housing made of aluminum profiles and black cover plates that ensure controlled imaging conditions on the inside. An area light is installed at the bottom of the housing, above which a movable shutter is fitted to shade the light. Additional lights are positioned above the mount on the rear wall and on both sides, whereby these area lights illuminate the test object from three directions. The camera is positioned above the test object. The object under consideration is a two-component injection molded part for which CAD data is provided. The top layer consists of transparent polymethyl methacrylate (PMMA), under which symbols with varying degrees of transparency are arranged. The next layer represents a deformed film, while the base consists of a thermoplastic base body. Fig. 4 visualizes



(a) 64 SPP **(b)** 2500 SPP

(c) 64 SPP, denoised **(d)** 2500 SPP, denoised

Figure 3: Scene rendered with different samples per pixel (SPP) and denoiser turned off or on.

the composition of the inspected object. For the simulation of image data, suitable materials for all components are defined using surface scattering models. The Mitsuba 3 renderer provides a principled bidirectional scattering distribution function (BSDF) model that can cover a wide range of materials and is used to simulate all materials contained in the scene. The individual parameters are adjusted, as far as possible, according to the real material properties, such as the refractive index of PMMA. Where it is not possible to transfer the material properties directly to the simulation model, the parameters are selected in such a

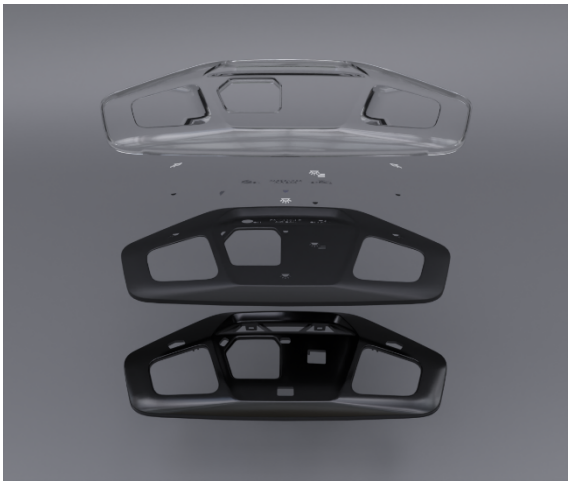
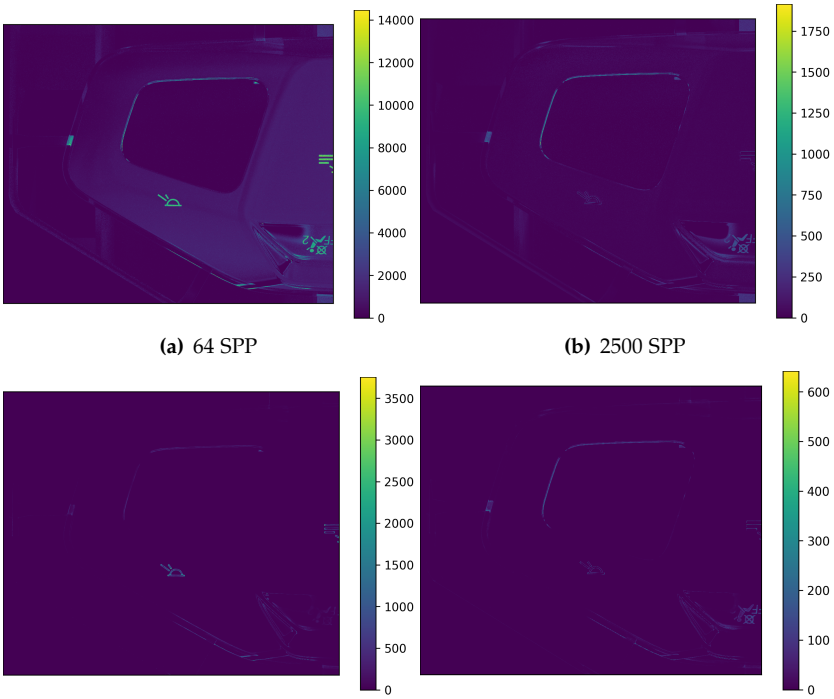


Figure 4: Composition of the modeled product.

way that the visual impression of the rendered images closely matches the real appearance.

For the evaluations, 50 images each with 64 SPP and 2500 SPP are generated using the pipeline described in Sec. 3.

The rendering noise (cf. Sec. 3) is not detectable from a single image, because any rendered image is only an approximation but we would need a perfectly rendered scene against which we could compare. Therefore, we render the same scene multiple times (50 in this paper). It is important to set a random seed. As a result, the imperfections, i.e., the spatial noise, occurs in different pixels for each rendered image. By looking at the rendered images (cf. Fig. 3) as a temporal sequence, like in a video, the noise now appears as temporal noise between the rendered images. Fig. 5 depicts the variance along the temporal axis of the rendered images. As can be seen, the noise decreases significantly with higher SPP as well with the denoiser turned on.



(a) 64 SPP **(b)** 2500 SPP

(c) 64 SPP, denoised **(d)** 2500 SPP, denoised

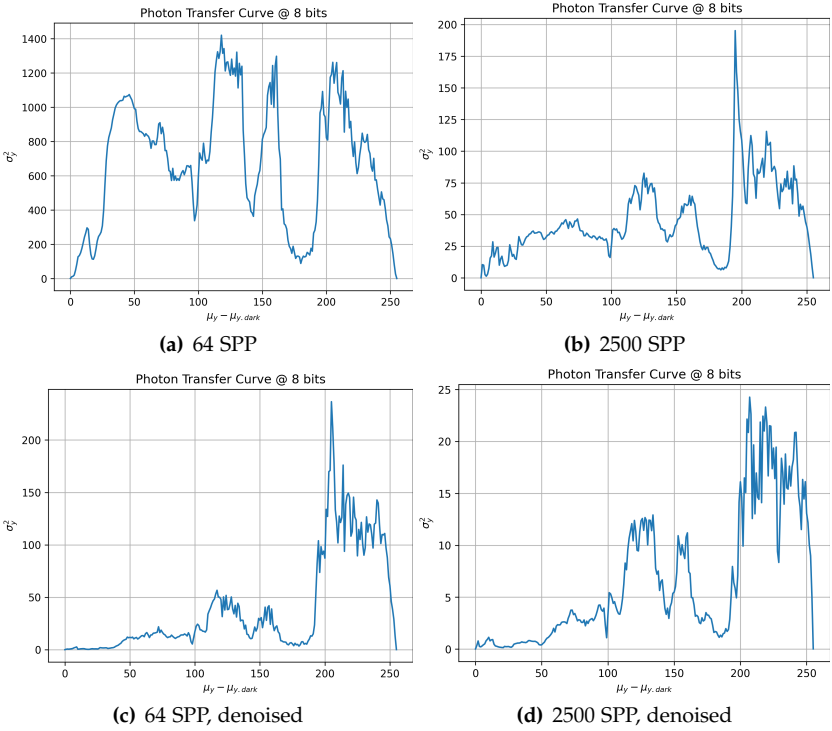
Figure 5: Variance along the temporal axes of the rendered images. Note the different scales.

5 Results

In this section we analyze how the rendering noise compares to the linear camera model.

To compute the photon transfer curve, we make use of the fact that the rendered data does not contain spatial non-uniformities as compared to a real camera sensor. Hence we do not have to apply the method described in the EMVA1288 standard and simply compute the average and variance along the temporal axis of the image sequence. We then quantize the average in bins with width one and average the variance at all pixels where the average has equal values.

Finally we plot the variance against the average yielding the graphs as shown in Fig. 6. They are very noisy compared to the photon transfer



(a) 64 SPP **(b)** 2500 SPP

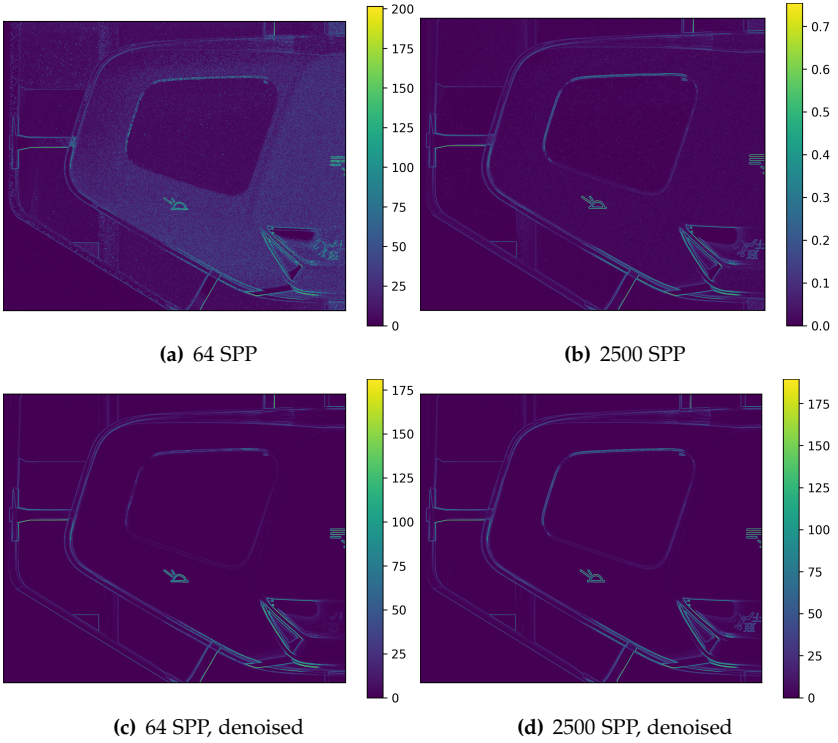
(c) 64 SPP, denoised **(d)** 2500 SPP, denoised

Figure 6: Photon transfer curve with different samples per pixel (SPP) and denoiser turned off or on.

curved of a real camera, cf. Fig. 2. They are non-linear and not even monotone. Therefore we conclude that the characteristic of rendering noise does not conform to the linear camera model according to the EMVA1288 standard.

It is very insightful to note that the variance is high near edges, i.e. where there are strong brightness changes in the image. Fig. 7 depicts the Sobel-filtered images for the four different rendering settings; besides the scaling they look quite similar to Fig. 5. As expected, it appears that the noise is particularly high in regions with complex light

propagation.



(a) 64 SPP **(b)** 2500 SPP

(c) 64 SPP, denoised **(d)** 2500 SPP, denoised

Figure 7: Sobel-filtered images with different samples per pixel (SPP) and denoiser turned off or on.

6 Proposed Method

A straight forward approach to minimize the noise is to render a scene multiple times and compute its variance along the temporal axis (cf. Fig. 5). We set for each pixel an individual SPP based on the targeted rendering time. It must be chosen in such a manner that the overall noise is minimal, i.e., low in regions with low temporal variance and

vice versa. However, to render a scene multiple times (50 to 100) beforehand is extremely time consuming.

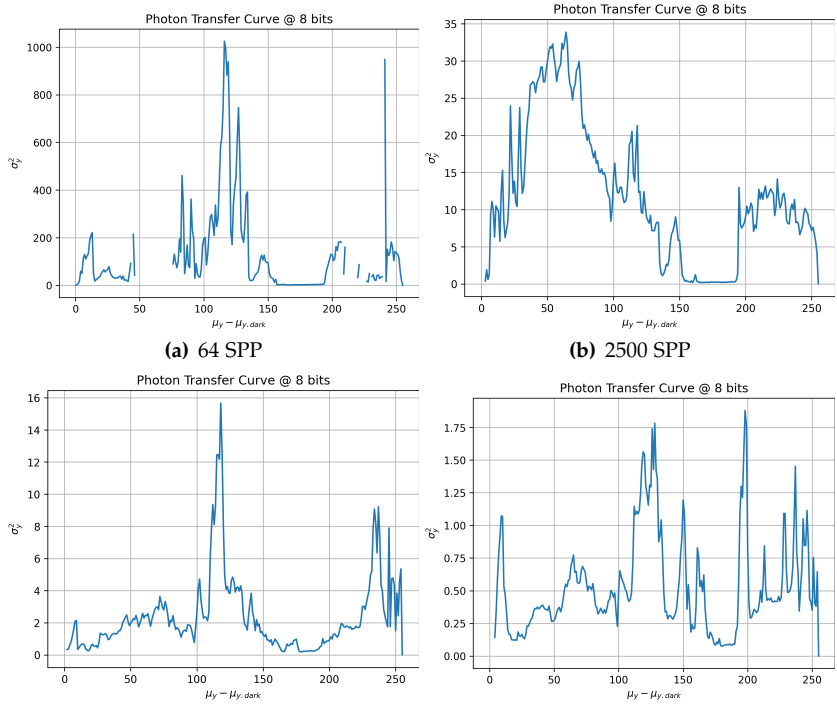
Therefore, we make use of the similarities between the variance images (cf. Fig. 5) and the Sobel images (cf. Fig. 7): The regions with strong edges can be extracted from a single rendered scene by edge detection, e.g. by using a Sobel-filter (cf. Fig. 7). This serves as an approximation for the variance but can be computed much faster.

If the photon transfer curves are now calculated without the regions with strong edges, the signal variance σ_y^2 is significantly reduced or even almost below 2. Here, the rendered image is practically noise free; to resemble the image of a real camera we can now add photon noise and dark current noise by simple parameterization based on the pixel gray values in compliance with the linear camera model [9].

7 Summary

Synthetically generated data contains temporal noise that does not correspond to the photon noise of real cameras, but rather correlates with the complexity of the scene. The rendered image is less accurate and therefore more susceptible to noise in regions with edges or strong brightness transitions and in places that exhibit diffuse volume scattering.

Based on these findings, a fast method was derived to identify the regions with high rendering noise from a single rendered image. These regions must be sampled with a higher SPP-setting, while the average SPP-setting is based on the targeted rendering time.



(a) 64 SPP **(b)** 2500 SPP

(c) 64 SPP, denoised **(d)** 2500 SPP, denoised

Figure 8: Photon transfer curve of filtered data with different samples per pixel (SPP) and denoiser turned off or on. The graph in (a) is discontinuous because some mean signal values μ_y no longer exist in the filtered data.

References

1. J. Meyer, T. Längle, and J. Beyerer, "About the Acquisition and Processing of Ray Deflection Histograms for Transparent Object Inspection," in *Irish Machine Vision & Image Processing Conference Proceedings*, 2016, pp. 9–16.
2. ——, "General Cramér-von Mises, a Helpful Ally for Transparent Object Inspection Using Deflection Maps?" in *Image Analysis*, P. Sharma and F. M. Bianchi, Eds. Cham: Springer International Publishing, 2017, vol. 10269, pp. 526–537, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-319-59126-1_44
3. J. Meyer, "Light Field Methods for the Visual Inspection of Transparent Objects," Ph.D. dissertation, 2019, medium: PDF Publisher: KIT Scientific Publishing. [Online]. Available: https://publikationen.bibliothek.kit.edu/1000091872
4. ——, "Next on stage: 'mc visi' – a machine vision simulation framework," Karlsruhe Institute of Technology, Tech. Rep. IES-2016-06, 2016.
5. J. Meyer, R. Gruna, T. Längle, and J. Beyerer, "Simulation of an inverse schlieren image acquisition system for inspecting transparent objects," in *Electronic Imaging*, 2016, pp. 1–9.
6. J. Meyer, T. Längle, and J. Beyerer, "About acquiring and processing light transport matrices for transparent object inspection," *tm-Technisches Messen*, pp. 731–738, 2016.
7. ——, "Acquisition and processing of light transport matrices for automated transparent object inspection," in *Forum Bildverarbeitung*, 2016, pp. 75–86.
8. ——, "Towards light transport matrix processing for transparent object inspection," in *Computing Conference*, 7 2017, pp. 244–248.
9. EMVA 1288 Working Group, "EMVA standard 1288 release 4.0 linear." [Online]. Available: https://www.emva.org/standards-technology/emva-1288/emva-standard-1288-downloads-2/
10. B. Jähne, *Digitale Bildverarbeitung und Bildgewinnung*, 7th ed. Springer Vieweg.
11. B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics*, 1st ed. Wiley. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/0471213748
12. J. R. Janesick, K. P. Klaasen, and T. Elliott, "Charge-coupled-device charge-collection efficiency and the photon-transfer technique," vol. 26, no. 10. [Online]. Available: http://opticalengineering.spiedigitallibrary.org/article.aspx?doi=10.1117/12.7974183

13. B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org
14. W. Jakob, S. Speierer, N. Roussel, M. Nimier-David, D. Vicini, T. Zeltner, B. Nicolet, M. Crespo, V. Leroy, and Z. Zhang, "Mitsuba 3 renderer," 2022, https://mitsuba-renderer.org.
15. A. T. Áfra, "Intel® Open Image Denoise," 2024, https://www.openimagedenoise.org.

Performance comparison of area-scan and event-based camera

Alkhazur Manakov[1], Helmut Herrmann[2], and Bernd Jähne[1,2]

[1] HCI at IWR, Heidelberg University,
Berliner Straße 43, 69120 Heidelberg
[2] AEON Verlag and Studio GmbH
Alter Rückinger Weg 31, 63452 Hanau

Abstract Event-based sensors asynchronously measure pixel brightness changes, and output a stream of events that encode the time, location and sign of the brightness changes instead of capturing images at a fixed rate, as conventional area-scan sensors do. Advantages of event-based sensors include: high temporal resolution and dynamic range (120 dB), low power consumption, and a compressed output stream. Comparison methodology between the two types of sensors is not available, therefore choosing between event-based and conventional area-scan camera for a given machine vision application is a challenge. We extended the dynamic range for the irradiance of the test equipment to 120 dB, characterized the performance in relation to irradiance (photon/(pixel s)), emulated event-based sensor functionality with conventional area-scan sensor and thus enabled a comparison. Normal EMVA 1288 standard measurement suffice for emulation, provided the irradiance series is dense enough. Area-scan cameras which meet the linear model of the EMVA 1288 standard, require no measurements, because it is possible to compute the emulated performance analytically. The comparison covers several area-scan cameras and three event-based cameras with different sensors.

Keywords Sensor characterisation, event-based, area-scan, performance, comparison

1 Introduction

State-of-the-art image sensors suffer from limitations imposed by their frame-based operation. The sensors acquire the visual information as a series of "snapshots" recorded at a predetermined frame rate. Biology does not know the concept of a frame. Biological vision systems outperform the best state-of-the-art artificial vision devices. Frames are not the most efficient form of encoding visual information. Firstly, the world, the source of the visual information, unlike frames, works asynchronously and in continuous time. Classical machine vision approach faces a dilemma loosing information between the frames or choosing high frame-rate. The latter requires more complex acquisition and processing hardware, with large bandwidth connection between them. Secondly, each recorded frame conveys the information from all pixels, regardless of whether this information, or a part of it, has changed since the last frame had been acquired. Two frames adjacent, dynamic contents of the scene, contain redundant information. Acquisition and handling of these dispensable data consume valuable resources and translate into high transmission power dissipation, increased channel bandwidth requirements, increased memory size, and processing power demands. An engineering solution inspired by the biological pixel-individual, frame-free approach may be more efficient than a traditional one.

The most advanced bioinspired vision sensors today [1] follow the natural, event-driven, frame-free approach, capturing transient events in the visual scene. Pixel analogue electronics stores a reference brightness level, and continuously compares it to the current brightness level. If the difference in brightness exceeds a threshold, that pixel resets its reference level and generates an event: a discrete packet that contains the pixel address, timestamp and polarity (increase or decrease) of a brightness change. Some sensors of these type do instantaneous measurement of the illumination level [2]. These type of sensors are called *even-based sensors*.

Choosing between cameras equipped with event-based and conventional area-scan sensors for a given machine vision application is a challenge, since the comparison methodology between the two types of sensors is not available. A first step in this direction was performed by Manakov and Jähne [3], who established the main concepts of event-

based sensors in extend to EMVA1288 characterization standard. Manakov et al [4] proposed the setup, data acquisition procedure and first measurement results. They also propose propose key performance indicators: event-delay and an analogous to signal-to-noise ratio defined for conventional area-scan cameras.

In this work we show the first direct performance comparison of event-based cameras with traditional are-scan cameras. The measuring equipment covers an extended irradiance range of 120 dB in order to cover the dynamic range of event-based cameras and also HDR area-scan cameras. In addition, we modified the characterization procedure to measure the performance not in relation to exposure (photon/pixel) as in the EMVA 1288 standard but to irradiance (photon/(pixel s)), because event-based cameras cannot be characterized by an exposure time. These changes enabled emulation the functionality of event-based sensor with conventional area-scan sensor. We demonstrate that for area-scan cameras which meet the simple linear model of the EMVA 1288 standard, no measurements are required, since their performance can be computed analytically. Thus, there is an additional advantage: it is possible to compute the best possible performance of an ideal area-scan camera with a quantum efficiency of one and no dark noise. Measurements of area-scan cameras require only normal EMVA 1288 measurements, provided the irradiance series is dense enough so that these measurements can be used to determine with which probability an intensity change can be detected, given a fixed exposure time with the corresponding frame rate.

2 Event-based sensor characterization basics

Sensitivity to small temporal contrasts, the response relation to the event-based sensor settings and its uniformity across the array are crucial performance parameters for the asynchronous, event-driven sensors. The minimum detectable temporal contrast or simply *noise equivalent contrast* is barely detectable by an event-based pixel step change of the irradiation level. Noise equivalent contrast sensitivity corresponds to the signal-to-noise ratio property of a conventional image sensor.

The simplest way of experimentally determining the irradiation contrast ΔE necessary for generating one event for given mean irradiance

level E and event threshold settings is gradually increase the stimulus step until an event is generated. In an ideal noise-free world, minimal found stimulus amplitude always results in an event when applied. In the real world conditions, the very same pixel will react differently to the same stimulus due to its, possibly different, initial condition, electronic noise, etc. Therefore, for event-based sensor characterization it has been proposed to operate with "event probability" instead [2,3]. It is defined for a given as ratio between the number of event responses M and the number of applied stimuli N, while background irradiance level and all the sensor settings remain unchanged.

$$p = \frac{M}{N} \tag{1}$$

Plotting the "event probability" vs. stimulus amplitude, in an ideal noise-free world, would yield a step function. In reality, such curve would have an "S"-shape, and is therefore named *S-curve*. Analysis of an S-curve provides crucial information about the performance of event-based sensor at the background irradiation levels and temporal contrasts the S-curve was acquired for. The contrast at 50% event probability point of an S-curve is the barely sensible contrast, similar to conventional area-scan cameras [5]. The slope at this point of an S-curve indicates the amount of noise. High slopes make S-curve closer to a step function, the influence of noise is small, low slopes indicate significant influence of noise on event probability. Vertical offset of an S-curve, as shown in Fig.1 for irradiation levels around 10^2 and 10^3, indicate the presence of events in absence of temporal contrast. In the next section we describe the S-curve acquisition procedure in detail, present the results for three different event-based sensors and introduce a metric, which enables area-scan and event-based sensor comparison.

3 Measures S-curves and change detectability

3.1 S-curves acquisition

The acquisition of S-curves presented in this section has been done on an EMVA1288 Standard conform setup, which consists of an in-

tegrating sphere, 4 LED modules, filter wheel with neutral density filters and a calibrated photo-diode. The LED modules are electronically controlled to generate background irradiation level and generate irradiation impulse with a controlled length. The neutral density filters allow to extend the dynamic range of the system, reaching very low irradiation levels and sample the irradiation space densely. The calibrated photo-diode provides the reference for the background irradiation levels E and the impulse amplitude ΔE. The acquisition of an S-curve is conducted with fixed sensor settings, namely biases and event-thresholds. The acquisition is performed for various background irradiation levels, varied by more than 6 orders of magnitude using neutral density filters. Each background irradiation level yields an S-curve. The sensor is stimulated by many impulses of various amplitudes are acquired for each background irradiation level. Thus, every sample of an S-curve corresponds to a pair (background irradiation; impulse amplitude). The measurement for each pair is repeated several hundred times for computing per-pixel event-probability. Three different event-based sensors were used in our measurements:

- Prophesee, gen. 3.1 (resolution 640x480, pixel 15µm x 15µm)
- Prophesee, gen. 4.1 (resolution 1280x720, pixel 4.86µm x 4.86µm)
- DAVIS 346, (resolution 346x260, 18.5µm x 18.5µm)

All the measurements were conducted with factory sensor settings, default bias values. There are 16 S-curves acquired, one for each neutral density filter, which determine the background irradiation level. The irradiation impulse amplitude is set by controlling LED module current. There are 128 different impulse logarithmically scaled amplitudes used for stimulating the sensors. Both Prophesee sensors were measured without a lens, limiting a region of active pixels to an area of 64x64 pixels around the center of the sensor. This was done in order to make sure, that the bandwidth of the sensor is not overloaded. Davis 346 sensor was measured with optics, which allowed to irradiate a small portion of the sensor. Davis the sensor does not provide the possibility to deactivate the pixels, therefore without the optics the bandwidth of the sensor is overloaded. The acquired S-curves are presented in figures 1 and 2.

3.2 S-curves analysis

All the three sensors demonstrate high dynamic range, aver 6 orders of magnitude, see Fig. 1. The S-curves of both Prophesee sensors in the lowest irradiation range are flat, namely the pixels did not produce any event for this background-impulse pairs. Davis 346 is more sensitive at this irradiation levels and produces events with over 60% and probability. Prophesee generation 3.1 sensor starts producing events at irradiation levels of 10 photons per second, unlike the Prophesee, generation 4.1 sensor. Sensors with larger light-sensitive part of the pixels have higher sensitivity at low background irradiation levels.

Vertical displacement the S-curves of the Prophesee sensors in the irradiation range from 20 to 100 photons per second is caused by the noise events sensors produce without stimulation. In case of the Davis 346 sensor the effect is less prominent. The slope of S-curves around 50% event probability, which can be observed from the distance between the samples on an S-curve, grows prominently and steadily from lower irradiation range to higher for Prophesee, generation 3.1 sensor. This means that for higher irradiation levels the influence of noise becomes less significant as S-curves' shape gets more similar to step function. Davis 346's S-curve slopes also grow not as fast and steady as in case of Prophesee, generation 3.1. The slope growth in case of Prophesee generation 4.1 sensor barely noticeable. The latter indicates, noise influence on the temporal contrast detection performance of the sensor is low.

3.3 Change detectability

S-curve is a useful performance indicator of an event-based sensor, but enable the comparison of contrast detection performance between area-scan and event-based sensors. Therefore, *change detectability* θ is introduced. It enables quantitative characterization of event-based image sensors and enabling their comparison to area-scan sensors. It is defined as the mean irradiance E at 50% event probability, $E_{50\%}$, divided by a measure for the "width" of the S-curve, which indicates the amount of noise mixed in with the signal. As a measure for the width of the S-curve, the inverse slope at 50% event probability is taken. This

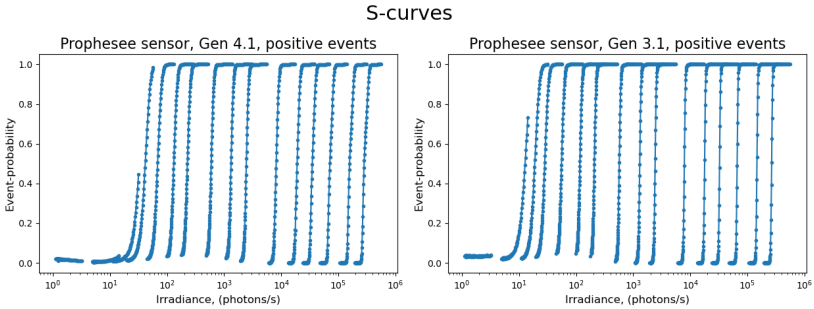


Figure 1: S-curves. Left - Prophesee sensor, generation 4.1; Right - Prophesee sensor, generation 3.1.

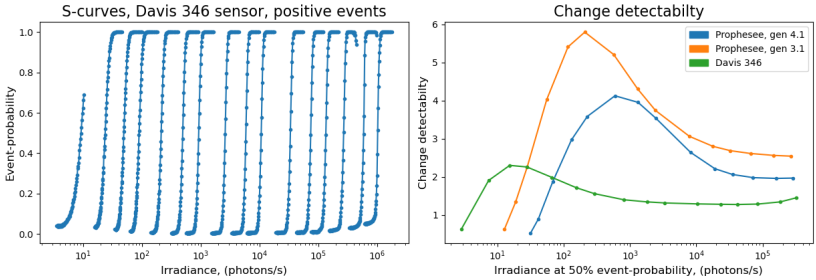


Figure 2: Left - S-curves for the Davis 346 sensor; Right - Change detectability comparison for the Prophesee and Davis sensors.

results in the following definition:

$$\theta = E_{50\%} \frac{\mathrm{d}S(E)}{\mathrm{d}E}. \tag{2}$$

The higher θ is, the lower contrast is required to detect an event.

Change detectability for the three sensors under test was computed and presented in Fig 2. The contrast detection performance of all the sensors in lower irradiation levels is low, but grows with the background irradiation. The peak of change detectability for all the three sensors coincides with the maximum of event noise in absence of stim-

uli, as if the noise would help the sensor reaching the 50% event-probability threshold. Further growth of the background irradiation levels leads to gradual decrease of the change detectability. That is, for these irradiation levels higher contrasts/impulse amplitudes are required to generate an event. In the highest irradiation levels the metric becomes constant.

4 Event-based sensor emulation

In this section we emulate an event-based sensor with an area-scan sensor having a linear response. Namely, we theoretically investigate the event-probability response of an area-scan sensor, which is used for temporal contrast detection.

4.1 Basic approach

In order to detect an event, two frames must be taken after each other. Thus the maximum frame rate of an area-based image sensor determines the rate and temporal resolution with which events can be detected. An event can be detected if the difference in the gray values is larger than a given threshold τ. The proper stetting of the threshold depends on the temporal noise. If the threshold is set too low, events will also be generated if there is no generated if there is no gray value change. It is therefore required to compute the probability density function (pdf) of the difference signal with a given noise level.

The goal is to compute the event probability and the resulting S-curves analytically. Therefore it was decided to use a normal distribution. Photon shot noise is Poisson distributed, but the normal distribution is a sufficiently good approximation. Standard industrial image sensors have saturation capacities in the order of 10,000 electrons. The normal distribution is already a good approximation for mean values of just 30 electrons [6]. In the following computations we neglect nonuniformities. This is justified because differences of two only slightly different gray values are subtracted from each other so that the stationary inhomogeneity is canceled out.

It is assumed that the standard deviation of the temporal noise changes with the gray value without assuming a special dependency.

With this flexible approach, it is possible to emulate any area sensor. Therefore two random variables with the distributions $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$ must be subtracted. This results in

$$N(\mu_n, \sigma_n) = \frac{1}{\sqrt{2\pi}\sigma_n} \cdot e^{-\frac{(g-\mu_n)^2}{2\sigma_n^2}} \quad \text{with} \quad n = 1, 2 \tag{3}$$

The distribution of the difference signal $\Delta g = g_2 - g_1$ is given by the convolution of the two distribution and also normally distributed with a mean $\Delta\mu = \mu_2 - \mu_1$ and with added variances ($\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$):

$$N_\Delta(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(\Delta g - \Delta\mu)^2}{2\sigma}} \tag{4}$$

4.2 Computation of the S-curve

As it is implemented in the event-based sensor a non-zero threshold τ is defined. In order to detect an event at a pixel, the difference Δg must be larger than τ in order for the sensor to generate an event. This means that $N_\Delta(\mu, \sigma)$ must be integrated from τ to ∞ resulting in

$$S(\tau, \Delta\mu, \sigma) = \frac{1}{2}\text{erf}(\frac{\tau - \Delta\mu}{\sqrt{2}\sigma}). \tag{5}$$

The Gaussian error function has S-curve shape. There is a consequence which follow from Equation 5: if $\tau = \Delta\mu$, then the event-probability $S(\tau, \Delta\mu, \sigma) = 1/2$. The slope of the S-curve $S(\tau, \Delta\mu, \sigma)$ is given by

$$\frac{\mathrm{d}S(\tau, \Delta\mu, \sigma)}{\mathrm{d}\Delta\mu} = -\frac{e^{-\frac{(\tau-\Delta\mu)^2}{4\sigma}}}{2\sqrt{\pi}\sigma}. \tag{6}$$

The slope of $S(\tau, \Delta\mu, \sigma)$ is a non-linear function. In order to find its maximum we compute the second order derivative of $S(\tau, \Delta\mu, \sigma)$.

$$\frac{\mathrm{d}^2S(\tau, \Delta\mu, \sigma)}{\mathrm{d}\Delta\mu^2} = -\frac{(\tau - \Delta\mu) \cdot e^{-\frac{(\tau-\Delta\mu)^2}{4\sigma^2}}}{2\sqrt{\pi}\sigma^3}. \tag{7}$$

The right side of the Equation 7 is equal to zero an the point where $\Delta\mu = \tau$. Therefore, the S-curve's maximum values is at $\Delta\mu = \tau$, where the value of the event probability is $1/2$. The slope is independent of the chosen threshold τ and is equal to $1/(2\sqrt{\pi}\sigma)$.

4.3 Signal-to-noise ratio (SNR) and chance detection

Characterization of the conventional image sensors is a well known procedure. With respect to temporal noise the essential parameter is the signal-to-noise ratio, or short SNR as a function of the exposure per pixel in photons N_p:

$$\mathrm{SNR}(N_p) = \frac{\mu}{\sigma}, \tag{8}$$

where μ and σ are the mean and standard deviation of the digital output signal.

The SNR can be measured using the measuring and evaluation techniques described by the EMVA standard 1288 using an irradiation series from dark to saturation [5]. For a simple linear image sensor without any noise changing preprocessing, the SNR can be related to the quantum efficiency η and the temporal variance of the dark signal σ_d^2:

$$\mathrm{SNR}(N_p) = \frac{\eta N_p}{\sqrt{\sigma_d^2 + \eta N_p}}, \quad \mathrm{SNR}_{\mathrm{ideal}}(N_p) = \sqrt{N_p} \tag{9}$$

In case of event-based sensor the SNR cannot be defined the same way. The definition proposed by Manakov et al [4] can be used for comparing event-based sensor between each other, but does not establish the relation to area-scan cameras. This can be established using the definition of *change detectability* in eq. 2, because the slope of the S-curve in known from eq. 6:

$$\theta = \frac{\mu_{50\%}}{2\sqrt{\pi}\sigma} = \frac{\mathrm{SNR}(\mu_p)}{2\sqrt{\pi}}, \tag{10}$$

As follows, the contrast detectability θ is $2\sqrt{\pi} \approx 3.54$ times smaller than the SNR of an are-scan sensor. For a direct comparison with event-based cameras, not the exposure N_p must be used, but the irradiance E. In this way the exposure time t_{exp} is introduced: $N_p = Et_{\mathrm{exp}}$ and the final result is

$$\theta = \frac{\mathrm{SNR}(Et_{\mathrm{exp}})}{2\sqrt{\pi}}. \tag{11}$$

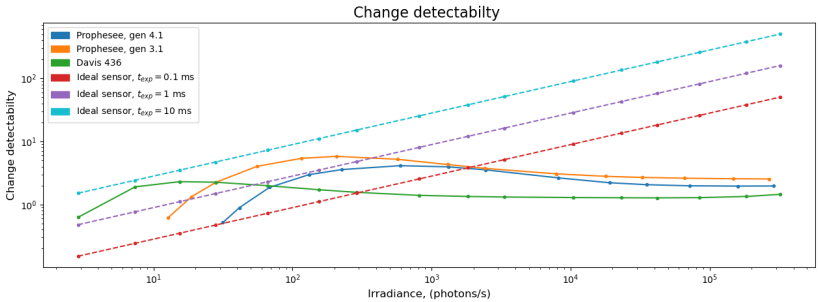


Figure 3: Change detectability comparison for event-based and an ideal area-scan sensor.

This can be applied to the SNR of a linear and ideal camera according to eq. 9 and results in

$$\theta(E) = \frac{\eta E t_{\text{exp}}}{2\sqrt{\pi}\sqrt{\sigma_d^2 + \eta E t_{\text{exp}}}}, \quad \theta_{\text{ideal}}(E) = \frac{\sqrt{E t_{\text{exp}}}}{2\sqrt{\pi}} \tag{12}$$

The used exposure time of an area image sensor thus determines which contrast is required to detect an event. In Fig. 3 the change detectability of an ideal area sensor with exposures times of 0.1, 1, and 10 ms is compared with measurements from event-based cameras

5 Conclusion and outlook

In this work the change detectability metric was introduced. It enables quantitative characterization of contrast detection performance of event-based cameras. Change detectability metric and the conducted theoretical investigation on the event-probability response of an area-scan sensor, which is used for temporal contrast detection was conducted, establish the comparison link between area-scan and event-based cameras. Moreover, it has been demonstrated that the metric can be calculated for any area-scan sensor with non-linear response, characterized in terms of EMVA 1288 standard. S-curve measurements performed over high dynamic range of irradiance levels was performed for three different event-based sensors. Change detectability for the

measurements was calculated and presented together with the S-curve analysis. The theoretical investigations with area-scan image sensors emulating event-based sensors will be complemented in the future by the measurements performed with linear and high dynamic range area-scan sensors.

References

1. G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.
2. C. Posch and D. Matolin, "Sensitivity and uniformity of a 0.18µm CMOS temporal contrast pixel array," in *Circuits and Systems, ISCAS 2011, IEEE International Symposium*, 2011, pp. 1572–1575.
3. A. Manakov and B. Jähne, "Characterization of event-based image sensors in extent of the emva 1288 standard," *Tagungsband "Forum Bildverarbeitung 2020"*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:269583737
4. A. Manakov, H. Herrmann, and B. Jähne, "Towards integration of event-based cameras into emva 1288 characterization," in *6th European Machine Vision Forum*, Wagingen, 2023.
5. EMVA 1288 Working Group, "EMVA Standard 1288 - standard for characterization of image sensors and cameras, release 4.0," European Machine Vision Association, open standard, 2021.
6. B. Jähne, *Digitale Bildverarbeitung*, 8th ed. Berlin, Heidelberg: Springer, 2024.

Benefiting from quantum? A comparative study of Q-Seg, quantum-inspired techniques, and U-Net for crack segmentation

Akshaya Srinivasan[1,4], Alexander Geng[1], Antonio Macaluso[2], Maximilian Kiefer-Emmanouilidis[3,4], and Ali Moghiseh[1]

[1] Fraunhofer Institute for Industrial Mathematics (ITWM), Fraunhofer-Platz 1, 67663 Kaiserslautern
[2] Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Campus D 3.2, 66123 Saarbrücken
[3] Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Trippstadter Str. 122, 67663 Kaiserslautern
[4] Department of Computer Science and Research Initiative QC-AI, RPTU Kaiserslautern-Landau, Erwin-Schrödinger-Straße 48, 67663 Kaiserslautern

Abstract Exploring the potential of quantum hardware for enhancing classical and real-world applications is an ongoing challenge. This study evaluates the performance of quantum and quantum-inspired methods compared to classical models for crack segmentation. Using annotated gray-scale image patches of concrete samples, we benchmark a classical mean Gaussian mixture technique, a quantum-inspired fermion-based method, Q-Seg a quantum annealing-based method, and a U-Net deep learning architecture. Our results indicate that quantum-inspired and quantum methods offer a promising alternative for image segmentation, particularly for complex crack patterns, and could be applied in near-future applications.

Keywords Quantum computing, quantum image segmentation, quantum optimization, image processing, disordered systems

1 Introduction

Quantum computing has emerged as one of the leading technologies to improve the efficiency and solvability of complex problems. Still,

the bridge between fundamental and applied research is very narrow and under construction. Unsupervised learning emerges as a particularly promising avenue for the adoption of quantum computing in machine learning. Classical algorithms often struggle to efficiently detect patterns in unlabeled data, a common scenario in many practical applications. Recent advancements have showcased the potential of quantum optimization techniques in addressing unsupervised segmentation tasks [1, 2]. Furthermore, combining quantum computing with classical methods have led to quantum-inspired (QI) and hybrid models like quantum transfer learning, which have been used for example for crack detection [3].

In this paper, we want to build on these developments, and furthermore evaluate how quantum effects in QI methods can be harnessed to advance classical algorithms as well as benchmarking current state-of-the-art approaches. As a use case we have chosen crack-segmentation, a real world problem, which we consider a tremendous important task to evaluate for example the quality of current roads and infrastructure, see Fig. 1 (a). By conducting a systematic comparison between four approaches, where two benefit from quantum, we seek to identify specific areas where non-classical approaches offer advantages. This research not only contributes to the understanding of quantum computing's practical applications but also guides future developments in algorithm design and implementation within the field.

2 Segmentation Techniques

This section examines four methodologies for segmenting concrete cracks: Mean Gaussian Mixture (MGM), QI Hamiltonian, Q-Seg, and U-Net. Using a dataset of 32×32 pixel images annotated with ground truth crack locations, each method processes input images to generate segmentation masks that delineate detected cracks. The approaches differ in complexity and computational demands, reflecting advancements in classical and quantum techniques. Figure 1 (c) provides a comparative overview of these workflows.

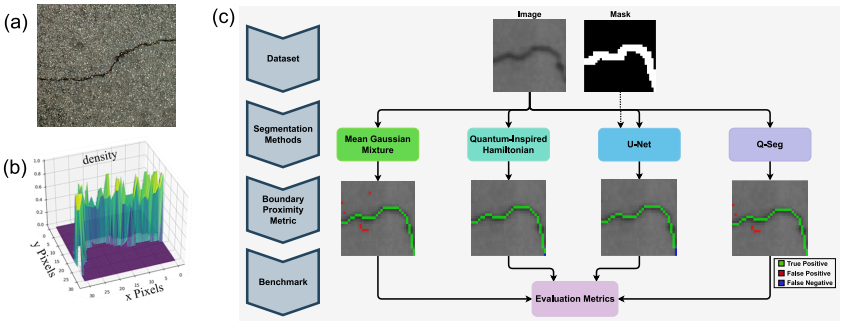


Figure 1: Overview of crack segmentation motivation and methodology: (a) Cracks on roads illustrating real-world infrastructure challenges, (b) Results from the QI approach, accurately identifying crack locations using localized states tied to negative eigenvalues, and (c) Comparative pipeline of crack segmentation methods.

2.1 Mean Gaussian Mixture

The Gaussian mixture model is a fundamental image segmentation technique known for its simplicity and efficiency, especially when objects of interest, like pores, have distinct intensity levels. This classical method is computationally inexpensive and versatile, making it ideal for preliminary segmentation tasks. In our study, we adapt Otsu Thresholding [4] to segment cracks in concrete images. Otsu's method determines the optimal threshold that maximizes between-class variance, effectively separating the foreground (cracks) from the background. To ensure consistent intensity across all samples, each image is normalized to a range of $[0, 255]$, addressing ambient lighting variations. Otsu method is applied to 30 images, calculating the optimal threshold for each. For consistency in our comparative analysis, we use the mean threshold across these images as a global threshold, allowing us to benchmark different segmentation methods. This approach balances individual image optimization and comparative consistency across the dataset.

2.2 Quantum-Inspired Hamiltonian

Due to the rise of quantum computing also QI-techniques have become more prominent in image processing [5]. In the original context, QI refers to the idea to evaluate classically how quantum effects like superposition, entanglement or wave function collapse (measurements) may change an algorithm of interest [6,7], and in the best case how to benefit from it. Simulating general many-body quantum systems and circuits becomes exponentially difficult as the number of particles or qubits increases. However, many problems can be reduced to polynomial complexity. For example, single-particle Hamiltonians allow each particle to be evaluated separately, with the combined dynamics described as presented here by Fermi-Dirac statistics [8]. We refer to [9] and [10] for a deeper discussion of the underlying physical effect, fitting in the context of this work. In this paper, we show in a proof-of-concept that the single-particle effect of Anderson Localization (AL) [11, 12] can be efficiently used for image- and especially crack-segmentation. Initially used to explain electron behavior in disordered lattices, this model introduces randomness into the potential energy landscape, leading to the localization of wave functions, see Fig.1 (b). Effects of AL and disorder have been found suitable for image and signal processing tasks such as image representation and denoising [10], augmentation [9, 13] and signal transfer in optical fibre [14]. Embedding an image in a Hamiltonian yield a simple matrix form $N \times N$, where N correspond to lattice sites or qubits. To formulate our Hamiltonian matrix, we slightly adjust the adaptive signal decomposition presented in [10]. The key QI idea is that the embedding of the images themselves will self-induce AL due to their disordered landscape, i.e. rough surfaces and cracks which seem like random disordered potentials. Thus, the eigenstates of the Hamiltonian will be close to a unit vector (hot encoded one), only in the strongly disordered areas, and rather extended in areas of weaker disorder. In our study, we embed the image on a 2D lattice. Here it is known that the localization length scales exponentially with disorder strength as well as energy. Thus leading to strong dependence on disorder effects, as particles will mostly localize in areas of the cracks and holes. The embedding works as follows. First, the $m \times n$ images are flattened $m \cdot n$, such that a pixel value at $A_{ij} \rightarrow a_l$ with $l = j + ni$. The corresponding single-particle Hamiltonian of $N \times N$ size where

$N = m \cdot n$ reads

$$H_{i,j} = \begin{cases} a_i & \text{if } i = j \\ G(a_i, a_j) & \text{if } |i-j| = 1 \text{ and} \\ & \quad i, j \mod n \neq 0, \\ G(a_i, a_j) & \text{if } |i-j| = n \\ 0 & \text{otherwise} \end{cases} \qquad G(a_i, a_j) = \exp\left(-\frac{(a_i - a_j)^2}{2\sigma^2}\right) \quad (1)$$

The $G(a_i, a_j)$ is the Gaussian difference only for nearest-neighboring pixels and σ^2 is the Gaussian variance. The diagonal elements of the Hamiltonian matrix a_i correspond to the pixel values (potentials), while the off-diagonal $G(a_i, a_j)$ elements represent the Gaussian weights between nearest-neighboring pixels (kinetic terms). The Hamiltonian in its diagonal form can already be considered as a thresholding technique, however, inferior to the MGM explained in Sec. 2.1. Only due to the kinetic terms we will get extended states which do not contribute significantly to the density of the crack or the whole picture at all. However, we have to be careful to construct kinetic terms for nearest neighbours, as otherwise we might generate extended states again [15]. Furthermore, we have found the Gaussian distance better than constant kinetic terms in [10]. The final mask shows the elementwise summation of magnitudes of all eigenstates (localized particles), tied only to negative eigenvalues. The sum of all negative eigenvalues corresponds to the many-body ground state energy and thus is the minimal energy of the system. This method shows surprising good results and efficiently finds the crack, see Fig.1 (b).

2.3 Q-Seg: Unsupervised Quantum Algorithm

Q-Seg is an innovative image segmentation method that utilizes quantum annealing [16, 17]. Initially developed for Earth observation images [2], Q-Seg adapts to detect cracks in concrete by efficiently solving the Maxcut problem using a D-Wave quantum annealer.

The segmentation procedure begins by converting the input image into a lattice graph where each pixel is a node, preserving spatial connectivity. Edges are weighted based on pixel similarity, calculated as squared differences in our case to enhance contrast to gray-scale crack images. The segmentation task becomes a graph cut problem, aiming

to find a maximum cut that best partitions the vertices based on edge weights. To overcome the computational challenges of finding maximum cuts, Q-Seg reformulates the problem into a Quadratic Unconstrained Binary Optimization (QUBO) formulation, suitable for quantum annealing [18]. The QUBO problem is mapped onto the *Pegasus* architecture [19] of the D-Wave quantum annealer, where the system starts in a superposition of all possible states and gradually evolves toward the lowest energy state that represents the optimal solution. The D-Wave quantum annealer iteratively adjusts system parameters and annealing cycles. This iterative adjustment increases the probability of reaching the global minimum.

The final result of the quantum annealing process is a binary string corresponding to the segmented image, providing a direct solution to the image segmentation problem. This unsupervised segmentation approach proved effective beyond its original Earth observation application in adapted scenarios such as crack detection in concrete structures, demonstrating Q-Seg's versatility and robustness in various image segmentation tasks.

2.4 U-Net

U-Net is a deep-learning architecture designed for biomedical image segmentation [20], renowned for its performance in tasks with limited annotated data. Its versatility extends to various applications, including medical imaging, satellite imagery, and material defect detection. This study focuses on utilizing U-Net for crack detection in concrete, leveraging its strength in producing detailed segmentation masks.

The U-Net architecture features a U-shape consisting of an encoder and a decoder. The encoder reduces the spatial dimensions of the input image using convolutional and max-pooling layers, creating a lower-resolution representation. The decoder upsamples the image, restoring lost spatial dimensions. A notable advantage of U-Net is its skip connections between encoder and decoder layers, which allow access to high-resolution feature maps, improving segmentation accuracy. For binary segmentation tasks like crack detection, a sigmoid activation function classifies each pixel as a crack or background.

In this study, U-Net architecture is modified to handle 32×32 pixel crack images, trained on a dataset of 456 labeled patches. The model,

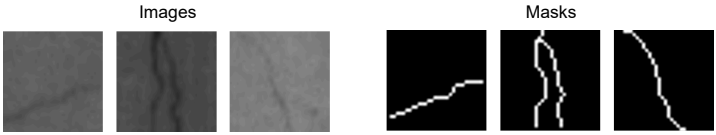


Figure 2: Sample images of cracks with corresponding masks.

with approximately 21.7 million trainable parameters, generates a binary mask indicating crack presence. The training utilized a batch size of 16 over 50 epochs, completed in about 13 minutes on a local machine with an Intel Core i7 CPU and 16GB of RAM. U-Net's ability to capture detailed features makes it a valuable tool for precise and reliable crack segmentation.

3 Dataset and Metrics for Segmentation Analysis

This study utilizes grayscale images of concrete for crack segmentation using four different methods. The original images measure approximately $16,000 \times 32,000$ pixels, with cracks only $1-3$ pixels wide, making detection challenging for the human eye and machine learning algorithms. To accommodate the limitations of the D-Wave quantum annealer, including the restricted number of qubits and limited runtime, we divide the images into smaller 32×32 pixel patches. Our complete dataset consists of 456 manually annotated patches, split into 70% for training and 30% for validation of the U-Net model. We evaluate the performance of the segmentation methods—MGM, QI Hamiltonian, Q-Seg, and U-Net on an unseen test dataset of 30 patches. Figure 2 presents example patches with manually annotated masks highlighting the cracks, which are used as ground truth for performance comparison.

3.1 Evaluation Metrics

To evaluate the segmentation methods, we use the confusion matrix, F1 score, and Intersection over Union (IoU). The confusion matrix provides four key metrics: True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN), which assess the accuracy of

crack predictions. The F1 score combines precision and recall, while IoU measures the overlap between predicted and ground truth masks, providing a comprehensive evaluation of segmentation performance.

$$F1\ Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad IoU = \frac{TP}{TP + FP + FN}. \tag{2}$$

3.2 Boundary Proximity Metric

In traditional segmentation tasks, evaluation metrics such as the confusion matrix may not adequately reflect performance when slight deviations in boundary prediction occur. In crack segmentation tasks, where cracks are typically thin structures with irregular boundaries, these minor deviations should be tolerated to some extent. The Boundary Proximity Metric (BPM) addresses this by adjusting the boundary around the cracks in both the predicted segmentation I_P and the ground truth I_{GT}, allowing for more lenient evaluation in cases of minor misalignment. The process starts by skeletonizing both the ground truth $S(I_{GT})$ and predicted segmentation results $S(I_P)$. Skeletonization reduces each crack to its core structure, which helps in focusing only on the most critical regions. After skeletonization, both the ground truth and the predicted results are dilated using flat disk structuring element B_r by a radius of r pixels. This dilation adjusts the boundary, expanding it to account for small deviations. Any predicted crack pixels that were previously identified as false positives or false negatives but fall within this dilated boundary (i.e., within r pixels of the ground truth) are then reassigned as true positives. This re-calibration of TP, TN, FN and FP are mathematically formulated as follows

$$\begin{aligned} I_{\widetilde{TP}} &= [S(I_{GT}) \oplus B_r] \cap S(I_P), & I_{\widetilde{FP}} &= [[S(I_{GT}) \oplus B_r] \cap S(I_P)] - S(I_P), \\ I_{\widetilde{FN}} &= [[S(I_P) \oplus B_r] \cap S(I_{GT})] - S(I_{GT}), & I_{\widetilde{TN}} &= I_{ones} - \sum \left(I_{\widetilde{TP}} + I_{\widetilde{FP}} + I_{\widetilde{FN}} \right), \end{aligned} \tag{3}$$

where I_{ones} is $n \times n$ matrix with all entries equal to 1 and n is the size of the crack image. The new counts for true positives $\widetilde{TP}$, false positives $\widetilde{FP}$, false negatives $\widetilde{FN}$, and true negatives $\widetilde{TN}$ are calculated by applying Eq. 3 and

$$\widetilde{X} = \left\| I_{\widetilde{X}} \right\|_1 = \sum_i \sum_j I_{\widetilde{x}}(i,j) \quad \text{where} \quad \widetilde{X} \in \left[\widetilde{TP}, \widetilde{FP}, \widetilde{FN}, \widetilde{TN} \right]. \tag{4}$$

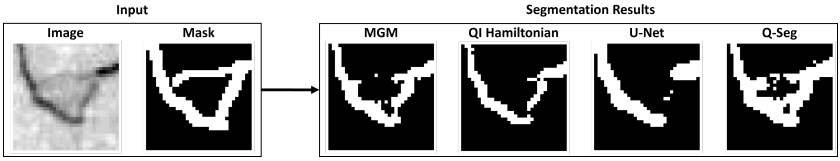


Figure 3: Crack segmentation results from four different techniques: MGM, the QI Hamiltonian method, U-Net, and Q-Seg.

Using this boundary proximity metric makes the evaluation more forgiving towards minor misalignment that would otherwise result in a higher count of false positives and false negatives. This approach is especially beneficial in crack segmentation, where small discrepancies in boundary prediction are often unavoidable due to the irregular shapes of cracks.

4 Results and Discussion

We have benchmarked MGM, QI Hamiltonian, Q-Seg, and U-Net using standard evaluation metrics and prediction time for segmenting 30 images. Additionally, we employ the BPM to refine the evaluation by considering slight deviations in the predicted crack boundaries compared to the ground truth. Each segmentation method shows distinct results in detecting cracks, as illustrated in Figure 3. This figure underscores the strengths and limitations of each approach in capturing fine details and improving prediction accuracy. Furthermore, Figure 4 demonstrates the visual comparison of the segmentation results, showing both the standard confusion matrix and the one after applying BPM. An overlay diagram illustrates the alignment between the predicted crack masks and the actual cracks. This comparison emphasizes the impact of BPM in improving segmentation accuracy, particularly in challenging cases where the cracks are faint or unclear. Table 1 provides a detailed comparison of the segmentation methods, both with and without BPM adjustments, highlighting their effectiveness in crack detection. The table includes average F1 scores and IoU values for each segmentation technique, allowing for a comprehensive performance assessment. We also present the corresponding prediction times to provide insight

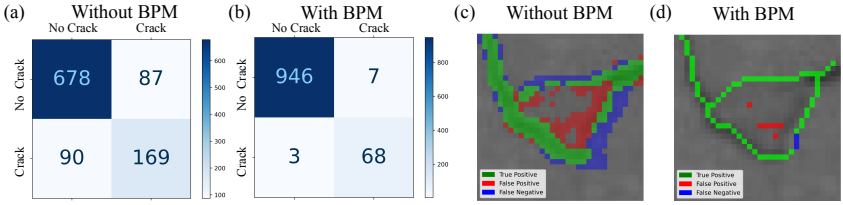


Figure 4: Visual comparison of segmentation results, including the standard confusion matrix (a), the confusion matrix post-BPM application (b), and an overlay of predicted crack masks against actual cracks before (c) and after BPM (d).

Table 1: Performance comparison of four crack segmentation techniques with and without boundary proximity metric (BPM) using standard evaluation metrics.

Segmentation Methods	Metrics without BPM		Metrics with BPM		Prediction Time (s)
	Avg IoU	Avg F1 Score	Avg IoU	Avg F1 Score	
MGM	0.5783 ± 0.1611	0.7197 ± 0.1449	0.7454 ± 0.1836	0.8439 ± 0.1781	0.032 ± 0.007
QI Hamiltonian	0.6218 ± 0.178	0.7478 ± 0.1766	0.9447 ± 0.1241	0.9693 ± 0.1016	156 ± 10
U-Net	0.6159 ± 0.1440	0.7522 ± 0.1145	0.8945 ± 0.1834	0.9395 ± 0.1697	2.292
Q-Seg	0.5728 ± 0.1687	0.7079 ± 0.1735	0.8014 ± 0.1431	0.8753 ± 0.1357	2.277 ± 0.250

into the computational efficiency of each segmentation task. From Table 1, we observe that the QI Hamiltonian method delivers the best overall segmentation performance, with the highest average IoU and F1 score using BPM. Without BPM, it performs within the same error bounds as U-Net, showcasing its robustness. However, its prediction time is significantly longer, which limits its efficiency for real-time applications, especially on larger datasets. U-Net, while slightly behind the QI Hamiltonian method in segmentation accuracy with BPM, is still highly competitive, especially without BPM. However, it demands considerable computational resources, and its training time of 13 minutes for 456 (32×32) samples is not included in the prediction time. Q-Seg has a prediction time similar to U-Net, which includes only the Quantum Processing Unit (QPU) access and qubit embedding time. Though it is not as accurate as QI Hamiltonian or U-Net, presents a competitive alternative with balanced performance and does not require labeled data for training, making it practical for scenarios where training data is limited. The MGM model performs comparably to Q-Seg but slightly worse with BPM. However, it is the fastest method, avoiding any training phase like U-Net. Despite its speed, this method lacks

segmentation accuracy and is unlikely to perform well with a single mean threshold on larger, more complex datasets. Overall, the outcomes suggest that while U-Net and the Hamiltonian method offer the highest accuracy, Q-Seg provides a balanced alternative with moderate performance and no training requirements.

Future work could focus on testing these approaches on larger datasets to assess their effectiveness in realistic scenarios. Optimizing the Hamiltonian method for GPU parallel processing could yield a 20x speedup [21] and exploring quantum simulations using ultra-cold gas setups [22, 23]. Additionally, exploring Q-Seg on gate-based quantum computing [24] and identifying other challenging domains for annotated data can enhance the application of quantum methods.

Acknowledgements

We'd like to thank the Quantum Initiative Rhineland-Palatinate (QUIP) for their support. This work was also partially funded by the Research Initiative 'Quantum Computing for Artificial Intelligence' (QC-AI) and the Federal Ministry for Economic Affairs and Climate Action through the EniQmA project (funding number 01MQ22007A).

References

1. T. Presles, C. Enderli, G. Burel, and E. H. Baghious, "Synthetic aperture radar image segmentation with quantum annealing," *IET Radar, Sonar & Navigation*, 2024.
2. S. M. Venkatesh, A. Macaluso, M. Nuske, M. Klusch, and A. Dengel, "Q-seg: Quantum annealing-based unsupervised image segmentation." *IEEE Computer Graphics and Applications*, 2024.
3. A. Geng, A. Moghiseh, C. Redenbach, and K. Schladitz, *arXiv:2307.16723*, 2023.
4. N. Otsu *et al.*, *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.
5. B. Singh, S. Majumdar, and S. Indu, *Quantum Studies: Mathematics and Foundations*, vol. 11, no. 3, pp. 427–458, Oct 2024.
6. M. Moore and A. Narayanan, "Quantum-inspired computing," *Dept. Comput. Sci., Univ. Exeter, Exeter, UK*, 1995.

7. L. Huynh, J. Hong, A. Mian, H. Suzuki, Y. Wu, and S. Camtepe, *arXiv:2308.11269*, 2023.
8. I. Peschel, *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2004, no. 06, p. P06004, jun 2004.
9. N. Palaiodimopoulos, V. F. Rey, M. Tschöpe, C. Jörg, P. Lukowicz, and M. Kiefer-Emmanouilidis, in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 116–120.
10. S. Dutta, A. Basarab, B. Georgeot, and D. Kouamé, *IEEE Open Journal of Signal Processing*, vol. 2, pp. 190–206, 2021.
11. E. Abrahams, *50 years of Anderson Localization*. world scientific, 2010, vol. 24.
12. P. W. Anderson, *Phys. Rev.*, vol. 109, pp. 1492–1505, Mar 1958.
13. N. Palaiodimopoulos, J. Frkatovic, V. F. Rey, M. Tschöpe, S. Suh, P. Lukowicz, and M. Kiefer-Emmanouilidis, *arXiv:2409.16180*, 2024.
14. A. Mafi and J. Ballato, "Review of a decade of research on disordered anderson localizing optical fibers," *Frontiers in Physics*, vol. 9, 2021.
15. N. E. Palaiodimopoulos, M. Kiefer-Emmanouilidis, G. Kurizki, and D. Petrosyan, *SciPost Phys. Core*, vol. 6, p. 017, 2023.
16. A. Das and B. K. Chakrabarti, *Rev. Mod. Phys.*, vol. 80, pp. 1061–1081, Sep 2008.
17. E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, *arXiv:0001106*, 2000.
18. H. Neven, G. Rose, and W. G. Macready, *arXiv:0804.4457*, 2008.
19. N. Dattani, S. Szalay, and N. Chancellor, *arXiv:1901.07636*, 2019.
20. O. Ronneberger, P. Fischer, and T. Brox, in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.
21. R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.
22. S. Barbosa, M. Kiefer-Emmanouilidis, F. Lang, J. Koch, and A. Widera, *Phys. Rev. Res.*, vol. 6, p. 033039, Jul 2024.
23. W. S. Bakr, J. I. Gillen, A. Peng, S. Fölling, and M. Greiner, *Nature*, vol. 462, no. 7269, pp. 74–77, 2009.
24. S. M. Venkatesh, A. Macaluso, M. Nuske, M. Klusch, and A. Dengel, *arXiv:2405.14405*, 2024.

Fast semantic segmentation CNNs for FPGAs

Simon Wezstein[12], Muen Jin[1], Michael Stelzl[2], and Michael Heizmann[1]

[1] Karlsruhe Institute of Technology, Institute of Industrial Information Technology,
Hertzstraße 16, 76187 Karlsruhe, Germany
[2] MSTVision GmbH,
Im Weiherfeld 10, 65462 Ginsheim-Gustavsburg, Germany

Abstract In this contribution small semantic segmentation CNNs are evaluated against traditional segmentation approaches and state of the art segmentation CNNs. The CNNs are optimized for the implementation on frame grabber FPGAs. A dataset of industrial burner flames and a dataset of transparent plastic granules is used to assess the segmentation performance of the models. VisualApplets by Basler AG is used to implement the models on an FPGA. The implemented models reach foreground IoU values of up to 96.7 %. The inference of a 552 x 552 pixel image takes slightly more than 1 ms. The latency between the start of an input line to the output of the line is 0.1 to 1.9 ms for streaming an 8192 pixel wide image.

Keywords Image signal processing, FPGA, CNN, segmentation

1 Introduction

Segmentation is a common task in image processing. There are many methods of segmentation available, from simple global thresholds to deep neural networks. One common use case in industrial image processing is to combine semantic segmentation with a Binary Large Object (BLOB) analysis to form an object detection algorithm. There are many more applications, often dependent on segmentation: measurement of objects in images, classification of objects, motion detection and

tracking, etc. Semantic segmentation may be seen as pixel-wise classification in an image. With semantic segmentation an image's pixels may be classified into various classes. Semantic segmentation with neural networks (NNs) recently gained big attention for complex tasks like autonomous driving and many other tasks with high variance regarding the imaging scene. any networks for semantic segmentation use convolutional filters, they are called Convolutional Neural Networks (CNN). In our work, we only refer to a binary segmentation, thus the classification in foreground and background.

Our former work on hybrid image processing with Field Programmable Gate Arrays (FPGAs) for low latency and high throughput applications ([1], [2]) was concentrated on balancing the computing load between CPUs, GPUs and FPGAs for optimized real time capability and image resolutions in sensor-based sorting. In our current architecture the FPGA is used for object detection and tracking and the GPU for object classification. The semantic segmentation in the object detection stage is realized by a global threshold operation. With this concept we are able to reduce the load on the PC host which allows low latencies and high raw image data throughput. Prior investigation and the correspondence with potential customers showed that simple rule-based approaches are often not powerful enough to fulfill the task. Employing CNNs for segmentation in a GPU would break our system architecture and running the whole raw image data through an NN would break the tight latency constraints (5 ms camera to actuator).

In our system design a frame grabber with an FPGA is always present. The approach is to develop simple yet sophisticated enough NNs to fit on this FPGA hardware as a drop-in replacement for the currently used global threshold. Many NNs are designed to fulfill more complex tasks than most of those in sensor-based sorting or industrial image processing in general. We seek to fill this gap. In industrial image processing, the imaging scene can be well controlled, which should allow the usage of simpler models in terms of parameters and operations, than the common state of the art ones. We want to optimize them for line scan cameras under low latency and high throughput demands.

2 Resources and Methods

Compared to GPU or CPU based development, on an FPGA the defined operations are configured in hardware instead of being broken down into machine code and being executed sequentially. Therefore all operations and parameters must fit into the FPGA's resources. The FPGA design is built with VisualApplets (VA), a proprietary platform by Basler AG for their frame grabbers [3]. Due to its exclusive use for the FPGA implementations at MSTVision GmbH, the set of possible operations is limited to the ones available in VA. The absolute hardware constraints lead to the unusual development strategy: "Which operations can be used and how many of them". We seek to find a sweet spot between model accuracy, hardware occupation and throughput/latency.

All currently available Basler frame grabber FPGA hardware is limited to integer arithmetics, forcing us to use quantized models. We use the Basler imaFlex CXP-12 Quad frame grabber for our experiments [4]. It is equipped with a Xilinx Ultrascale+ KU3P FPGA and 1.5 GB DRAM [5]. It has 160679 lookup tables (LUTs), 323224 flip flops (FFs), 720 block ram (BRAM) cells with 18 KiB each and 1368 48 bit digital signal processing (DSP) units.

For quantization aware training of our networks, QKeras [6] in conjunction with Keras [7] and TensorFlow [8] is used. The models are trained on an Nvidia RTX3080 GPU.

2.1 Available operations and limitations

The operator set of VA is limited to basic image and signal processing operations. These include: base arithmetics, convolution, image upscaling, lookup tables (LUTs), histograms, counters, BLOB detector, etc. Additionally there are many operations for data flow control like first in first out (FIFO) buffers, pipeline synchronisation, etc. Common operations in CNNs like matrix multiplication, activation functions, pooling, etc. are missing. If needed, they have to be implemented from scratch using the available operations. A complete list of available operators can be retrieved from [9].

2.2 Proposed models

Due to the limitations in hardware resources, implemented operations in VA and possible computation latency, we aim to build the models as simple and lightweight as possible. Our models need to be able to be trained from scratch to avoid legal problems with foreign datasets prohibiting industrial usage. For example the ImageNet dataset is restricted to non-commercial use [10].

Our most simple model (fig. 1) is a two layer convolutional model:

1. Convolution with 5x5 kernel, from 1 channel to 16 channels
2. Quantized ReLu
3. Convolution with 5x5 kernel, from 16 channel to 1 channel
4. Quantized ReLu

This forms the baseline of complexity and parameter count.

All other models are simple convolution only encoder-decoder-structures (fig. 2(a) and 2(b)). They consist of 2D convolutions, ReLu activations, max pooling, 2D transposed convolutions and upsampling with nearest neighbor interpolation. All models are quantized to 8 bit integer representation. The various tested models have varying filter sizes and encoder/decoder layer count. One part of the networks runs upsampling before transposed convolution, the other part after. Using upsampling after transposed convolution reduces the bandwidth to be processed in transposed convolution. Using smaller filter sizes reduces the amount of parameters. Tuning these parameters allows greater depth. The models, except the baseline model, use 8 channels and pooling/upsampling by the factor of 2 in the intermediate layers. All models are listed with their parameters in tab. 1. The model shown in fig. 2(a) consists of the following operations:

1. Convolution block with 5x5 kernel, 1 input and 8 output channels
   a) Convolution with 8 bit integer mask and input
   b) Offset with 16 bit values
   c) Rounding to 8 bit integers
   d) ReLu

2. Max pooling
3. Convolution block with 5x5 kernel and 8 channels
4. Max pooling
5. Upsampling
6. Transposed convolution block with 5x5 kernel and 8 channels
7. Upsampling
8. Transposed convolution block with 5x5 kernel and 1 channel

To save FPGA resources, all convolutions except the first and last layer of a model are carried out sequentially. This should pose only a minor impact on throughput and latency due to the reduced data rate after the max pooling operation. Other common operations like skip connections, fully connected layers, etc. were not considered due to their hardware requirements and/or implementation complexity.

Table 1: The models evaluated. ”Type” denotes the structure of the model with the abbreviations : ”c” for convolution, ”e” for encoder layer, ”d” for decoder layer and ”d (c)” for a decoder layer with convolutions instead of transposed convolutions. ”K” denotes the kernel size, e.g. 5 by 5. ”Ups.” denotes the order of upsampling operations: before the transposed convolution or after it. ”Para Cnt” denotes the number of model parameters.

Model	Type	K.	Ups.	Para Cnt
Base	2 Conv.	5	-	817
4 layer 5 a	2 e, 2 d	5	before	3625
4 layer 5 b	2 e, 2 d	5	after	3625
6 layer 5 a	3 e, 3 d	5	before	6841
6 layer 3 a	3 e, 3 d	3	before	2489
6 layer 5 b	3 e, 3 d	5	after	6841
6 layer 3 b	3 e, 3 d	3	after	2489
6 layer 3 a c	3 e, 3 d (c)	3	before	2489
6 layer 3 b c	3 e, 3 d (c)	3	after	2489

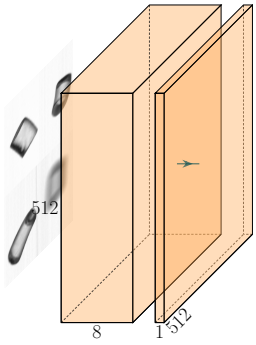


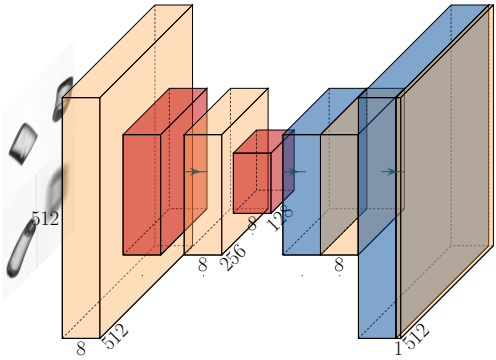
Figure 1: 2 layer baseline network. Generated with [11].

2.3 Data

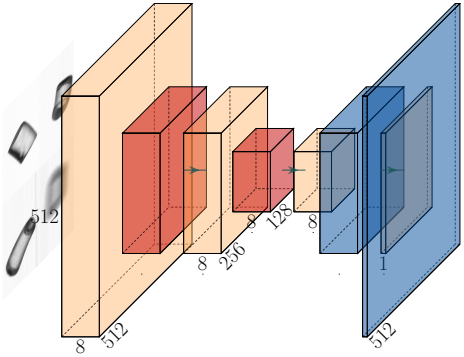
We use two industrial datasets to compare the proposed model's performances. The first dataset is the refined industrial burner dataset published in [12], [13]. It contains images with a resolution of 552×552 pixels. We use the dataset without augmentation to compare our results with theirs. They provide two datasets, "DataA" and "DataB", we use the first for our experiments, see fig. 3 for an example image. We swapped the test (160 images) and train (40 images) folder as they seem to be accidentally swapped.

The dataset has no predefined test subset, we use the validation set for testing.

The second dataset is a transparent plastic granule dataset based on our own data. The raw data was generated with a 16384 pixel wide line scan camera in a transmitted light setup. The granules to scan were poured on a slide while the camera was triggered at a line rate of 100 kHz. The raw data was filtered with a global threshold to remove most of the empty images. Segment Anything Model (SAM) [14] was used to generate masks for the granules. The masks were manually refined, the objects cropped to single 256×256 pixel images and randomly stitched to 512×512 pixel images (fig. 4). For training we use a 60/20/20 percent split. The stitched dataset, which is used for training, consists of 1004 images.



(a) 4 layer encoder-decoder-architecture with upsampling before transposed convolution.



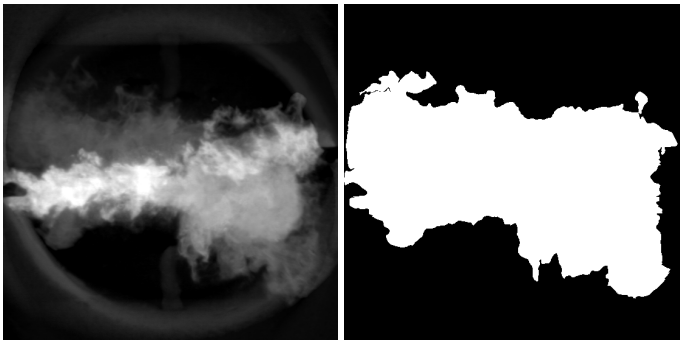
(b) 4 layer encoder-decoder-architecture with upsampling after transposed convolution.

Figure 2: 4 layer examples of the proposed encoder-decoder-architecture. The evaluated models vary in convolution kernel and pooling/upsampling sizes and in layer count. Generated with [11].

2.4 Evaluation workflow

For both datasets, each model is trained 10 times to gain statistics about the reached model performance. The models are trained from scratch with their default initialization defined by QKeras.

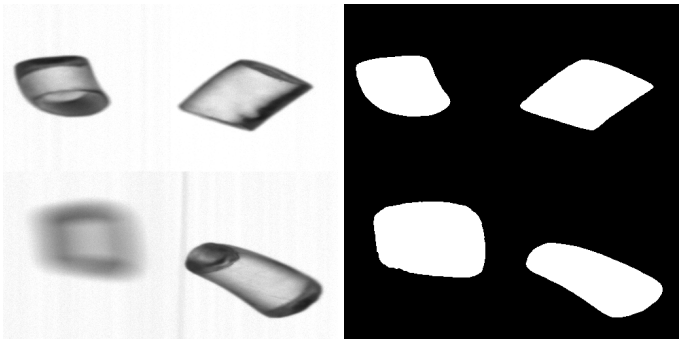
Training parameters:



(a) Industrial burner dataset image data example. Resolution: 552 x 552 pixel.

(b) Industrial burner dataset ground truth example. Resolution: 552 x 552 pixel.

Figure 3: Example image pair of the industrial burner dataset. [13, "DataA", image/mask 172]



(a) Image data example, consisting of four cropped granule samples. Resolution: 512 x 512 pixel.

(b) Segmentation ground truth example, consisting of four cropped granule samples. Resolution: 512 x 512 pixel.

Figure 4: Example image pair of our granule dataset.

- Batch size: 32
- Epoch count: 1500 for granules, 5000 for burner flames

- Optimizer: Adam
- Loss function: Binary Cross Entropy

Based on the captured statistics and the estimated performance, the best performing parameters are picked and implemented with VisualApplets. Timing measurements are implemented, too. The FPGA simulation results are compared to the results of QKeras. After synthesis, a 8192 x 512 pixel test image is uploaded to the FPGA and processed. The timing data is then evaluated.

3 Results

The test results of the best training run for each model are shown in table 2. All models perform better than the global threshold experiment, which yielded a foreground class intersection over union (IoU) of 81.4 % (test set of ”Data A”) and 80.2 % (test set of ”Data B”) for the burner flames and 51.5 % for the granules, except the baseline model which is below for the burner flames. We perform a grid search like [12] did. We consider our result of ”Data B” for our comparison, because of the same result in [12, tab. 1].

Due to problems in the implementation of transposed convolutions with VA, all models were trained with normal convolutions in the decoder layers. The models which were not implemented for the FPGA use transposed convolutions and are listed for comparison. The results for the FPGA implementations are listed in table 3.

Our best model on the burner dataset is ”6 layer 3 b c” with a mean IoU of 92.8 % and a foreground class IoU of 89.7 %. We implemented this architecture on the FPGA, see tab. 3. Our best model on the granules dataset is ”6 layer 5 a” with a mean IoU of 97.9 % and a foreground class IoU of 96.5 %. The best model which we could implement on the FPGA is ”6 layer 3 b c” with a mean IoU of 96.7 % and a foreground class IoU of 94.7 %. In comparison to the reference results of [12, tab. 1], most of our models perform better than their traditional machine learning models, with 86.6 % at best for an MLP. Our models perform worse compared to their neural networks with their worst results at 91.9 % (U-Net (MN)) and their best at 92.3 % (DL3+ (RN101)). We only consider their results for training from scratch as we did. The results

for the granule dataset show even better IoU values compared to the burner dataset results. This shows the potential of our models for segmentation in granule sorting.

In terms of inference time, the baseline model and the two 6 layer models take roughly 1 ms while the others take more time. This is due to the sequential calculation of the 8 channels while the amount of data is only reduced to 1/4 of the input bandwidth. The inference of the big image drops the throughput to around half the bandwidth. This behavior requires further investigation. Performing the upsampling operation after the convolution has positive effects for the IoU and for throughput. The simulation shows small differences between PC inference and FPGA inference. We suspect rounding problems as root case.

Table 2: The intersection over union (IoU) results of the models segmentation performance. "B" denotes the industrial burner dataset. "G" denotes the granule dataset. "FG" denotes the foreground class, "Mean" the mean IoU of background and foreground class. All values in %.

Model	Mean IoU B	IoU FG B	Mean IoU G	IoU FG G
Global Threshold	86.3	80.2	69.8	51.5
Base	87.7	82.4	79.7	67.4
4 layer 5 a	92.2	88.8	93.7	89.9
4 layer 5 b	92.7	89.5	96.1	93.6
6 layer 5 a	92.6	89.4	**97.9**	**96.5**
6 layer 3 a	92.3	89.0	94.8	91.5
6 layer 5 b	92.6	89.5	97.7	96.3
6 layer 3 b	92.7	89.6	96.8	94.8
6 layer 3 a c	92.5	89.3	95.5	92.6
6 layer 3 b c	**92.8**	**89.7**	96.7	94.7

4 Conclusion

We showed the potential of low parameter models for the usage in semantic segmentation with FPGAs. The models perform better than initially expected, superseding the traditional machine learning methods of [12] while having more throughput and lower latencies. Having an additional latency between 0.105 ms and 1.904 ms, the models and

Table 3: The throughput/latency and resource occupation results of the FPGA implementations. "L2L" denotes the time between processing the first pixel of a line and retrieving the first processed pixel of that line using the 8192 pixel wide test image. "Time B" denotes the inference time for a single 552 x 552 pixel image of the burner dataset. "LUT, FF, DSP and BRAM" show the relative resource consumption of the model on the FPGA.

Model	L2L [ms]	Time B [ms]	LUT [%]	FF [%]	DSP [%]	BRAM [%]
Base	0.105 ± 2.3e-3	1.041	48.83	32.33	2.41	37.78
4 layer 5 a	1.780 ± 0.14	2.755	28.29	31.27	85.31	78.89
4 layer 5 b	1.904 ± 0.16	2.074	28.41	31.3	85.31	61.25
6 layer 3 a c	1.393 ± 0.13	1.067	65.46	40.89	37.79	38.61
6 layer 3 b c	1.669 ± 0.17	1.058	65.68	40.68	37.79	32.5

implementations are considerable candidates for line scan applications. Future work will target the rounding problems in the FPGA implementation. Because of the usage of quantization aware training, we suspect the FPGA to have exactly the same output as computed on the PC. In addition the throughput decrease for large images needs will be investigated, too. We expect to be able to increase the throughput and parameter count further with bigger FPGAs, hopefully available in the near future. Future work will target the implementation of more sophisticated CNN-operations, too.

References

1. S. Wezstein, M. Stelzl, and M. Heizmann, "Latency evaluation of an FPGA-based sorting system," in *9th Sensor-Based Sorting & Control 2022*, K. Greiff, H. Wotruba, A. Feil, N. Kroell, X. Chen, D. Gürsel, and V. Merz, Eds., 04 2022, pp. 143–160.
2. S. Wezstein, O. Gräff, M. Stelzl, and M. Heizmann, "Latency evaluation of a CNN enhanced FPGA-based sorting system," in *10th Sensor-Based Sorting & Control 2024*, ser. Sensor-Based Sorting & Control, K. Greiff, A. Feil, L. Weitkämper, N. Kroell, T. Scherling, D. Gürsel, and V. Merz, Eds., vol. 10. Düren: Shaker Verlag, 03 2024, pp. 67–88.
3. Basler AG, "VisualApplets Graphical FPGA Programming," https://www.baslerweb.com/en/software/visualapplets/, 2024, online, accessed 21-September-2024.

4. ——, “imaFlex CXP-12 Quad,” https://www.baslerweb.com/en/shop/imaflex-cxp-12-quad/, 2024, online, accessed 21-September-2024.
5. ——, “VisualApplets User Manual,” https://docs.baslerweb.com/visualapplets/files/manuals/content/device_resources.html, 2024, online, accessed 21-September-2024.
6. C. N. C. J. au2, A. Kuusela, S. Li, H. Zhuang, T. Aarrestad, V. Loncar, J. Ngadiuba, M. Pierini, A. A. Pol, and S. Summers, “Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors,” 2021. [Online]. Available: https://arxiv.org/abs/2006.10159
7. F. Chollet *et al.*, “Keras,” https://keras.io, 2015.
8. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/
9. Basler AG, “VisualApplets User Manual,” https://docs.baslerweb.com/visualapplets/files/manuals/content/operator_documentations.html, 2024, online, accessed 21-September-2024.
10. Stanford Vision Lab, Stanford University and Princeton University, “ImageNet,” https://www.image-net.org/, 2021, online, accessed 21-September-2024.
11. H. Iqbal, “Harisiqbal88/plotneuralnet v1.0.0,” Dec. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.2526396
12. S. Landgraf, M. Hillemann, M. Aberle, V. Jung, and M. Ulrich, “Segmentation of industrial burner flames: A comparative study from traditional image processing to machine and deep learning,” Tech. Rep., 2023.
13. S. Landgraf, M. Hillemann, M. Ulrich, M. Aberle, and V. Jung, “Dataset for the segmentation of industrial burner flames,” 2023. [Online]. Available: https://publikationen.bibliothek.kit.edu/1000159497
14. A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment anything,” *arXiv:2304.02643*, 2023.

Semantic segmentation and uncertainty quantification with vision transformers for industrial applications

Kira Wursthorn[1], Lili Gao[2], Steven Landgraf[1], and Markus Ulrich[1]

[1] Karlsruhe Institute of Technology (KIT), Institute of Photogrammetry and Remote Sensing (IPF), Englerstr. 7, 76131 Karlsruhe
[2] Torc Robotics, Augsburger Str. 540, 70327 Stuttgart

Abstract Vision Transformers (ViTs) have recently achieved state-of-the-art performance in semantic segmentation tasks. However, their deployment in critical applications necessitates reliable uncertainty quantification to assess model confidence. To tackle this challenge, we combine a state-of-the-art ViT with the popular uncertainty quantification method Monte Carlo Dropout (MCD) to predict both segmentation and uncertainty maps. We focus on an industrial machine vision setting and carry out the experiments on the T-LESS dataset. The evaluation is carried out with regard to both the segmentation accuracy and the predicted uncertainties using appropriate metrics.

Keywords Semantic segmentation, uncertainty quantification, vision transformers

1 Introduction

In computer vision, deep-learning-based approaches like convolutional neural networks (CNNs) have proven their success at solving the fundamental task of semantic segmentation of (RGB) images. Recently, Vision Transformers (ViTs) have been applied to this task and have gained much attention. The prediction of pixel-wise class labels in images is relevant for applications such as autonomous driving, and quality assurance in industry. These applications involve safety-critical

and high-risk scenarios. Therefore, it is important to not only predict the class labels correctly but also to determine the prediction's reliability [1–4]. Estimating uncertainty of predictions allows to make informed decisions and to identify potentially inaccurate predictions.

Most classification and segmentation tasks use softmax to estimate class-wise pseudo probabilities to quantify the confidence in the predictions. It is well-known that softmax predictions tend to be overconfident, especially in cases where the input data of the model is out-of-domain [5, 6]. One popular method to quantify uncertainty in deep learning is Monte-Carlo Dropout (MCD) [7] that uses dropout at inference time. Multiple forward passes are used to sample from the posterior distribution of the predictions and approximate it, e.g., with a Gaussian distribution. The final segmentation map is determined by assigning each pixel the class with the highest average softmax output across all classes. The corresponding uncertainty map is either its standard deviation (STD) over the samples or the entropy of the mean values over the classes.

In this contribution, we combine a state-of-the-art ViT, the SegFormer [8], with MCD for semantic segmentation with uncertainty quantification (UQ). We choose SegFormer as our ViT baseline because of its efficient design and good performance, which both are relevant criteria in industry. Our goal is to quantify the quality and reliability of the SegFormer's predicted semantic segmentation maps as well as the corresponding uncertainty maps for industrial applications. Therefore, we train the model on the T-LESS [9] dataset that consists of various scenes of parts with characteristics that are typical for industry. As part of the Benchmark for 6D Object Pose Estimation (BOP) [10], the T-LESS training set can be augmented with physically-based rendered (PBR) synthetic training data. While the real training images show systematically captured and isolated views of each object respectively, the PBR subset consists of cluttered scenes with varying image acquisition conditions, scene backgrounds, and occlusions by both T-LESS objects and those of other BOP datasets. Figure 1 shows two examples of the T-LESS dataset from both a simple as well as a cluttered scene together with the corresponding segmentation and uncertainty maps that our trained uncertainty-aware SegFormer model predicted. We use the mean Intersection over Union (IoU) and the expected calibration error (ECE) [11] as metrics to measure the segmentation quality and model

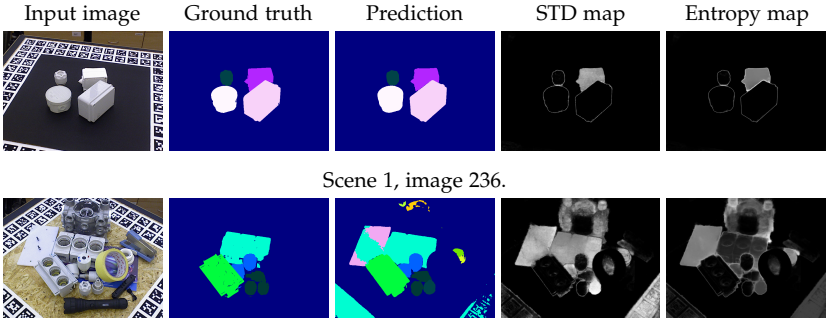


Figure 1: Example predictions of segmentation and uncertainty maps for images from a simple (top row) and a complex scene (bottom row) of the T-LESS test dataset, using the MCD with a dropout rate of 30 % and 20 samples. In the uncertainty maps, brighter pixels represent higher uncertainty values.

calibration and the Patch Accuracy versus Patch Uncertainty (PAvPU), $p(\textbf{accurate}|\textbf{certain})$, and $p(\textbf{uncertain}|\textbf{inaccurate})$ [12] for the uncertainty evaluation.

After giving a short overview over the state-of-the-art approaches for semantic segmentation with ViTs and uncertainty quantification in Section 2, we explain our training and evaluation methodology in Section 3. In Section 4, we describe our experiments and present our results, which are discussed in Section 5. Section 6 concludes our paper.

2 Related Work

Due to the success of ViTs for image classification, many publications have been dedicated to applying the method to the task of semantic segmentation. Next to SegFormer, notable approaches include Segmenter [13], SETR [14], MaskFormer [15] and its successor Mask2Former [16] as well as general ViT approaches for dense predictions like Swin Transformer [17], DPT [18], and HRFormer [19].

Regarding UQ in RGB image-based semantic segmentation tasks, many works have successfully integrated MCD in their workflows, including applications like landcover prediction from remote sensing

images [20], medical imaging [21], autonomous driving, and robotics [22–24]. To overcome the disadvantage of the additional runtime of sample-based UQ methods, knowledge distillation can be applied [25].

Recently, successful efforts have been made to combine SegFormer with UQ. While Chen et al. [26] propose their own UQ approach and compare its performance against MCD and ensembling using SegFormer, Landgraf et al. [27] add monocular depth estimation and and UQ with MCD to the SegFormer architecture. Both works conduct their experiments in the context of autonomous driving.

3 Methodology

Our methodology aims to achieve two main goals: i) Training and testing a SegFormer model to achieve the best possible segmentation performance on T-LESS, and ii) combining SegFormer with MCD for UQ. Both the segmentation and the uncertainty results are evaluated by their respective metrics (see below). The first goal provides a basic training setup, including suitable hyperparameters such as learning rate, model backbone, dataset settings, and data augmentations. This also leads to a baseline model without UQ. Next to testing the segmentation quality of the baseline model, it also includes the evaluation of the mean segmentation maps of the trained MCD models and the influence of performing dropout at inference time. For this, the mean IoU and the ECE metrics are used. The second goal that focuses on the UQ with SegFormer includes model training with different dropout rates for MCD and the evaluation of the predicted uncertainty maps with different sample sizes.

The uncertainty evaluation metrics proposed by Mukhoti and Gal (2018) [12] are computed based on the confusion matrix that includes four categories of pixel counts: accurate and certain (n_{ac}), accurate and uncertain (n_{au}), inaccurate and certain (n_{ic}), and inaccurate and uncertain (n_{iu}). To determine whether a prediction is certain or uncertain, an uncertainty threshold has to be defined. Here, we use the mean uncertainty over all pixels across the T-LESS test dataset. Based on the estimated counts, two metrics are computed that are defined as $p(\textbf{accurate}|\textbf{certain}) = n_{ac}/(n_{ac} + n_{ic})$ and $p(\textbf{uncertain}|\textbf{inaccurate}) = n_{iu}/(n_{ic} + n_{iu})$. The former returns higher values if predictions are ac-

curate when the model is certain. The latter returns higher values if the model is uncertain when the predictions are inaccurate. Consequently, meaningful uncertainty values lead to large values for both metrics. Furthermore, the third metric PAvPU $= (n_{ac} + n_{iu})/(n_{ac} + n_{au} + n_{ic} + n_{iu})$ combines the first two metrics and, hence, presents an equivalent UQ metric to an overall accuracy. In the following, the metrics $p(\textbf{accurate}|\textbf{certain})$ and $p(\textbf{uncertain}|\textbf{inaccurate})$ are abbreviated as p_{ac} and p_{ui}.

4 Experiments

To address our first goal described in Section 3, we test different combinations of hyperparameters and training settings. We find that the best model performance in terms of mean IoU on the BOP test dataset of T-LESS is achieved by combining both real and PBR training data, a SegFormer-B5 backbone, and a learning rate of $6 \cdot 10^{-5}$. The combination of real and synthetic training data increases the mean IoU by roughly 50 %. Thus, we train all models in our experiments on both training data subsets. Similarly, a subsequent increase in the size of the backbone from B1 to B5 leads to increasing mean IoU scores and decreasing ECE values. For instance, replacing the smaller SegFormer-B1 architecture with the larger SegFormer-B5, which has the highest parameter count, leads to a 19.23 % increase in mean IoU and a 5.38 % reduction in ECE, as shown in Table 1. Thus, we select SegFormer-B5 for testing different subsets of data augmentation techniques of the AugSeg [28] framework. AugSeg includes geometric augmentations (random flip, random scale, and random crop) as well as a list of intensity-based augmentations (e.g., blurring, brightness and contrast modifications). The hyperparameter k denotes how many intensity-based augmentation techniques are randomly selected for each training instance. The results in terms of mean IoU and ECE are shown in Table 1.

We find that a combination of the geometric augmentations and the random intensity-based augmentations with a random selection parameter $k = 3$ works best, both in terms of highest mean IoU of 79.40 % and lowest ECE value of 4.29 %. We also test different learning rates where a learning rate of $1 \cdot 10^{-4}$ achieves the best results of a mean

IoU of 80.70 % and an ECE of 2.88 %. As learning rates higher than $2 \cdot 10^{-4}$ lead to model divergence during in our experiments at training time, we adopted the learning rate of $6 \cdot 10^{-5}$ of the original SegFromer publication to guarantee a stable training procedure. For a better comparison, all models are trained for 100 epochs on a NVIDIA H100 hardware.

Table 1: Ablation study using different model backbones and geometric and intensity-based data augmentation techniques with different values of the random selection parameter k from AugSeg [28].

	Augmentations		Metrics in %	
Model	Geometric	Intensity-based	Mean IoU ↑	ECE ↓
SegFormer-B1	-	-	44.69	9.92
SegFormer-B5	-	-	63.92	8.92
	✓	-	74.81	6.65
	✓	$k = 1$	76.35	6.07
	✓	$k = 3$	**79.40**	**4.29**
	✓	$k = 5$	79.22	5.30

In order to incorporate MCD for the second goal of UQ, we activate the implemented but dormant dropout layers in the SegFormer architecture. We train the models with dropout rates of 10 %, 20 %, 30 %, and 50 % resulting in four different models. In contrast to dropout regularization, the dropout layers remain active for MCD at test time to obtain samples. We evaluate each model with sample sizes of $N = \{2, 5, 10, 20, 100\}$ respectively and compare them using both mean IoU and ECE for segmentation quality and p_{ac}, p_{ui}, and PAvPU for uncertainty quality. The results are summarized in Table 2.

Our evaluations show that smaller dropout rates lead to a higher mean IoU but not necessarily to lower ECE values. With regard to UQ metrics, all models achieve similar scores. Furthermore, increasing values for N result in increasing values in p_{ac} and p_{ui}, as expected. However, they surprisingly also result in slightly lower PAvPU scores. This is caused by decreasing counts of n_{ac} with increasing N. Nevertheless, these changes in PAvPU as well as in mean IoU and ECE are not substantial as they are all smaller than 3 %. In terms of required runtime, the minimum sample size of $N = 2$ takes around 89 ms while $N = 100$ results in 3711 ms runtime. Therefore, in time-critical applications, it should be possible to decrease N in order to speed-up the application without sacrificing too much predictive quality. For exam-

ple, an uncertainty-aware prediction with $N = 20$ takes less than a second at 751 ms.

Table 2: Performance of SegFormer-B5 with MCD. Tested were different dropout rates and sample sizes N. The results were evaluated in terms of both the segmentation and uncertainty quality using the respective metrics described in Section 3. The subscript "std" indicates that the metrics are based on standard deviation, while the subscript "en" indicates that the metrics are based on entropy. All metrics are in %.

N	$p_{ac,\text{std}} \uparrow$	$p_{ui,\text{std}} \uparrow$	$\text{PAvPU}_{\text{std}} \uparrow$	$p_{ac,\text{en}} \uparrow$	$p_{ui,\text{en}} \uparrow$	$\text{PAvPU}_{\text{en}} \uparrow$	Mean IoU ↑	ECE ↓
dropout rate = 10 %								
2	98.88 ± 0.01	74.09 ± 0.15	92.41 ± 0.03	99.49 ± 0.01	88.14 ± 0.08	90.48 ± 0.03	76.93 ± 0.08	5.75 ± 0.13
5	99.28 ± 0.01	81.76 ± 0.11	91.74 ± 0.02	99.53 ± 0.01	88.60 ± 0.07	90.25 ± 0.02	77.13 ± 0.06	5.68 ± 0.11
10	99.34 ± 0.01	83.00 ± 0.07	91.42 ± 0.02	99.55 ± 0.01	88.83 ± 0.05	90.14 ± 0.02	77.24 ± 0.06	5.64 ± 0.10
20	99.38 ± 0.01	83.64 ± 0.07	91.17 ± 0.02	99.56 ± 0.01	88.96 ± 0.04	90.07 ± 0.02	77.25 ± 0.05	5.62 ± 0.09
100	99.43 ± 0.00	84.44 ± 0.03	90.82 ± 0.01	99.57 ± 0.00	89.12 ± 0.03	90.01 ± 0.01	77.25 ± 0.03	5.63 ± 0.02
dropout rate = 20 %								
2	98.25 ± 0.02	73.16 ± 0.20	91.57 ± 0.05	99.09 ± 0.01	87.94 ± 0.11	89.68 ± 0.04	75.51 ± 0.12	6.09 ± 0.13
5	98.78 ± 0.01	81.70 ± 0.15	90.78 ± 0.05	99.19 ± 0.01	88.77 ± 0.09	89.35 ± 0.04	75.95 ± 0.09	6.12 ± 0.14
10	98.93 ± 0.01	83.27 ± 0.11	90.37 ± 0.03	99.24 ± 0.01	89.11 ± 0.08	89.17 ± 0.03	76.10 ± 0.07	6.10 ± 0.10
20	99.01 ± 0.01	84.14 ± 0.08	90.02 ± 0.02	99.27 ± 0.00	89.32 ± 0.05	89.03 ± 0.02	76.16 ± 0.07	6.06 ± 0.07
100	99.12 ± 0.01	85.23 ± 0.03	89.50 ± 0.02	99.30 ± 0.00	89.55 ± 0.02	88.93 ± 0.01	76.27 ± 0.04	6.00 ± 0.04
dropout rate = 30 %								
2	98.49 ± 0.02	74.39 ± 0.28	91.54 ± 0.04	99.28 ± 0.01	89.02 ± 0.13	89.58 ± 0.04	74.52 ± 0.17	5.57 ± 0.20
5	98.04 ± 0.01	83.35 ± 0.17	90.56 ± 0.04	99.39 ± 0.01	89.95 ± 0.09	89.15 ± 0.04	75.00 ± 0.15	5.51 ± 0.17
10	98.20 ± 0.01	85.09 ± 0.09	90.03 ± 0.03	99.45 ± 0.01	90.42 ± 0.08	88.90 ± 0.02	75.24 ± 0.11	5.45 ± 0.13
20	99.28 ± 0.01	86.09 ± 0.10	89.62 ± 0.03	99.48 ± 0.01	90.70 ± 0.07	88.75 ± 0.03	75.34 ± 0.09	5.42 ± 0.14
100	99.39 ± 0.01	87.36 ± 0.07	88.96 ± 0.02	99.51 ± 0.00	90.99 ± 0.04	88.61 ± 0.02	75.41 ± 0.04	5.43 ± 0.06
dropout rate = 50 %								
2	95.84 ± 0.03	66.93 ± 0.33	88.12 ± 0.04	97.48 ± 0.03	83.34 ± 0.13	86.93 ± 0.03	68.66 ± 0.22	7.78 ± 0.16
5	96.86 ± 0.05	77.47 ± 0.36	87.17 ± 0.08	97.70 ± 0.03	85.02 ± 0.18	86.40 ± 0.07	69.54 ± 0.12	8.05 ± 0.18
10	97.17 ± 0.03	80.04 ± 0.18	86.53 ± 0.07	97.81 ± 0.02	85.80 ± 0.13	86.08 ± 0.06	69.84 ± 0.12	8.17 ± 0.17
20	97.37 ± 0.02	81.65 ± 0.09	85.99 ± 0.04	97.89 ± 0.01	86.32 ± 0.07	85.87 ± 0.04	70.06 ± 0.11	8.15 ± 0.10
100	97.67 ± 0.02	83.99 ± 0.11	85.14 ± 0.04	97.96 ± 0.01	86.85 ± 0.06	85.77 ± 0.03	70.24 ± 0.04	8.17 ± 0.06

Figure 2 shows some qualitative results for different dropout rates and with $N = 20$ on an example image of a complex scene in the T-LESS test dataset. Next to the predicted segmentation and uncertainty maps, the accuracy and the binary uncertainty maps are shown. For the binary uncertainty maps, we applied the same mean uncertainty threshold mentioned in Section 3 that is used for the estimation of the UQ metrics. Overall, it shows that accurate pixel predictions correspond to low uncertainty patches and vice versa. Increasing dropout rates lead to higher uncertainty values, which can be seen in the binary uncertainty maps. In case of the 10 % dropout model, the falsely segmented object in the lower right part of the image and background pixels exhibit high uncertainties.

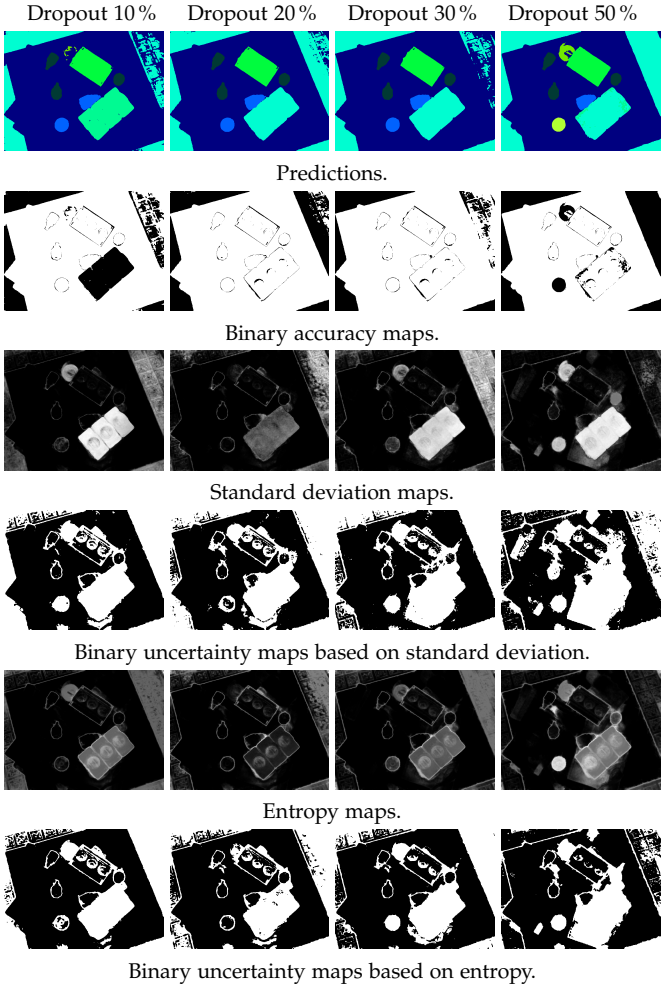


Figure 2: Comparison of uncertainty maps for the image from a complex scene (Scene 17, image 50) across different dropout rates. Predictions and uncertainties are generated with 20 samples. In uncertainty maps, brighter pixels represent higher uncertainty. In accuracy/uncertainty binary maps, white pixels represent accurate/uncertain pixels.

5 Discussion

Our experiments demonstrate that increasing the sample size generally improves the segmentation accuracy and calibration in terms of mean IoU and ECE, while also enhancing the reliability of uncertainty estimation, as indicated by higher p_{ac} and p_{ui} scores. However, PAvPU decreases with larger sample sizes due to an increase in accurately classified but uncertain pixels, n_{au}, suggesting a more cautious model that flags more pixels as uncertain. It has to be noted that the UQ metrics depend on the chosen uncertainty threshold used to generate the underlying confusion matrix as described in Section 3 and may therefore vary with different thresholds.

Lower dropout rates result in better segmentation accuracy and model calibration, with the best performance observed when dropout is deactivated. However, a 30 % dropout rate optimizes p_{ui}, which is critical for detecting potentially incorrect predictions while reducing the calibration and segmentation quality only by 1.19 % ECE and 4.30 % mean IoU on average compared to the baseline model of our first goal. Thus, a 30 % dropout rate balances accurate segmentation and effective uncertainty estimation, making it optimal for practical applications.

Entropy is identified as a more suitable uncertainty metric than standard deviation, as it provides higher p_{ui}, indicating a better capacity to flag incorrect predictions. Although entropy-based metrics slightly reduce PAvPU, the trade-off is justified by a significant improvement in detecting uncertain inaccuracies.

Overall, the results suggest that using 20 samples, a 30 % dropout rate, and entropy as the uncertainty metric provides an optimal configuration for balancing segmentation accuracy, calibration, and uncertainty quantification quality in the SegFormer model with MCD.

6 Conclusion

In this contribution, we successfully trained SegFormer, a ViT variant, on the T-LESS dataset for the task of semantic segmentation with UQ in an industrial application. In combination with MCD, SegFormer is able to effectively handle challenging objects in varying complex scenes while producing meaningful uncertainty estimates. In future work, we

want to extend the methodology for instance segmentation, which allows the integration of an ViT model in a deep-learning-based 6D object pose estimation pipeline. In the evaluation, we want to include additional UQ metrics like UCS [29,30]. While MCD is easy to implement, it does not capture the full uncertainty in the predictions [23]. Therefore, in future work, we aim to combine SegFormer with other state-of-the-art UQ methods like the recently proposed Deep Deterministic Uncertainty (DDU) [31] approach to produce robust uncertainty estimates even under data shift.

Acknowledgments

The authors acknowledge support by the state of Baden-Württemberg through bwHPC.

References

1. B. Ghoshal, A. Tucker, B. Sanghera, and W. Lup Wong, "Estimating uncertainty in deep learning for reporting confidence to clinicians in medical image segmentation and diseases detection," *Computational Intelligence*, vol. 37, no. 2, pp. 701–734, 2021.
2. M.-H. Laves, S. Ihler, J. F. Fast, L. A. Kahrs, and T. Ortmaier, "Well-calibrated regression uncertainty in medical imaging with deep learning," in *MIDL*, vol. 121, 2020, pp. 393–412.
3. D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in *ITSC*, 2018, pp. 3266–3273.
4. S. Shafaei, S. Kugele, M. H. Osman, and A. Knoll, "Uncertainty in machine learning: A safety perspective on autonomous driving," in *SAFECOMP 2018 Workshops*, 2018, pp. 458–464.
5. D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv e-prints*, vol. arXiv:1610.02136, 2016.
6. Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *NeurIPS*, vol. 32, 2019.

7. Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *ICML*, vol. 48, 2016, pp. 1050–1059.
8. E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *NeurIPS*, vol. 34, 2021, pp. 12 077–12 090.
9. T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects," in *IEEE WACV*, 2017, pp. 880–888.
10. T. Hodaň, Y. Labbé, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, "BOP: Benchmark for 6D Object Pose Estimation," in *ECCV*, 2018.
11. M. P. Naeini, G. Cooper, and M. Hausknecht, "Obtaining well calibrated probabilities using bayesian binning," in *AAAI*, vol. 29, 2015, pp. 2901–2907.
12. J. Mukhoti and Y. Gal, "Evaluating bayesian deep learning methods for semantic segmentation," *arXiv e-prints*, vol. arXiv:1811.12709, 2018.
13. R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *IEEE/CVF ICCV*, 2021, pp. 7262–7272.
14. S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *IEEE/CVF CVPR*, 2021, pp. 6881–6890.
15. B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *NeurIPS*, vol. 34, pp. 17 864–17 875, 2021.
16. B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *IEEE/CVF CVPR*, 2022, pp. 1290–1299.
17. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF IICCV*, 2021, pp. 10 012–10 022.
18. R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *IEEE/CVF ICCV*, 2021, pp. 12 179–12 188.
19. Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, and J. Wang, "Hrformer: High-resolution transformer for dense prediction," *arXiv e-prints*, vol. arXiv:2110.09408, 2021.

20. C. Dechesne, P. Lassalle, and S. Lefèvre, "Bayesian deep learning with monte carlo dropout for qualification of semantic segmentation," in *IEEE IGARSS*, 2021, pp. 2536–2539.
21. M. Abdar, S. Salari, S. Qahremani, H.-K. Lam, F. Karray, S. Hussain, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "Uncertaintyfusenet: robust uncertainty-aware hierarchical feature fusion model with ensemble monte carlo dropout for covid-19 detection," *Information Fusion*, vol. 90, pp. 364–381, 2023.
22. A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv e-prints*, vol. arXiv:1511.02680, 2015.
23. A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *NeurIPS 2017*, vol. 30, 2017.
24. S. Landgraf, M. Hillemann, K. Wursthorn, and M. Ulrich, "Uncertainty-aware cross-entropy for semantic segmentation," in *ISPRS Annals*, vol. X-2-2024, 2024, pp. 129–136.
25. S. Landgraf, K. Wursthorn, M. Hillemann, and M. Ulrich, "Dudes: Deep uncertainty distillation using ensembles for semantic segmentation," *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 92, no. 2, pp. 101–114, 2024.
26. B. Chen, W. Peng, X. Cao, and J. Röning, "Hyperbolic uncertainty aware semantic segmentation," *IEEE T-ITS*, vol. 25, no. 2, pp. 1275–1290, 2024.
27. S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich, "Efficient multi-task uncertainties for joint semantic segmentation and monocular depth estimation," in *GCPR*, 2024.
28. Z. Zhao, L. Yang, S. Long, J. Pi, L. Zhou, and J. Wang, "Augmentation matters: A simple-yet-effective approach to semi-supervised semantic segmentation," in *IEEE/CVF CVPR*, 2023, pp. 11 350–11 359.
29. K. Wursthorn, M. Hillemann, and M. Ulrich, "Uncertainty quantification with deep ensembles for 6d object pose estimation," in *ISPRS Annals*, vol. X-2-2024, 2024, pp. 223–230.
30. D. W. Wolf, P. Balaji, A. Braun, and M. Ulrich, "Decoupling of neural network calibration measures," in *GCPR*, 2024.
31. J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. Torr, and Y. Gal, "Deep Deterministic Uncertainty: A New Simple Baseline," in *IEEE/CVF CVPR*, 2023, pp. 24 384–24 394.

Evaluation of multi-task uncertainties in joint semantic segmentation and monocular depth estimation

Steven Landgraf, Markus Hilleman, Theodor Kapler, and Markus Ulrich

Karlsruhe Institute of Technology (KIT),
Institute of Photogrammetry and Remote Sensing (IPF),
Karlsruhe, Germany.

Abstract Deep neural networks achieve outstanding results in perception tasks such as semantic segmentation and monocular depth estimation, making them indispensable in safety-critical applications like autonomous driving and industrial inspection. However, they often suffer from overconfidence and poor explainability, especially for out-of-domain data. While uncertainty quantification has emerged as a promising solution to these challenges, multi-task settings still need to be investigated in this regard. In an effort to shed light on this, we evaluate Monte Carlo Dropout, Deep Sub-Ensembles, and Deep Ensembles for joint semantic segmentation and monocular depth estimation. Thereby, we reveal that Deep Ensembles stand out as the preferred choice and show the potential benefit of multi-task learning with regard to the uncertainty quality in comparison to solving both tasks separately.

Keywords Deep learning, uncertainty quantification, multi-task learning, semantic segmentation, monocular depth estimation

1 Introduction

Deep neural networks are increasingly being used in real-time and safety-critical applications like autonomous driving [1], industrial inspection [2], and automation [3]. Although they achieve incomparable

performance in fundamental perception tasks like semantic segmentation [4] or monocular depth estimation [5], they still suffer from problems like overconfidence [6], lack explainability [7], and struggle to distinguish between in-domain and out-of-domain samples [8].

In order to tackle these critical challenges and prevailing shortcomings of deep neural networks, a number of promising uncertainty quantification methods [9–12] have been proposed. Surprisingly, however, quantifying predictive uncertainties in the context of joint semantic segmentation and monocular depth estimation has not been thoroughly explored yet [13]. Since many real-world applications are multi-modal in nature and, hence, have the potential to benefit from multi-task learning, this is a substantial gap in current literature.

To this end, we conduct a comprehensive series of experiments to study how multi-task learning influences the quality of uncertainty estimates in comparison to solving both tasks separately. Our contributions can be summarized as follows:

- We combine three different uncertainty quantification methods - Monte Carlo Dropout (MCD), Deep Sub-Ensembles (DSE), and Deep Ensembles (DE) - with joint semantic segmentation and monocular depth estimation and evaluate how they perform in comparison to each other.
- In addition, we reveal the potential benefit of multi-task learning with regard to the uncertainty quality compared to solving semantic segmentation and monocular depth estimation separately.

2 Related Work

2.1 Joint Semantic Segmentation and Monocular Depth Estimation

Semantic segmentation and monocular depth estimation are both essential tasks in image understanding, requiring pixel-wise predictions from a single input image. Due to the strong correlation and complementary nature of these tasks, several previous works have focused on addressing them jointly [14–18].

Notably, almost all previous works employ out-of-date architectures and require complex adaptations to either the model, the training process, or both. Instead of following this trend, we adapt a modern

Vision-Transformer-based architecture similar to Xu et al. [18], achieving competitive predictive performance while maintaining simplicity and transparency of the results.

2.2 Uncertainty Quantification

In order to address the shortcomings of deep neural networks, a variety of uncertainty quantification methods [9–12] and studies [19–21] have been proposed. The predictive uncertainty can be decomposed into aleatoric and epistemic uncertainty [22], which can be an essential for applications like active learning and detecting out-of-distribution samples [23]. The aleatoric component captures the irreducible data uncertainty, such as image noise or noisy labels from imprecise measurements. The epistemic uncertainty accounts for the model uncertainty and can be reduced with more or higher quality training data [22, 24].

Remarkably, quantifying uncertainties in joint semantic segmentation and monocular depth estimation has been largely overlooked [13]. Therefore, we compare multiple uncertainty quantification methods for this task and show how multi-task learning influences the quality of the uncertainty quality in comparison to solving both tasks separately.

3 Evaluation Strategy

3.1 Baseline Models.

To explore the impact of multi-task learning on the uncertainty quality, we conduct our evaluations with three models:

1. SegFormer [25] for the segmentation task,
2. DepthFormer for the depth estimation task,
3. SegDepthFormer for joint semantic segmentation and monocular depth estimation.

SegFormer. For solving the semantic segmentation task by itself, we use SegFormer [25], a modern Transformer-based architecture. Due to its high efficiency and performance, it is particularly suitable for real-time applications that might rely on uncertainty quantification. We

train all SegFormer models with the categorical Cross-Entropy loss

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{n,c}\cdot\log(p(z)_{n,c}) \tag{1}$$

for a single image, where N is the number of pixels in the image, C is the number of classes, $y_{n,c}$ is the corresponding ground truth label, and $p(z)_{n,c}$ is the predicted softmax probability.

To obtain a measure for the aleatoric uncertainty [24] of the baseline model, we compute the predictive Entropy

$$H(p(z)) = -\sum_{c=1}^{C} p(z)_c\cdot\log(p(z)_c)\ . \tag{2}$$

DepthFormer. Highly inspired by the efficiency and performance of SegFormer [25], we propose DepthFormer for monocular depth estimation. We use the same hierarchical Transformer-based encoder and all-MLP decoder. In contrast to SegFormer, the output layer differs by having two output channnels: one for the predictive mean $\mu(z)$ and one for the predictive variance $s^2(z)$ [26]. The first output channel uses a ReLU output activation function, while the second output channel applies Softplus activation, which is a smooth approximation of the ReLU functon with the advantage of being differentiable at $z = 0$. We found Softplus to work better than ReLU for the predictive variance, following the work of Lakshminarayanan et al. [11].

For all DepthFormer models we follow Nix and Weigend [27] and treat the output of the model as a sample from a Gaussian distribution with the predictive mean $\mu(z)$ and a corresponding predictive variance $s^2(z)$. Based on this, we can minimize the Gaussian Negative Log-Likelihood (GNLL) loss

$$\mathcal{L}_{\text{GNLL}} = \frac{1}{2}\left(\frac{(y-\mu(z))^2}{s^2(z)} + \log(s^2(z))\right)\ , \tag{3}$$

where y is the the ground truth depth.

Through GNLL minimization, DepthFormer inherently learns corresponding variances, which can be interpreted as the aleatoric uncertainty [24,26].

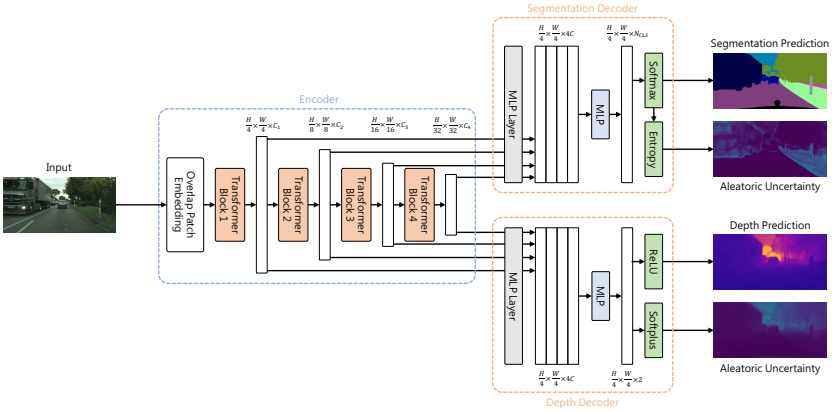


Figure 1: A schematic overview of the SegDepthFormer architecture. It combines the SegFormer [25] architecture with a lightweight all-MLP depth decoder.

SegDepthFormer. To jointly solve semantic segmentation and monocular depth estimation, we propose SegDepthFormer. The architecture, which is shown in Figure 1, combines SegFormer [25] and DepthFormer. It comprises three modules: a hierarchical Transformber-based encoder, an all-MLP segmentation decoder, and an all-MLP depth decoder. Both decoders fuse the multi-level features obtained through the shared encoder to solve the joint prediction task.

SegDepthFormer is trained to minimize the weighted sum of the two previously described objective functions: $\mathcal{L} = \mathcal{L}_{\text{CE}} + w_1 \mathcal{L}_{\text{GNLL}}$, where w_1 is a weighting factor, which we set to $w_1 = 1$ for the sake of simplicity and because both loss values are of similar magnitude.

The respective aleatoric uncertainty is obtained by computing the predictive entropy $H(p(z))$ for the segmentation task or by the predictive variance $s^2(z)$, which is learned implicitly through the optimization of $\mathcal{L}_{\text{GNLL}}$.

3.2 Uncertainty Quantification

We evaluate Monte Carlo Dropout (MCD) [10], Deep Ensembles (DEs) [11], and Deep Sub-Ensembles (DSEs) [12], motivated by their simplicity, ease of implementation, parallelizability, minimal tuning require-

ments, and state-of-the-art performance.

Monte Carlo Dropout. MCD depends on the number and placement of dropout layers and particularly the dropout rate. We adopt the original SegFormer [25] layer placement and consider two dropout rates , 20% and 50%. We sample ten times to obtain the prediction and predictive uncertainty [10,28].

Deep Ensemble. DEs achieve the best results if they are trained to explore diverse modes in function space, which we accomplish by randomly initializing all decoder heads, using random augmentations, and by applying random shuffling of the training data points [11,29]. We report results of a DE with ten members, following the suggestions of previous work [11,29,30].

Deep Sub-Ensemble. Consistent with DEs and MCD, we train the DSE with ten decoder heads for each task on top of a shared encoder [12]. During training, we only optimize a single decoder head per training batch and alternate between them. Thereby, we aim to introduce as much randomness as possible, analogous to the training of DEs. For inference, we utilize all decoder heads.

4 Experimental Setup

Predictions. Regardless of the uncertainty quantification method, we report the results of the mean prediction.

Uncertainty. For the segmentation task, we compute the predictive entropy based on the mean softmax probabilities as a measure for the predictive uncertainty [31]. For the depth estimation task, however, we calculate the predictive uncertainty based on the mean predictive variance and the variance of the depth predictions of the samples [26].

Datasets. We conduct all experiments on Cityscapes [32] and NYUv2 [33].

Data Augmentations. Regardless of the trained model, we apply random scaling with a factor between 0.5 and 2.0, random cropping with a crop size of 768×768 pixels on Cityscapes and 480×640 pixels on NYUv2, and random horizontal flipping with a flip chance of 50%.

Implementation Details. For all training processes, we use AdamW [34] optimizer with a base learning rate of $6 \cdot 10^{-5}$ and employ a polynomial rate scheduler. Besides, we use a batch size of 8 and train for

250 epochs on Cityscapes and for 100 epochs NYUv2, respectively.

Metrics. For semantic segmentation, we report mean Intersection over Union (mIoU) and Expected Calibration Error (ECE) [35]. For monocular depth estimation, we use root mean squared error (RMSE). The uncertainty is evaluated using the following metrics proposed by Mukhoti and Gal [31]:

1. $p(accurate|certain)$: The probability of accurate predictions given low uncertainty.
2. $p(uncertain|inaccurate)$: The probability of high uncertainty given inaccurate predictions.
3. *PAvPU*: The combination of both cases, i.e. accurate|certain and inaccurate|uncertain.

Although these metrics have originally been proposed for semantic segmentation [31], we also use them to evaluate the depth uncertainty. We use the following formula to determine whether a depth prediction is accurate:

$$\max\left(\frac{\mu(z)}{y}, \frac{y}{\mu(z)}\right) = \delta_1 < 1.25 \ , \tag{4}$$

where $\mu(z)$ is the predicted depth value of a pixel and y is the corresponding ground truth depth.

For the sake of simplicity and to simulate real-world employment, we set the uncertainty threshold to the mean uncertainty of a given image for all evaluations.

5 Results

In this section, we describe the results of our joint uncertainty evaluation quantitatively. Tables 1 and 2 contain a detailed comparison, primarily focusing on the uncertainty quality.

Single-task vs. Multi-task. Looking at the differences between the single-task models, SegFormer and DepthFormer, and the multi-task model, SegDepthFormer, the single-task models generally deliver slightly better prediction performance. However, SegDepthFormer exhibits greater uncertainty quality for the semantic segmentation task in comparison to SegFormer. This is particularly evident for

Table 1: Quantitative comparison on the Cityscapes dataset [32] between the three baseline models paired with MCD, DSE, and DEs, respectively. Best results are marked in **bold**.

		Semantic Segmentation					Monocular Depth Estimation				
		mIoU ↑	ECE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑	RMSE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑	Inference Time [ms]
Baseline	SegFormer	0.772	0.033	0.882	0.395	0.797	-	-	-	-	17.90 ± 0.47
	DepthFormer	-	-	-	-	-	7.452	0.749	0.476	0.766	17.59 ± 0.82
	SegDepthFormer	0.738	0.028	0.913	0.592	0.826	7.536	0.745	0.472	0.762	22.04 ± 0.27
MCD (20%)	SegFormer	0.759	**0.007**	0.883	0.424	0.780	-	-	-	-	177.13 ± 0.64
	DepthFormer	-	-	-	-	-	7.956	0.749	0.555	0.739	139.32 ± 0.78
	SegDepthFormer	0.738	0.020	0.911	0.592	0.803	7.370	0.761	0.523	0.757	202.23 ± 0.39
MCD (50%)	SegFormer	0.662	0.028	0.883	0.485	0.760	-	-	-	-	176.98 ± 0.53
	DepthFormer	-	-	-	-	-	21.602	0.181	0.366	0.431	139.81 ± 1.20
	SegDepthFormer	0.640	0.021	0.906	0.616	0.782	8.316	0.733	**0.558**	0.723	203.82 ± 0.81
DSE	SegFormer	0.772	0.037	0.890	0.456	0.797	-	-	-	-	132.30 ± 3.16
	DepthFormer	-	-	-	-	-	**7.036**	0.762	0.467	0.772	91.82 ± 2.01
	SegDepthFormer	0.749	0.009	**0.931**	**0.696**	**0.844**	7.441	0.751	0.463	0.766	212.11 ± 8.44
DE	SegFormer	**0.784**	0.033	0.887	0.416	0.798	-	-	-	-	667.51 ± 2.89
	DepthFormer	-	-	-	-	-	7.222	0.759	0.486	0.771	626.79 ± 2.05
	SegDepthFormer	0.755	0.015	0.917	0.609	0.828	7.156	**0.763**	0.493	**0.773**	743.23 ± 32.95

Table 2: Quantitative comparison on the NYUv2 dataset [33] between the three baseline models paired with MCD, DSE, and DEs, respectively. Best results are marked in **bold**.

		Semantic Segmentation					Monocular Depth Estimation				
		mIoU ↑	ECE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑	RMSE ↓	p(acc/cer) ↑	p(inacc/unc) ↑	PAvPU ↑	Inference Time [ms]
Baseline	SegFormer	0.470	0.159	0.768	0.651	**0.734**	-	-	-	-	18.09 ± 0.41
	DepthFormer	-	-	-	-	-	0.554	0.786	0.449	0.610	17.51 ± 0.87
	SegDepthFormer	0.466	0.151	0.769	0.659	0.733	0.558	0.776	0.446	0.594	22.31 ± 0.23
MCD (20%)	SegFormer	0.422	0.102	0.767	0.706	0.724	-	-	-	-	222.67 ± 0.61
	DepthFormer	-	-	-	-	-	0.605	0.741	0.478	0.568	139.58 ± 052
	SegDepthFormer	0.433	0.093	0.771	0.710	0.725	0.610	0.731	0.450	0.560	251.25 ± 0.81
MCD (50%)	SegFormer	0.273	0.083	0.705	**0.722**	0.713	-	-	-	-	223.25 ± 0.82
	DepthFormer	-	-	-	-	-	0.978	0.516	**0.492**	0.526	139.27 ± 0.69
	SegDepthFormer	0.272	0.084	0.702	0.721	0.711	0.837	0.576	0.473	0.525	251.98 ± 0.60
DSE	SegFormer	0.469	0.092	0.776	0.681	0.726	-	-	-	-	180.42 ± 3.93
	DepthFormer	-	-	-	-	-	0.547	0.782	0.423	0.596	91.66 ± 0.26
	SegDepthFormer	0.461	**0.077**	0.776	0.692	0.723	0.584	0.738	0.403	0.573	261.69 ± 5.10
DE	SegFormer	**0.486**	0.125	0.782	0.675	**0.734**	-	-	-	-	715.97 ± 7.55
	DepthFormer	-	-	-	-	-	**0.524**	**0.808**	0.475	**0.613**	624.30 ± 2.07
	SegDepthFormer	0.481	0.122	**0.783**	0.682	0.733	0.552	0.785	0.453	0.590	788.76 ± 2.00

$p(uncertain|inaccurate)$ on Cityscapes. For the depth estimation task, there is no significant difference in terms of uncertainty quality.

Baseline Models. As expected, the baseline models have the lowest inference times, being 5 to 30 times faster without using any uncertainty quantification method. While their prediction performance turns out to be quite competitive, only beaten by DEs, they show poor calibration and uncertainty quality for semantic segmentation. Surprisingly, the uncertainty quality for the depth estimation task is very

decent, often only surpassed by the DE.

Monte Carlo Dropout. MCD causes a significantly higher inference time compared to the respective baseline model. Additionally, leaving dropout activated during inference to sample from the posterior has a detrimental effect on the prediction performance, particularly with a 50% dropout ratio. Nevertheless, MCD outputs well-calibrated softmax probabilities and uncertainties, although the results should be interpreted with caution because of the deteriorated prediction quality.

Deep Sub-Ensemble. Across both datasets, DSEs show comparable prediction performance compared with the baseline models. Notably, DSEs consistently demonstrate a high uncertainty quality across all metrics, particularly in the segmentation task on Cityscapes.

Deep Ensemble. In accordance with previous work [28], DEs emerge as state-of-the-art, delivering the best prediction performance and mostly superior uncertainty quality. At the same time, DEs suffer from the highest computational cost.

6 Conclusion

By comparing uncertainty quantification methods in joint semantic segmentation and monocular depth estimation, we find Deep Ensembles offer the best performance and uncertainty quality, albeit at higher computational cost. Deep Sub-Ensembles provide an efficient alternative with minimal trade-offs. Additionally, we reveal the potential benefit of multi-task learning with regard to uncertainty quality of the semantic segmentation task compared to solving both tasks separately.

Acknowledgments

The authors acknowledge support by the state of Baden-Württemberg through bwHPC.

References

1. R. McAllister, Y. Gal, A. Kendall, M. van der Wilk, A. Shah, R. Cipolla, and A. Weller, “Concrete Problems for Autonomous Vehicle Safety: Advantages

of Bayesian Deep Learning," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 4745–4753.

2. C. Steger, M. Ulrich, and C. Wiedemann, *Machine Vision Algorithms and Applications*. John Wiley & Sons, 2018.
3. S. Landgraf, M. Hillemann, M. Aberle, V. Jung, and M. Ulrich, "Segmentation of industrial burner flames: A comparative study from traditional image processing to machine and deep learning." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 10, 2023.
4. S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2022.
5. X. Dong, M. A. Garratt, S. G. Anavatti, and H. A. Abbass, "Towards real-time monocular depth estimation for robotics: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16 940–16 961, 2022.
6. C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.
7. J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A Survey of Uncertainty in Deep Neural Networks," *arXiv:2107.03342*, 2022.
8. K. Lee, H. Lee, K. Lee, and J. Shin, "Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples," *arXiv:1711.09325*, 2018.
9. D. J. C. MacKay, "A Practical Bayesian Framework for Backpropagation Networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992.
10. Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48. PMLR, 2016, pp. 1050–1059.
11. B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.
12. M. Valdenegro-Toro, "Sub-ensembles for fast uncertainty estimation in neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4119–4127.

13. S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich, "Efficient multi-task uncertainties for joint semantic segmentation and monocular depth estimation," in *DAGM German Conference on Pattern Recognition*. Springer, 2024.
14. L. He, J. Lu, G. Wang, S. Song, and J. Zhou, "Sosd-net: Joint semantic object segmentation and depth estimation from monocular images," *Neurocomputing*, vol. 440, pp. 251–263, 2021.
15. T. Gao, W. Wei, Z. Cai, Z. Fan, S. Q. Xie, X. Wang, and Q. Yu, "Ci-net: A joint depth estimation and semantic segmentation network using contextual information," *Applied Intelligence*, vol. 52, no. 15, pp. 18 167–18 186, 2022.
16. N. Ji, H. Dong, F. Meng, and L. Pang, "Semantic segmentation and depth estimation based on residual attention mechanism," *Sensors*, vol. 23, no. 17, p. 7466, 2023.
17. D. Brüggemann, M. Kanakis, A. Obukhov, S. Georgoulis, and L. Van Gool, "Exploring relational context for multi-task dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 869–15 878.
18. X. Xu, H. Zhao, V. Vineet, S.-N. Lim, and A. Torralba, "Mtformer: Multi-task learning via transformer and cross-task reasoning," in *European Conference on Computer Vision*. Springer, 2022, pp. 304–321.
19. S. Landgraf, M. Hillemann, K. Wursthorn, and M. Ulrich, "Uncertainty-aware cross-entropy for semantic segmentation," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-2, 2024.
20. K. Wursthorn, M. Hillemann, and M. Ulrich, "Uncertainty quantification with deep ensembles for 6d object pose estimation," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-2, 2024.
21. D. W. Wolf, P. Balaji, A. Braun, and M. Ulrich, "Decoupling of neural network calibration measures," in *DAGM German Conference on Pattern Recognition*. Springer, 2024.
22. Y. Gal, "Uncertainty in deep learning," *Ph.D. thesis, University of Cambridge*, 2016.
23. Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *International conference on machine learning*. PMLR, 2017, pp. 1183–1192.
24. A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 5580–5590.

25. E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

26. A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3153–3160, 2020.

27. D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, vol. 1. IEEE, 1994, pp. 55–60.

28. F. K. Gustafsson, M. Danelljan, and T. B. Schon, "Evaluating scalable bayesian deep learning methods for robust computer vision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 318–319.

29. S. Fort, H. Hu, and B. Lakshminarayanan, "Deep Ensembles: A Loss Landscape Perspective," *arXiv:1912.02757*, 2020.

30. S. Landgraf, K. Wursthorn, M. Hillemann, and M. Ulrich, "Dudes: Deep uncertainty distillation using ensembles for semantic segmentation," *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 92, no. 2, pp. 101–114, 2024.

31. J. Mukhoti and Y. Gal, "Evaluating bayesian deep learning methods for semantic segmentation," *arXiv preprint arXiv:1811.12709*, 2018.

32. M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

33. N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, 2012, Proceedings, Part V 12*. Springer, 2012, pp. 746–760.

34. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

35. M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

Evaluation of 3D-LiDAR based person detection algorithms for edge computing

Dennis Basile[1], Dennis Sprute[1], Helene Dörksen[2], and Holger Flatt[1]

[1] Fraunhofer IOSB, Industrial Automation Branch (IOSB-INA), Campusallee 1, 32657 Lemgo
[2] OWL University of Applied Sciences and Arts, Campusallee 2, 32657 Lemgo

Abstract This paper addresses the need for reliable person detection systems in public spaces by developing a novel dataset tailored for solid-state 3D-LiDAR sensors and evaluating various neural network architectures. The dataset was created using a Blickfeld solid-state 3D-LiDAR, capturing 265 point clouds in a controlled test environment modeled on a three-lane pedestrian crossing. The neural network architectures evaluated include VoxelNeXt, PillarNet, SECOND, PointPillar, CenterPoint, Voxel-R-CNN, PointRCNN, PartA2, and PV-RCNN. The evaluation methodology follows the KITTI benchmark metric for performance analysis. Key results indicate that voxel-based approaches like SECOND and VoxelNeXt achieve inference speeds of 10.3 FPS and 9.8 FPS on an NVIDIA Jetson AGX platform, respectively, with mean Average Precision (mAP) scores of 95% and 90%. In contrast, the hybrid approach PV-RCNN, which combines voxel-based and point-based methods, achieves a mAP of 92% but a slower inference speed of 2.5 FPS. These results underscore the trade-offs between speed and accuracy in person detection using solid-state 3D-LiDAR, highlighting the potential of voxel-based methods for real-time applications. The results contribute to the advancement of person detection technologies in public security and smart city initiatives.

Keywords 3D-LiDAR, person detection, edge computing

1 Introduction

The increasing demand for robust and reliable person detection systems in public spaces has driven advancements in sensor technology and machine learning algorithms. Accurate detection is crucial for applications like public security, traffic management, and smart city initiatives. In the domain of public space surveillance, these systems must accurately localize and classify objects in real-time and operate under challenging conditions such as fog, snow, and rain, while complying with the General Data Protection Regulation (GDPR) in the European Union. Existing systems use various sensors like PIR, laser barriers, radar, and cameras. However, each technology has drawbacks. For example, PIR sensors struggle with detecting groups due to lack of a classical field of view, while cameras, although effective with AI for detection and classification, raise privacy concerns under EU-GDPR [1].

In contrast, solid-state 3D-LiDAR technology shows great potential by generating precise 3D point clouds for privacy-preserving and reliable detection [2]. This makes 3D-LiDAR ideal for applications requiring accuracy, real-time operation, environmental resilience, and data privacy. Currently, 3D-LiDAR is extensively used and researched in autonomous driving systems [3]. However, the available datasets for training neural networks focus on automotive use and may not encompass the broader range of potential applications. They are captured with rotating 3D-LiDAR sensors, whose characteristics, such as resolution, range, and field of view, differ significantly from solid-state 3D-LiDARs. Transferring an existing dataset to the characteristics of a solid-state 3D-LiDAR is challenging. Consequently, there is no sufficient dataset for independent analysis using solid-state 3D-LiDAR sensors. This necessitates the creation of new datasets targeting the specific hardware characteristics of solid-state 3D-LiDAR to achieve optimal performance in people detection with deep learning approaches. Furthermore, there has been no comprehensive comparison of neural network architectures with respect to the specific requirements for person detection in public spaces using solid-state 3D-LiDAR sensors and edge computing. Therefore, this paper contributes by developing a novel dataset for solid-state 3D-LiDAR sensors and performing a thorough comparison of various neural network architectures addressing the requirements for person detection systems in public environments.

The rest of this paper is organized as follows: Section II reviews related work focusing on 3D-LiDAR datasets. Section III describes the generation of the novel dataset based on a design flow and a person classification scheme. In Section IV, the approach is applied within a case study by creating a dataset used to train different CNN architectures. Section V presents the results of the evaluation, and finally, Section VI concludes the paper.

2 Related Work

3D-LiDAR technology has become crucial for advanced driver assistance systems, primarily used for detecting obstacles [4]. Current implementations mainly utilize rotating 3D-LiDARs, as demonstrated by datasets like KITTI, which is a standard benchmark in this field [5]. Several other datasets have been created (refer to Tab. 1), all based on rotating 3D-LiDARs. These datasets are primarily designed for automotive applications, potentially limiting their broader applicability.

Table 1: Overview of various LiDAR datasets.

Dataset	**LiDAR Type**	**LiDAR System**	**Licensing**
KITTI [6]	Rotating	Velodyne HDL-64E	Non-commercial
Waymo Open Dataset [7]	Rotating	In-house development	Non-commercial
nuScenes [8]	Rotating	Velodyne HDL-32E	Non-commercial
PandaSet [9]	Rotating	Hesai Pandar64	Commercial
Argoverse 2 [10]	Rotating	Velodyne VLP-32C	Non-commercial
ONCE [11]	Rotating	40-Beam LiDAR	Non-commercial

Solid-state LiDARs, however, offer several advantages over rotating LiDARs, such as being more compact, lighter, more energy- and cost-efficient. Additionally, without mechanical components, they are maintenance-free and have a longer lifespan [12]. Recent studies suggest that solid-state LiDARs can also be used effectively for tasks beyond obstacle detection, like pedestrian recognition. For example, Peng et al. [13] explored using solid-state LiDAR and cameras for pedestrian detection. However, using camera data raises privacy concerns as it can capture identifiable personal information.

Sprute et al. [14] address the challenge of achieving high-resolution

spatial coverage with solid-state LiDAR without cameras, focusing on detecting people using deep learning techniques. 3D-LiDAR sensors capture point clouds, which are then converted into depth images through clustering techniques. Afterwards, they are processed with a ResNet-based neural network for object classification. This method is computationally intensive, limiting real-time processing on embedded systems. While it improves spatial coverage and detection accuracy, it does not offer direct real-time processing of point clouds, which can be a limitation in scenarios requiring immediate feedback.

Several points from current research highlight the need for further investigation. First, detecting people using solid-state LiDAR and deep learning is feasible, but existing datasets are designed for rotating LiDAR systems, limiting their applicability. A new dataset for solid-state LiDAR is needed.

Second, direct processing of point clouds for person detection is rarely explored. Most studies convert point clouds into depth images before classification, which is computationally demanding and unsuitable for real-time applications.

Third, embedded systems have not been sufficiently considered. Mapping deep learning architectures onto embedded systems could enhance efficiency and applicability, especially for compact, energy-efficient use cases.

This work develops a new dataset for solid-state LiDAR and evaluates deep learning architectures for direct point cloud processing. The aim is to identify effective deep learning models for implementation on embedded systems for efficient person detection.

3 Novel dataset for person detection

Since there are currently no publicly available datasets specifically tailored to the requirements for person detection using solid-state LiDAR, a custom dataset for model training is required.

A solid-state 3D-LiDAR sensor system is employed to capture data, mapping the surroundings as a 3D point cloud. The orientation and position of the sensor remain static throughout the data collection process, ensuring consistent raw data acquisition.

To create a dataset, several processing steps must be carried out. The

raw data has to be stored, followed by storing the raw data in individual frames. These frames then have to be normalized and converted into a point cloud format. Subsequently, the LiDAR coordinate data has to be adjusted to meet the specific requirements for training. After this, the data has to be annotated, and labels have to be created. Finally, the dataset has to be split into training, validation and test subsets.

The sensor setup was established in a specially designed test environment on the premises of the Fraunhofer IOSB-INA Institute, ensuring unobstructed visibility. The setup is based on previous work of Sprute et al. [15]. The LiDAR sensor was installed at a height of four meters with a 16° tilt to ensure optimal coverage of the entire area. The setup was focused on a distance of 9 meters and was directed towards a three-lane pedestrian crossing at an intersection. The data was captured using a solid-state 3D LiDAR sensor from the company Blickfeld [16]. The sensor was configured with a field of view of 72° x 30°, a framerate of 2.4 Hz, and 200 scan lines.

The dataset is collected from the recorded raw data, where different individuals passed by the LiDAR within a range of up to 30 meters. For the training of the deep learning algorithms, a single class 'Person' with different variations was considered. This ensured that the model could detect and analyze various person types and their movement patterns. The manual annotation of the single objects was carried out carefully, as it directly impacts the quality of the detection results after training. The entire dataset consists of 265 different point clouds. An example of the classes annotated in the dataset can be seen in Fig. 1.

To ensure the versatility and robustness of the proposed recognition system, various classes of people based on their relevance in public space have to be provided [15]. The following classes are used to extend the dataset: 1) individuals without physical disabilities, 2) individuals with forearm crutches, 3) individuals with rollators, 4) individuals with mobile phones, 5) groups of people, and 6) individuals with walking sticks.

These annotations reflect common situations in public areas, capturing a wide range of human activities and interactions. Recognizing such diverse scenarios is particularly relevant for surveillance and public safety applications, enhancing the model's ability to detect individuals accurately in various contexts. This variety of annotated classes ensures that the developed model is capable of recognizing and

correctly classifying different situations and groups of people.

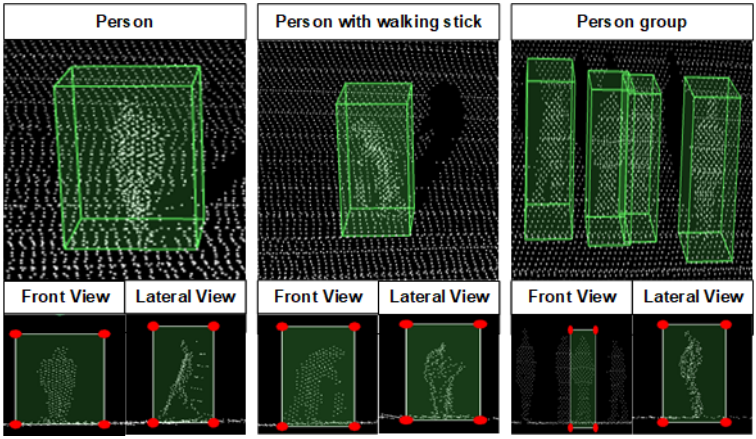


Figure 1: Examples from the custom dataset of manually annotated people.

4 Case Study

4.1 Neural Network

There are various approaches of deep learning architectures for the direct processing of point clouds, which can be categorized into voxel-based, point-based, and hybrid methods.

Voxel-Based Approaches

Voxel-based approaches partition the point cloud into small 3D cubes (voxels) and extract features from each voxel using a Voxel Feature Encoding (VFE) layer. These methods convert the irregular point cloud data into a regular grid, which can then be processed using sparse convolutional neural networks (CNNs) [17]. Advantages of these methods include fast inference times and reduced computational load. However, there are drawbacks, such as information loss due to the choice of voxel size [18]. Examples of voxel-based approaches used in this

study include: (1) SECOND [17], (2) PointPillars [19], (3) PillarNet [20], (4) CenterPoint [21], (5) VoxelNeXt [22], (6) PartA2 [23], (7) Voxel-R-CNN [24].

Point-Based Approaches

Point-Based approaches directly process the point cloud . These method use PointNet++ [25], to learn features directly from the raw points, achieving a higher level of detail. However, they often incur higher computational costs due to the unstructured nature of the data, increased memory usage, and slower inference speeds [26]. Example of a point-based approach used in this study is PointRCNN [27]

Hybrid-Based Approaches

The hybrid method is an extension that combines voxel- and point-based approaches to point cloud processing, combining the strengths of each. Voxel-based methods are faster but can lose information, while point-based methods retain all information but are slower to process. This hybrid approach attempts to combine efficient computation with comprehensive data representation. An example of a hybrid-based approach used in this study is PV-RCNN [28]

4.2 Training

The open-source framework Point Cloud Detection (OpenPCDet) [29] was employed for training and execution of the deep learning architectures described in Section 4.1. The deep learning architectures were trained on a Windows system with the following specifications: 64 GB of DDR4 RAM, an AMD Ryzen 9-3900X 12-core processor, and an RTX2080 graphics card with 8 GB of memory. To ensure the reliability of the results, the dataset was randomly divided into two distinct sets: 70% for training and 30% for validation. To enhance the performance of the trained model, data augmentation techniques were employed to artificially expand the dataset [30]. These techniques included rotation, scaling, and mirroring of the point cloud, as well as the generation of additional bounding boxes and their point data based on the training dataset through the introduction of artificial elements.

The extended Adam algorithm, OneCycleLR [31], was employed for all architectures for the optimization of the neural network's weights mentioned in Section 4.1, wherein a variable learning rate was utilized during training. A maximum learning rate of 10^{-4} was selected, with a momentum of 0.95–0.85. The training process was performed with batch sizes of 6 point clouds over 120 epochs.

5 Results

The calculation to analyze performance is based on the KITTI benchmark procedure [5]. Nine distinct neural network architectures were trained on our novel dataset and subsequently evaluated in terms of their performance, including measures such as average precision (AP) and inference time. In Tab. 2, the results of the conducted investigation of the evaluated deep learning architectures with the custom dataset for AP and the measured inference time on an edge computing device Nvidia Jetson AGX system are presented.

The results demonstrate that voxel-based approaches, such as SECOND, VoxelNeXt, or Voxel-R-CNN, achieve notable performance in both AP and inference time, offering a suitable balance between speed and accuracy when compared to point-based and hybrid approaches. These results are significantly better when compared to the performance of a point-based approach, such as PointRCNN and a hybrid approach, such as PV-RCNN. Some qualitative detection results are shown in Fig. 2.

Table 2: Comparison of architecture performance.

Type	Architecture	AP (IoU = 0.5)	Inference time (FPS)
Voxel	CenterPoint	0.92	7.7
Voxel	Part-A2	0.95	5.0
Voxel	PillarNet	0.91	7.2
Voxel	PointPillar	0.89	9.1
Voxel	SECOND	0.95	**10.3**
Voxel	Voxel-R-CNN	**0.97**	7.2
Voxel	VoxelNeXt	0.90	9.8
Point	PointRCNN	0.92	1.6
Voxel/Point	PV-RCNN	0.90	2.5

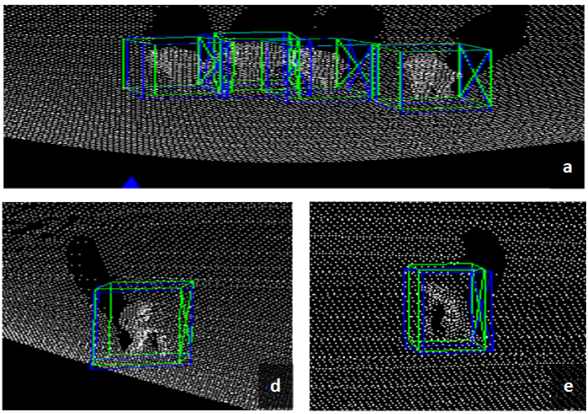


Figure 2: Exemplary person detection based on SECOND architecture. The blue rectangles represent the reference bounding boxes, while the green rectangles indicate the predicted bounding boxes from the neural network.

6 Conclusions and Future Work

The study employing the newly created dataset demonstrates that voxel-based methods, particularly SECOND, achieved the best results, reaching 10.3 FPS with an average precision (AP) of 95%. This indicates that classification and localization using point clouds collected with solid-state 3D-LiDAR sensor are possible with an embedded system like the Nvidia Jetson AGX. The evaluation of nine deep learning algorithms for processing 3D point clouds with a solid-state 3D-LiDAR sensor on an edge computing system revealed that single-stage methods based on voxel preprocessing are most effective. Specifically, SECOND, VoxelNeXt, and PointPillar showed high classification and localization performance with real-time processing capabilities. These results confirm that appropriate voxel-based deep learning architectures exist to implement a person detection system on an edge computing platform with a solid-state 3D-LiDAR sensor, enabling efficient real-time person detection and visualization of 3D point clouds.

Future work will focus on refining the dataset to include more diverse and realistic point cloud scenes, addressing variations in weather conditions and background objects. Class separation and the inclusion

of new classes, such as people with bicycles and strollers, will be investigated to enhance the system's robustness and flexibility. Finally, the approach will be integrated into an embedded smart sensor system, designed for usage in public spaces.

References

1. A.-M. C. Drăgulinescu, I. Marcu, S. Halunga, and O. Fratu, "Persons Counting and Monitoring System Based on Passive Infrared Sensors and Ultrasonic Sensors (PIRUS)," in *Pervasive Computing Paradigms for Mental Health*. Springer International Publishing, 2018, vol. 207, pp. 100–106.
2. N. Li, C. P. Ho, J. Xue, L. W. Lim, G. Chen, Y. H. Fu, and L. Y. T. Lee, "A progress review on solid-state lidar and nanophotonics-based lidar sensors," *Laser & Photonics Reviews*, vol. 16, no. 11, p. 2100511, 2022.
3. K. Li and L. Cao, "A Review of Object Detection Techniques," in *International Conference on Electromechanical Control Technology and Transportation (ICECTT)*. IEEE, May 2020, pp. 385–390.
4. Y. Li and J. Ibanez-Guzman, "Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, Jul. 2020.
5. A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.
6. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
7. P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2443–2451.
8. H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, jun 2020, pp. 11 618–11 628.
9. P. Xiao, Z. Shao, S. Hao, Z. Zhang, X. Chai, J. Jiao, Z. Li, J. Wu, K. Sun, K. Jiang, Y. Wang, and D. Yang, "PandaSet: Advanced Sensor Suite Dataset

for Autonomous Driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Sep. 2021, pp. 3095–3101.

10. B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting," *arXiv*, 2023, version Number: 1.
11. J. Mao, M. Niu, C. Jiang, H. Liang, J. Chen, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, H. Xu, and C. Xu, "One Million Scenes for Autonomous Driving: ONCE Dataset," *arXiv*, 2021, version Number: 3.
12. Y. Li and H. Shi, Eds., *Advanced driver assistance systems and autonomous vehicles: from fundamentals to applications*. Springer, 2022.
13. Z. Peng, Z. Xiong, Y. Zhao, and L. Zhang, "3-d objects detection and tracking using solid-state lidar and rgb camera," *IEEE Sensors Journal*, vol. 23, no. 13, pp. 14 795–14 808, 2023.
14. D. Sprute, T. Westerhold, F. Hufen, H. Flatt, and F. Gellert, "DSGVO-konforme Personendetektion in 3D-LiDAR-Daten mittels Deep Learning Verfahren (in German)," in *Bildverarbeitung in der Automation: Ausgewählte Beiträge des Jahreskolloquiums BVAu 2022*. Springer, 2023, pp. 33–45.
15. D. Sprute, F. Hufen, T. Westerhold, and H. Flatt, "3D-LiDAR-based pedestrian detection for demand-oriented traffic light control," in *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*, 2023, pp. 1–7.
16. Blickfeld GmbH, "Technical manual of the blickfeld lidar cube 1," *https://www.blickfeld.com*, 2021. [Online]. Available: https://www.blickfeld.com/wp-content/uploads/2022/10/Blickfeld-A5-Manual_en_v.4.2.pdf,[Onlineaccess:2024-02-23]
17. Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018.
18. M. Ye, S. Xu, and T. Cao, "Hvnet: Hybrid voxel network for lidar based 3d object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1628–1637.
19. A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 689–12 697.
20. G. Shi, R. Li, and C. Ma, "Pillarnet: Real-time and high-performance pillar-based 3d object detection," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Springer Nature Switzerland, 2022, pp. 35–52.

21. T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2021, pp. 11 779–11 788.
22. Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "VoxelNeXt: Fully sparse voxelnet for 3D object detection and tracking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 21 674–21 683.
23. S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 08, pp. 2647–2664, 2021.
24. J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel R-CNN: Towards high performance voxel-based 3D object detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1201–1209, 05 2021.
25. C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.
26. M. Drobnitzky, J. Friederich, B. Egger, and P. Zschech, "Survey and systematization of 3D object detection models and methods," *The Visual Computer*, vol. 40, no. 3, pp. 1867–1913, 2024.
27. S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2019, pp. 770–779.
28. S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 526–10 535.
29. O. D. Team, "Openpcdet: An open-source toolbox for 3d object detection from point clouds," https://github.com/open-mmlab/OpenPCDet, 2020.
30. C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, Dec. 2019.
31. L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," SPIE, pp. 369–386, 2019.

Explainable fatigue detection in assembly tasks through graph neural networks

Vishwesh Vishwesh[1], Maximillian Becker[2*], Pascal Birnstill[1], and Jürgen Beyerer[1,2]

[1] Fraunhofer IOSB, Karlsruhe, Germany
[2] Vision and Fusion Laboratory, Karlsruhe Institute of Technology, Karlsruhe, Germany
*maximilian.becker@kit.edu

Abstract Fatigue during assembly tasks can have a negative effect on subjective as well as objective quality of work. We recorded a novel dataset for the purpose of detecting fatigue in assembly scenarios. Participants were instructed to assemble and disassemble model cars with the help of a robot arm. The recordings consist of video, depth video, EEG and eye tracking data as well as questionnaires on the participants' fatigue. The dataset can be provided to researchers on demand. In addition to recording a dataset, we implemented a proof of concept system to detect fatigue solely on image data. In our approach the eye tracking data was used to label the participants' fatigue. Afterwards, a graph neural network was trained on poses extracted from the video data and the generated labels. The classifications of the model are made transparent through the use of explainable AI using saliency maps and GradCAM. This work can have a positive impact on human-machine interaction and assistance systems. Through explainability, we aim to increase the acceptance of such systems by workers and industries.

Keywords GNN, XAI, GradCAM, saliency maps, fatigue detection, dataset

1 Introduction

As the pace of industrial work intensifies, understanding and mitigating the effects of fatigue on human performance has emerged as a

challenge. Fatigue can significantly impair cognitive and physical capabilities, leading to reduced productivity, increased error rates and potentially hazardous working conditions. Therefore, detecting fatigue is essential for enhancing productivity and improving the safety and comfort of workers. The goal of this work is to develop a system that can accurately detect fatigue levels in workers during assembly tasks and provide simple explanations for the decisions made by the system. By doing so, we aim to contribute to the development of more adaptive and worker-friendly industrial environments that are optimized for both efficiency and safety.

To explore this issue, we recorded a dataset where participants performed assembly tasks in a controlled environment. We used two modalities of the dataset: eye tracking and video data. The eye tracking data is employed to generate fatigue labels for each timestamp with pupil diameter variability (PDV) as the indicator, which has been empirically validated as a reliable marker of overall fatigue [1]. For the fatigue detection we only use video data. Video data does not disrupt the worker as opposed to wearable sensors and is often already available as it is needed for many assistance systems. On the video data, pose estimation is performed using Mediapipe [2], a tool that extracts human poses from video frames. The resulting pose data is then used to train a Graph Convolution Network (GCN), which is designed to predict fatigue levels based on body posture.

Incorporating transparency into the decision-making process of AI systems is critical, particularly in industrial contexts where the acceptance and trust in assistance systems are paramount. Furthermore, the European AI Act demands transparency if AI systems are used in ”work-related relationships [...] to allocate tasks” and ”monitor and evaluate the performance and behaviour of persons” (Annex III, 4 b) [3]. To address this, our system integrates Explainable artificial intelligence (XAI) techniques to provide local explanations for its decisions.

The primary contribution of this research is the development of a fatigue detection system that integrates deep learning methodologies with XAI techniques while operating only on camera data. This work has the potential to enhance the quality of work environments by fostering transparency and trust in AI-driven assistance systems and Industry 4.0.

2 Related work

Fatigue detection has become an area of increasing interest due to its wide-ranging applications, from workplace safety to medical diagnostics. Various techniques have been employed to capture and assess fatigue levels, each offering unique advantages depending on the domain and context of usage. In this section, we explore different approaches to fatigue detection, from conventional methods like eye tracking to more recent advancements involving pose detection and XAI.

One of the widely used methods for fatigue detection is eye tracking, particularly in domains like automotive safety and air traffic control. By measuring parameters such as blink rate, saccadic movement, and gaze patterns, researchers have been able to infer levels of cognitive and physical fatigue. Benedetto et al. [4] demonstrated the correlation between eye blink frequency and driver fatigue in simulated driving environments. Di Stasi et al. [5] leveraged saccadic velocity to evaluate cognitive load and fatigue. Lengenfelder et al. [6] observed mental fatigue from eye tracking while performing interactive image exploitation. Sirois et al. [7] showed that pupil dilation responds to task difficulty and cognitive effort, reinforcing the role of pupil diameter variability (PDV) in fatigue detection. However, these methods, while effective, are often constrained by environmental factors and require specialized, obtrusive hardware, limiting their adaptation and applications.

Opposed to eye trackers, which are highly specialized, nearly any camera can be used for facial recognition and pose detection. Facial recognition techniques leverage the subtle changes in facial expressions and muscle movements that occur as fatigue sets in. For instance, Haque et al. [8] argued that features like drooping eyelids, yawning frequency, and overall facial muscle relaxation can serve as strong indicators of fatigue. In driver monitoring systems, facial recognition has been applied to track drowsiness and fatigue by detecting changes in eye closure duration, blink frequency, and facial muscle slackness, as demonstrated in studies by Bergasa et al. [9], Ji et al. [10] and García et al. [11]. Similarly, Sikander et al. [12] and Liu et al. [13] explored the use of facial landmarks in real-time monitoring systems to detect early signs of cognitive and physical fatigue in drivers.

Pose detection has traditionally been used in fields such as sports

science [14] and rehabilitation [15], but its recent application in health monitoring has gained traction. The rise of pose estimation libraries like OpenPose [16] and Mediapipe [2] have made this approach more accessible, enabling the detection of joint coordinates in real-time using just standard cameras. Hawley et al. [17] demonstrated using machine learning that postural sway and joint angle deviations could be used as indicators of physical fatigue in lifting tasks. Similarly, Wang et al. [18] used pose estimation in athletic assistance system by incorporating deep learning methods. Strain and fatigue are detected for risk analysis by Papoutsakis et al. [19] in an industrial environment using pose estimation. This paper aims to provide a feedback system to industry workers on safe and unsafe poses while working. Pose-based methods offer the advantage of being non-invasive and relatively inexpensive making them attractive for broader deployment [20]. There exist many more techniques for fatigue detection like EEG, body-borne sensors of physiological markers. They do however require specialized, wearable hardware for every worker.

XAI's role in fatigue detection is particularly crucial because of the need for trust and validation in AI-driven decisions. Rivera et al. [21] detect mental fatigue using EEG data and deploy XAI techniques to interpret the results. They argue that applying deep learning techniques to detect fatigue levels is of limited use and a thorough XAI technique needs to be implemented. Hussain et al. [22] demonstrated how XAI could be used in cognitive fatigue detection using EEG to highlight the importance of specific brainwave patterns, allowing healthcare professionals to validate the AI's interpretation of EEG signals. The potential for XAI in fatigue detection systems is growing, but research is still in its infancy, with most efforts focused on improving prediction accuracy rather than interpretability. As fatigue detection systems are increasingly integrated into workplaces and healthcare, ensuring that their decisions are explainable will become essential for achieving broader acceptance and fulfilling regulatory requirements.

3 Dataset

We started our work by recording a novel dataset for fatigue detection in assembly tasks. The dataset is multimodal, containing EEG, eye

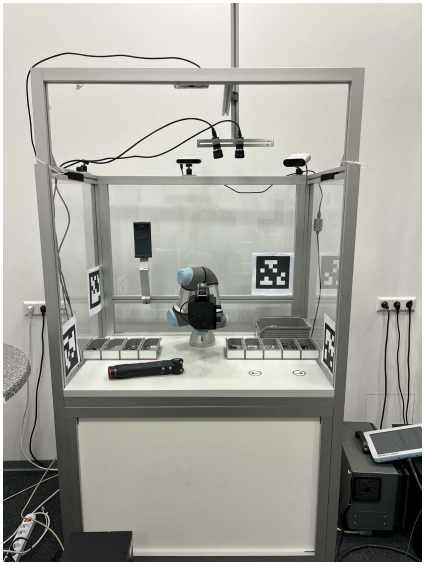


Figure 1: Setup of the assembly table during recording.

tracking, video and depth video data. Additionally, we gathered data from questionnaires that participants had to fill out before and after the experiment. These include the NASA-TLX [23] after the experiment and participants' fatigue on the rating of fatigue (ROF) scale [24] before and after the experiment. The ROF scale is a scale from 0-10 with 0 indicating no fatigue at all and 10 indicating total fatigue and exhaustion.

We invited 30 participants to the recordings. 5 of them participated 3 times each resulting in 40 total recordings. During the experiment, the participants wore an EEG-headset, eye tracking glasses and were recorded with a regular RGB and a stereoscopic depth camera. Pupil Labs Core[1] was used as the eye tracker. To simulate an assembly task participants were asked to first assemble and then disassemble 3 model cars form 3D-printed components. A monitor showed step by step

[1] `https://pupil-labs.com/products/core`

instructions which the participants could control via buttons. Help was also provided by a robot arm that held the partly assembled model cars in place. The setup of the assembly table can be seen in Figure 1.

The dataset will not be publicly available but can be provided to researchers on demand.

4 Explainable Fatigue Detection

Our approach to fatigue detection, once trained, relies only on camera data to predict fatigue levels. It builds on the existing research in fatigue detection and pose estimation, but it introduces a novel combination of these fields using XAI. We begin by extracting key labels from the eye tracking data. While several features are available from eye tracking systems, PDV has been selected as our feature of choice due to its established correlation with cognitive load and fatigue. PDV offers an intuitive measure of how the eye's pupil reacts to changes in focus and brightness, which is often a strong indicator of mental fatigue.

The PDV is calculated using standard algorithms that compute the pupil diameter based on frames obtained from the eye tracker. These frames are timestamped, and the change in pupil size over time is measured to yield the PDV. When labeling the data for fatigue detection, a rolling window approach was utilized, assigning fatigue scores based on a 0-5 scale to reflect varying fatigue intensities.

For extracting human poses, RGBD data from our dataset was used. We used Mediapipe, a state-of-the-art library for pose estimation, which provides 33 3D skeletal keypoints of the participants. Each joint comes with X, Y, Z coordinates (representing spatial location) and a visibility score (indicating how clearly the joint was visible in the frame). As the participants were recorded from the front while standing at an assembly table their legs were not visible. Therefore, we removed joints below hips during preprocessing to avoid noisy data.

Once the pose data was extracted, we aligned them with the previously calculated labels. This allowed us to use supervised learning using Graph Convolutional Networks (GCNs). The GCN consisted of three convolutional layers. It was tasked with predicting the fatigue level of a participant based on their pose. To improve the model's ro-

bustness, we experimented with several preprocessing steps, such as balancing the dataset using SMOTEENN, which addressed the issue of class imbalance by combining oversampling of minority classes and under-sampling of majority classes. This technique has proven useful in ensuring that the model does not overfit to the dominant classes while maintaining sufficient samples for the minority classes [25].

To make the model explainable we applied two XAI techniques: saliency maps and Grad-CAM. These methods provided insights into which keypoints (joints) and indirectly which skeletal connections were most influential in determining fatigue levels. These XAI techniques were instrumental in validating that the model was focusing on anatomically relevant areas, aligning with known indicators of physical fatigue [26] [27].

We developed a comprehensive system for detecting fatigue based on video data. By integrating pose estimation, graph-based learning models and leveraging XAI techniques, our method enables better interpretability, which is crucial for identifying key factors contributing to fatigue prediction. Our model lays a strong foundation for future improvements that could enhance its practical applicability with further optimization.

5 Results and discussion

In this section, we will provide a detailed analysis of the outcomes from our experiments, starting with model performance improvements, followed by XAI applications to interpret model predictions using saliency maps and Grad-CAM. Finally, we will demonstrate our XAI techniques in skeletal visualizations with heatmaps, showing how various nodes and skeletal joints contribute to the fatigue prediction.

We started with a baseline model, consisting of three convolution layers, which was trained using a stationary window of 30ms, relying solely on x, y, and z coordinates as features. This model achieved a 54% testing accuracy. To improve performance, we introduced a rolling window of 1.5 seconds, added visibility as a fourth feature, incorporated a dropout layer for regularization and a learning rate scheduler for dynamic optimization. This raised the testing accuracy to 67%.

The next steps involved increasing the convolution layers to five,

which further pushed the accuracy to 70%. Early stopping is added to monitor and prevent overfitting and ensure better generalization to test data. Further, we introduced a preprocessing step on the skeleton by removing joints below the hips as they often were often hidden by the assembly table which reduced the skeleton from 33 to 24 joints. When the model was trained on this data, a testing accuracy of 77% was achieved.

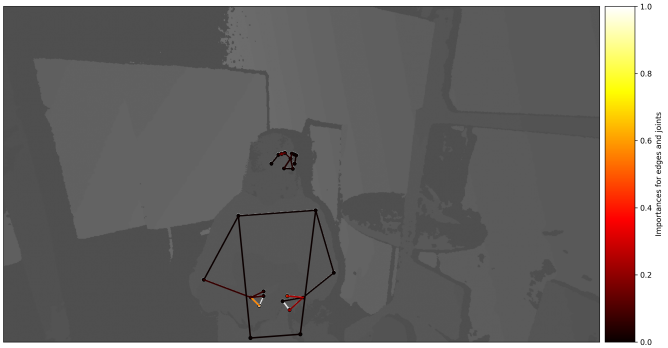


Figure 2: Heatmap of correctly predicted label 0 on depth image.

Figure 3: Heatmap of correctly predicted label 5 on depth image.

We initially trained with ten labels (1-10) but reduced them to 6 (0-5). By reducing the number of fatigue labels from 10 to 6, the model achieved an accuracy improvement from 77% to 80%. By reducing the number of classes, the impact of label noise is reduced, particularly in cases where subjective assessments of fatigue might be inconsistent between adjacent levels. Additionally, simpler categorizations can be more easily understood by non-technical users, leading the user to make better decisions that are more aligned with practical application [28].

As mentioned in the previous chapter we incorporated XAI techniques in the form of saliency maps and Grad-CAM to make our approach more transparent. The saliency maps highlighted the importance of joints such as the shoulders and elbows, which tend to show signs of fatigue during manual tasks. Grad-CAM, on the other hand, visualized the influence of broader skeletal regions, showing how postural deviations in the upper body contributed to the model's predictions. The combined saliency and Grad-CAM visualizations offer a detailed insight into how different parts of the body contribute to fatigue prediction. Figure 2, representing a correctly predicted label of 0 (low fatigue), shows a higher importance around the hands, particularly in the wrist and elbow regions, showing that these regions are indicative of low fatigue levels. Figure 3, which was correctly classified as a 5 (high fatigue), shows a broader spread of important regions, with higher intensity around both the upper body and shoulders, suggesting some reliance on the upper limbs as fatigue increases. However the most important regions are still the hands. In future, we plan to enhance the fatigue prediction model by integrating additional features, such as temporal data from video streams. We also aim to explore advanced explainability techniques to gain deeper insights into the factors influencing fatigue levels.

6 Conclusion

Recognising fatigue in assembly environments is an issue of work safety. We presented an explainable fatigue detection system that works only on image data. Workers do not have to wear any additional devices or sensors hindering them in their work. The video data

for our system can stem from cameras that are often already present for assistance systems. Additionally, we incorporated explainability into our system through the use of saliency maps and Grad-CAM. This makes our system more transparent and helps to comply with the European AI Act which demands transparency when monitoring people in work environments. We would like to build on our existing system and develop it into a real-time assistance system.

Acknowledgment

This work was supported by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.

References

1. M. A. Boksem and M. Tops, "Mental fatigue: costs and benefits," *Brain research reviews*, vol. 59, no. 1, pp. 125–139, 2008.
2. C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.
3. E. Commission, "Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (ec) no 300/2008, (eu) no 167/2013, (eu) no 168/2013, (eu) 2018/858, (eu) 2018/1139 and (eu) 2019/2144 and directives 2014/90/eu, (eu) 2016/797 and (eu) 2020/1828 (artificial intelligence act)." [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1689
4. S. Benedetto, M. Pedrotti, L. Minin, T. Baccino, A. Re, and R. Montanari, "Driver workload and eye blink duration," *Transportation research part F: traffic psychology and behaviour*, vol. 14, no. 3, pp. 199–208, 2011.
5. L. L. Di Stasi, M. B. McCamy, S. L. Macknik, J. A. Mankin, N. Hooft, A. Catena, and S. Martinez-Conde, "Saccadic eye movement metrics reflect surgical residents' fatigue," *Annals of surgery*, vol. 259, no. 4, pp. 824–829, 2014.

6. C. Lengenfelder, J. Hild, M. Voit, and E. Peinsipp-Byma, "Pilot study on gaze-based mental fatigue detection during interactive image exploitation," in *International Conference on Human-Computer Interaction*. Springer, 2023, pp. 109–119.
7. S. Sirois and J. Brisson, "Pupillometry," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 5, no. 6, pp. 679–692, 2014.
8. M. A. Haque, R. Irani, K. Nasrollahi, and T. B. Moeslund, "Facial video-based detection of physical fatigue for maximal muscle activity," *IET Computer Vision*, vol. 10, no. 4, pp. 323–330, 2016.
9. L. M. Bergasa and J. Nuevo, "Real-time system for monitoring driver vigilance," in *Proceedings of the IEEE International Symposium on Industrial Electronics, 2005. ISIE 2005.*, vol. 3. IEEE, 2005, pp. 1303–1308.
10. Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *IEEE transactions on vehicular technology*, vol. 53, no. 4, pp. 1052–1068, 2004.
11. H. García, A. Salazar, D. Alvarez, and Á. Orozco, "Driving fatigue detection using active shape models," in *Advances in Visual Computing: 6th International Symposium, ISVC 2010, Las Vegas, NV, USA, November 29-December 1, 2010, Proceedings, Part III 6*. Springer, 2010, pp. 171–180.
12. G. Sikander and S. Anwar, "Driver fatigue detection systems: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339–2352, 2018.
13. F. Liu, D. Chen, J. Zhou, and F. Xu, "A review of driver fatigue detection and its advances on the use of rgb-d camera and deep learning," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105399, 2022.
14. Y. Yang, Y. Zeng, L. Yang, Y. Lu, X. Lee, and Y. Enomoto, "Action recognition and sports evaluation of running pose based on pose estimation," *International Journal of Human Movement and Sports Sciences*, vol. 12, pp. 148–163, 2024.
15. P. Picerno, A. Cereatti, and A. Cappozzo, "Joint kinematics estimate using wearable inertial and magnetic sensing modules," *Gait & posture*, vol. 28, no. 4, pp. 588–595, 2008.
16. Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.
17. S. J. Hawley, A. Hamilton-Wright, and S. L. Fischer, "Detecting subject-specific fatigue-related changes in lifting kinematics using a machine learning approach," *Ergonomics*, vol. 66, no. 1, pp. 113–124, 2023.

18. J. Wang, K. Qiu, H. Peng, J. Fu, and J. Zhu, "Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance," in *Proceedings of the 27th ACM international conference on multimedia*, 2019, pp. 374–382.

19. K. Papoutsakis, G. Papadopoulos, M. Maniadakis, T. Papadopoulos, M. Lourakis, M. Pateraki, and I. Varlamis, "Detection of physical strain and fatigue in industrial environments using visual and non-visual low-cost sensors," *Technologies*, vol. 10, no. 2, p. 42, 2022.

20. J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2640–2649.

21. M. M. Rivera, L. Martinez, A. Ochoa, A. N. Zezzatti, J. Rodarte, and N. Lopez, "Prototype interface for detecting mental fatigue with eeg and xai frameworks in industry 4.0," *Explainable Artificial Intelligence in Medical Decision Support Systems*, vol. 117, 2023.

22. I. Hussain, R. Jany, R. Boyer, A. Azad, S. A. Alyami, S. J. Park, M. M. Hasan, and M. A. Hossain, "An explainable eeg-based human activity recognition model using machine-learning approach and lime," *Sensors*, vol. 23, no. 17, p. 7452, 2023.

23. S. G. Hart, "Nasa task load index (tlx)," 1986.

24. D. Micklewright, A. St Clair Gibson, V. Gladwell, and A. Al Salman, "Development and validity of the rating-of-fatigue scale," *Sports Medicine*, vol. 47, pp. 2375–2393, 2017.

25. G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

26. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

27. K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Workshop at International Conference on Learning Representations*, 2014.

28. F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

Visual car brand classification by implementing a synthetic image dataset creation pipeline

Jan Lippemeier[1], Stefanie Hittmeyer[2], Oliver Niehörster[3], and Markus Lange-Hegermann[4]

[1] TH OWL University of Applied Sciences and Arts,
Campusallee 12, 32657 Lemgo
[2] Fraunhofer IOSB-INA,
Campusallee 1, 32657 Lemgo
[3] iplus1 GmbH,
Vogelsang 12, 33104 Paderborn
[4] Institute for Industrial Information Technology (inIT),
Campusallee 6, 32657 Lemgo

Abstract Recent advancements in machine learning, particularly in deep learning and object detection, have significantly improved performance in various tasks, including image classification and synthesis. However, challenges persist, particularly in acquiring labeled data that accurately represents specific use cases. In this work, we propose an automatic pipeline for generating synthetic image datasets using Stable Diffusion, an image synthesis model capable of producing highly realistic images. We leverage YOLOv8 for automatic bounding box detection and quality assessment of synthesized images. Our contributions include demonstrating the feasibility of training image classifiers solely on synthetic data, automating the image generation pipeline, and describing the computational requirements for our approach. We evaluate the usability of different modes of Stable Diffusion and achieve a classification accuracy of 75%.

Keywords Image synthesis, image classification, computer vision car brand classification, traffic monitoring, synthetic training data

1 Introduction

In the last twelve years advancements in machine learning, deep learning and object detection achieved remarkable results in performance. The deep learning revolution in image classification started with the publication of AlexNet [1] in 2012. Further performance enhancements have been achieved in the following years with models such as ResNet [2]. Object detection has achieved significant success with models like YOLO (You Only Look Once) [3], which have produced high-accuracy results.

Transfer learning which uses pre-trained existing models is a common approach for solving image classification tasks [4]. Although this approach needs less data than training a model from ground up, it still requires large amounts of labeled data. Existing publicly available datasets can only be successfully used for training if they actually resemble the use case. Even if large amounts of data from the actual use case can be acquired, labeling this data remains time consuming and therefore expensive, as this is often a manual task. Biases within the data present a challenge as there is a compromise to be made, either in the form of potentially keeping the bias, reducing the dataset size to balance classes or oversampling underrepresented classes. In most cases this leads to small available datasets and consequently overfitting. Another common challenge is a low variance within the available data.

Solving computer vision tasks when only limited data is available, is a common major challenge in practice. Limited datasets often lead to overfitted or poorly performing models, endangering the success of a project. We face this challenge by illustrating an adaptable approach. For this approach we synthesize images on demand that are tailored to the respective use case. This leads us to posing our abstract main research question: Is it possible to use image synthesis in an automated manner to create suitable datasets for computer vision tasks with otherwise limited existing data? We evaluate this general approach on a specific real-world application.

Our real-world image classification task is to visually predict the brand of a car as an unequivocal visually determinable feature (see Figure 1). We recorded and labeled data that represents the German automotive traffic; this data was recorded by traffic cameras in Lemgo

Figure 1: The selected brands we aim to classify. These eight brands occur the most in our recorded footage.

- a medium sized town in Germany. However, we are limited by the traffic volume and the capacities for human labeling. Even if unlimited gathering of real labeled data from the German traffic was possible, the data would still include the biases of the real world. Filtering these out and labeling the images would still remain a time consuming task. Existing related datasets such as the Stanford Car Dataset [5] tend to resemble the North American market. Some car brands common in Germany such as Skoda are normally not even present within existing datasets. With limited data from the actual application and no usable existing dataset this problem is a prime example for our main research question.

With the emergence of image synthesis models that create highly realistic images with correct proportions and details we propose the usage of synthetic images as training data for image classification tasks. In theory image synthesis models are a prompt-guided way to synthesize an image of a desired object. Stable Diffusion is an open source model that allows for programmatic image synthesis [6]. The creation of images therefore becomes a question of time and computing power.

We create an automatic pipeline for image dataset creation using Stable Diffusion as a tool to synthesize images. We are able to control the distribution of the generated images by controlling the distribution of the used prompts. By using YOLO we automatically determine the bounding boxes of a car inside a generated image. YOLO also allows us to estimate whether the synthetic image is suitable by giving a confidence score, the bounding box and the class of the detected object.

Although we can avoid class distribution bias by balancing the number of generated images per manufacturer, image synthesis models may still introduce inherent biases. If there are biases within the training data of the image synthesis model it might pass this bias on to the

generated images. In regards to cars the data used in the training of Stable Diffusion could be unbalanced, for example, by favoring new over old, famous over unfamous, and popular over unpopular cars. Further it is not automatically confirmable whether a synthetic image actually matches the desired output encoded by the prompt. Also the perspective, illumination, contrast and other photographic properties might differ from the actual task.

Our main contributions are: We automate an image generation pipeline that also includes labels, bounding boxes and a quality assessment for the synthetic images. We show that training on purely synthetic data from our image generation pipeline is sufficient to train an image classifier that can visually predict the car brand on a real photograph. We describe the required amount of and necessary computation time for synthetic images in our use case. We include and compare different modes of Stable Diffusion for synthesizing images in our pipeline.

2 Related Work

Large-scale text-to-image diffusion models can be fine-tuned to augment the ImageNet training set [7] leading to significant improvements in ImageNet classification accuracy [8]. Moreover, the authors of [9] investigate using synthetic images produced with Stable Diffusion [6] when training models for ImageNet classification. Whether and how synthetic images generated from text-to-image generation models can be used for image classification in data-scarce settings and in large-scale model pre-training for transfer learning is considered in [10] using the GLIDE diffusion model [11].

Synthetic data has been successfully used to improve identification and classification tasks in other applications such as lung edema identification in chest X-ray images [12] and the diagnosis of skin diseases [13]. In the latter two references Stable Diffusion was used to generate the corresponding synthetic image datasets. Introducing synthetic test data has been proposed as a means to improve model evaluation on diverse and underrepresented population subgroups [14].

In the field of vehicle type classification, or, more specifically, car brand and model identification, mainly models trained on real-world

data have been investigated so far. Examples are the extension of models trained on limited-size datasets to handle extreme lighting conditions [15], balanced sampling to address the challenge of classifying imbalanced data from visual traffic surveillance sensors [16], improving accuracy of car type classification through the adaptation of specific CNN architecture models [17], as well as adapting deep learning techniques for vehicle color classification [18] and vehicle logo recognition [19]. The detection, recognition, and counting of vehicles based on their car types using a combination of YOLOv5 and ResNet has been investigated in [20].

3 Method

The goal of this work is to develop a pipeline (illustrated in Figure 2), which can generate a balanced dataset for a computer vision task with otherwise limited available labeled data.

I Dataset Input For this work we used the official car registrations [21] from the Federal Motor Transport Authority of Germany (Kraftfahrt-Bundesamt). They provide data for registered car models in Germany. The features are the vehicle class, the brand, the model name, the build years and the registered number of cars in each category. An example for a car model is the Skoda Karoq, a SUV of the brand Skoda produced in the years 2017, 2018 and 2020. However we ruled out production years earlier than 1990 as they rarely occur in everyday traffic.

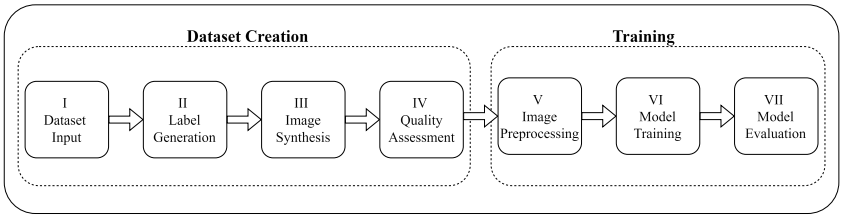


Figure 2: The scheme of the developed pipeline consisting of dataset creation and training. The illustrated pipeline produces a dataset of synthetic images with corresponding labels and bounding boxes. The dataset can be used to automatically train an image classification model.

While it might be possible to generate suitable data even for rare brands we limit our approach to the brands shown in Figure 1. By focusing on these brands we aim to include the often occurring brands such as Volkswagen and Ford but also more rarely occurring brands such as Skoda and Renault.

II Label Generation In order to balance the dataset we use a multi-step hierarchical uniform probability distribution. In the first hierarchy step each brand has the same probability. For each brand the probability of the respective car models is also uniformly distributed. The same principle applies to the construction years for each model as well as the most common colors.

III Image Synthesis In this pipeline step we sample from the labels created in the previous step and create a prompt for each sample as shown in Figure 3. We create two datasets, one for Text-to-Image and one for Image-to-Image using Stable Diffusion XL Turbo. If not otherwise noted we use the standard parameters set by the Python Diffusers library [22]. For Text-to-Image we use four inference steps with one image per prompt and a guidance scale of zero. This guidance scale is recommended in the documentation for the usage of this model [23]. For Image-to-Image we use ten inference step with a guidance scale of 0.4 and a strength of 0.6. These parameters are manually tuned to subjectively fit the desired output.

When using Stable Diffusion in Image-to-Image mode we also have to provide a base image in conjunction with a prompt as the input for the model. To create these base images we use real photographs of cars at different positions on the road cropped to the car with padding. With these base images we intend to implicitly give the desired perspective so that the generated images strongly resemble the real images. The input base image for this mode is scaled up to 720x720 pixels

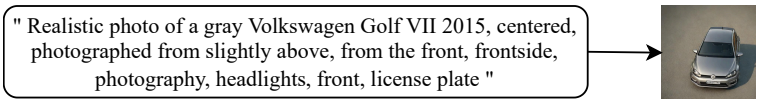


Figure 3: The prompt used to generate the images with Stable Diffusion alongside a generated image using Text-to-Image. The substring *"gray Volkswagen Golf VII 2015"* is changed accordingly for different car models.

Figure 4: Illustration of differences between real images and modes of image generation. Text-to-Image tends to encompass more perspectives contrary to the narrow range of perspectives with Image-to-Image.

as this is the minimal size for Stable Diffusion XL. Figure 4 illustrates real photographs compared to images generated by Text-to-Image and Image-to-Image.

IV Quality Assessment The output of image synthesis models such as Stable Diffusion normally matches the expected output. In most of the cases there is exactly one car as the main subject of the image. The location and the size of the image's subject differs. Therefore an object detection model such as YOLOv8x has to be used to automatically determine the bounding boxes (see Figure 5). However, we observe that in rare cases Stable Diffusion produces something other than the desired output of a singular car (see Figure 5). Therefore, we use YOLOv8x in object detection mode on each image. This allows us to automatically confirm that there is exactly one car in the image. It further assesses the quality of the generated image as YOLO provides a score for the certainty of a detected bounding box. We then crop the image to the bounding boxes of the detected car.

V Image Preprocessing Stable Diffusion models allow to specify the dimensions of the resulting image. However, the subject of the images vary in size so that the images cropped to their subjects' bounding boxes differ in aspect ratios and sizes. We transformed the images to

Figure 5: Bounding Boxes detected with YOLOv8x. This allows to crop the image and provides a confidence score for the presence of a car. It also allows to automatically sort out the two undesired images on the right where more than one car is detected.

64x64 pixels. The small resolution is chosen as the traffic cameras in our use case record in HD and cars at different positions in the image may therefore have a similar resolution. Random Rotation as a classical data augmentation method is also applied to the dataset.

VI Model Training We then use the generated and pre-processed image dataset to train image classifiers. In using the model Resnet-18 that is pre-trained on ImageNet [7], we apply the principle of transfer learning [4]. For adapting this model we replace the last fully connected layer by a new fully connected layer with the same input size and an output size of eight which encodes the classes we want to classify. We do not lock any pretrained layer.

VII Evaluation The performance of the model is validated against real-world data recorded by traffic cameras mounted in Lemgo, Germany. These images were manually labeled. Images of the same car at different positions can exist in the datasets as the cars are moving forward. The split between validation and test dataset is therefore performed based on location of the respective camera and the time of recording. Due to the biases in the real world and the described split the classes are unevenly distributed.

4 Experiments and Results

The time for generating images with Stable Diffusion depends on output size and the number of inference steps. In the following we consider the performance for the parameters described in Section 3 (Method) when run on a Nvidia RTX 3060. When using Text-to-Image (four inference steps) it took 0.85 seconds, for Image-to-Image (ten inference steps) 2.33 seconds. These durations account for image synthesis, bounding box detection, automatic quality assessment and storing of the image.

We retrain Resnet-18 on varying datasets. The model is trained on the images in random order with an exponentially decaying learning rate starting at 0.01 and the stochastic gradient descent optimizer. An epoch for training the Resnet-18 on 100.000 images takes approximately 18 seconds on a Nvidia RTX 3060.

We evaluate the performance of this model regarding the different modes of image synthesis and the required amount of data. As we

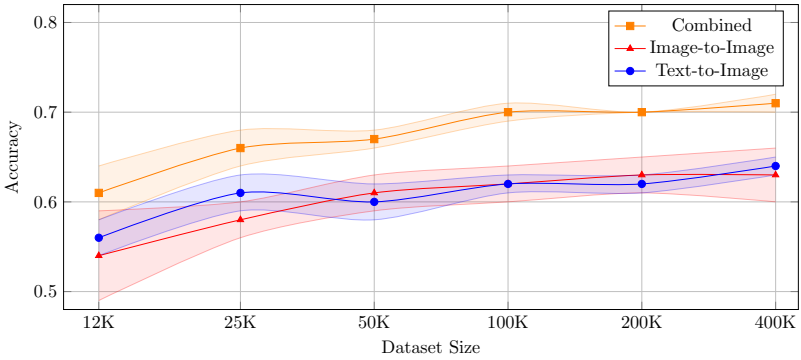


Figure 6: Training Results for Resnet-18 trained on the different dataset sizes. Model performance and dataset size correlate. The experiments were performed five times per dataset size and Stable Diffusion mode. The graph shows a confidence interval of one standard deviation.

use real world photographs of cars in traffic in Lemgo, Germany, we have an unbalanced dataset. To illustrate the unbalanced distribution: VW occurs the most with 976 images in contrast to Renault with only 165 images. The split, described earlier, results in a validation dataset consisting of 1503 images and a test dataset consisting of 1317 images.

Figure 6 shows the training results for different dataset sizes and datasets generated by Image-to-Image mode, Text-to-Image mode and a combination of these modes. These results show that we are able to train a model on the generated images that can exceed the primitive baseline by far, even reaching up to 75% in accuracy on the given eight classes.

We observe that the performance of the retrained Resnet-18 model has a reliable and stable performance on both modes of image generation and both modes combined. The performance of this model in regards to the mode of image generation does not differ significantly. The Resnet-18 model clearly benefits from using images from both modes combined leading to a score of 75% at maximum with a dataset size of 400.000 images. We can see that the model performance correlates with the dataset size and the variety in images, as the performance when trained on images from both modes combined surpasses the performance when trained only on images from one mode.

The retrained models are typically biased towards predicting Volkswagen which is the most common brand in the photographs of the real traffic. We also observe a difference in accuracy per brand. Volkswagen, Ford, BMW, Audi and Mercedes achieve a performance of at least 70% while Opel, Renault and Skoda are only correct in about half of the cases.

5 Conclusion

We are able to train image classifiers for real world data solely on synthetic images that require no human labeling. The images we evaluate the classifiers on are taken in real moving traffic. Therefore, we face challenges such as a large variety of objects, image artifacts, different lighting, low resolution and motion blur. The generated data is sufficient to exceed the primitive baseline by far. These results are achieved whilst needing human work only for engineering the pipeline, tweaking hyperparameters and labeling the validation and test images. Thus the engineered pipeline may illustrate a potential approach to overcome challenges associated with traditional data acquisition methods.

On average the Resnet-18 performs better when retrained on the combined images instead of the same amount from just one mode of image generation. This may result from the fact that both modes combined cover a broader variety regarding the characteristics of the images. We assume that using varying prompts and varying parameters for the Stable Diffusion model could increase the variation in images and could therefore be beneficial.

To illustrate one possibility of this pipeline: With our pipeline we are able to create a perfectly balanced dataset of 100.000 images by using both modes combined. We can directly train a Resnet-18 model on this generated dataset. The time to perform this consecutively on a singular Nvidia RTX 3060 without further optimizations sums up to about two days. This provides a solid baseline model on short term with very little human work required.

There are significant differences in performance of the retrained Resnet-18 per class. These differences may be explained by biases inside of the Stable Diffusion models as they are more likely trained on more images of Volkswagen than images of Skoda. Another may be

that, on one hand, brands like Volkswagen, Mercedes, BMW and Audi have very prominent visual features that are easy recognizable for humans. On the other hand, earlier models of Renault have very small logos and Skoda has a dark radiator grill with only a small logo as an identifier. This also can attribute to a lower performance for these brands.

The pipeline introduced in this work is possible as we can automatically assess and crop the output of the Stable Diffusion model with YOLO. For other computer vision tasks with classes that are a subset of the classes YOLO can predict, we can adapt the pipeline easily. However, for completely other classes one would have to engineer another way to implement the fourth step of the pipeline. As this presents a challenge, the possible use cases of the introduced pipeline are limited by the capabilities of object detection models like YOLO. Another limitation lies within the capability of Stable Diffusion as it is unlikely that these models can generate usable images for every situation.

Acknowledgments

Markus Lange-Hegermann acknowledges support from the German Federal Ministry of Education and Research (BMBF) for the project SyDaPro with grant number 01IS21066A.

References

1. A. Krizhevsky *et al.*, "ImageNet classification with deep convolutional neural networks," in *NeurIPS*, vol. 25, 2012.
2. K. He *et al.*, "Deep residual learning for image recognition," 2015.
3. J. Redmon *et al.*, "You only look once: Unified, real-time object detection," 2016.
4. F. Zhuang *et al.*, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, 2021.
5. J. Krause *et al.*, "3d object representations for fine-grained categorization," in *IEEE ICCVW*, 2013.
6. R. Rombach *et al.*, "High-resolution image synthesis with latent diffusion models," 2022.

7. J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *IEEE CVPR*, 2009.
8. S. Azizi *et al.*, "Synthetic data from diffusion models improves imagenet classification," 2023.
9. M. B. Sariyildiz *et al.*, "Fake it till you make it: Learning transferable representations from synthetic imagenet clones," in *CVPR*, 2023.
10. R. He *et al.*, "Is synthetic data from generative models ready for image recognition?" *ICLR, spotlight*, 2023.
11. A. Nichol *et al.*, "GLIDE: towards photorealistic image generation and editing with text-guided diffusion models," *CoRR*, vol. abs/2112.10741, 2021.
12. Z. Liang *et al.*, "Covid-19 pneumonia chest x-ray pattern synthesis by Stable Diffusion," in *IEEE SSIAI*, 2024.
13. P. Patcharapimpisut and P. Khanarsa, "Generating synthetic images using Stable Diffusion model for skin lesion classification," in *16th KST*, 2024.
14. B. van Breugel *et al.*, "Can you rely on your model evaluation? improving model evaluation with synthetic test data," in *NeurIPS*, vol. 36, 2023.
15. Y. Zhou *et al.*, "Image-based vehicle analysis using deep neural network: A systematic study," in *IEEE ICDSP*, 2016.
16. W. Liu *et al.*, "An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors," *IEEE Access*, vol. 5, 2017.
17. M. Taqiyuddin *et al.*, "Accuracy improvement of cnn mobilenet-v1 and residual network 50 layers models using adam setting for car type classification," in *ISESD*, 2022.
18. J. Kim, "A study on the trend of vehicle types and color classification technology for intelligent transportation systems," in *IEEE ICCE-Asia*, 2021.
19. W. Lu *et al.*, "Category-consistent deep network learning for accurate vehicle logo recognition," *Neurocomputing*, vol. 463, 2021.
20. A. S. Rao *et al.*, "Identification of car make and model using deep learning and computer vision techniques," in *AIDE*, 2022.
21. Kraftfahrt-Bundesamt, "Official registrations of cars by segments and models (fz12, fz2, sv 4.2 - 2023)," 2023.
22. P. von Platen *et al.*, "Diffusers: State-of-the-art diffusion models," https://github.com/huggingface/diffusers, 2022.
23. Huggingface, "Using Diffusers Stable Diffusion XL Turbo," https://huggingface.co/docs/diffusers/using-diffusers/sdxl_turbo, 2023.

Robuste Ampeldetektion und Haltelinienfreigabe durch Kartenassoziation in automatisierten Fahrzeugen

Robust traffic light detection and stopline release by map association for automated driving

Richard Fehler[1], Kevin Rösch[1], Fabian Immel[1] und Christoph Stiller[1,2]

[1] FZI Forschungszentrum Informatik, Intelligent Systems and Production Engineering
Haid-und-Neu-Straße 10-14, 76131 Karlsruhe
[2] KIT, Institut für Mess- und Regelungstechnik
Engler-Bunte-Ring 21, 76131 Karlsruhe

Zusammenfassung In dieser Arbeit präsentieren wir eine in Deutschland auf öffentlichen Straßen erfolgreich erprobte Systemarchitektur, um robust Haltelinien zugeordnete Ampeln wahrzunehmen und den resultierenden Freigabezustand zu filtern. Ampeldetektionen werden mit Ampeln aus einer HD-Karte [1] assoziiert, um sie der Haltelinie zuzuordnen die das Fahrzeug betrifft. Somit kann die in der Karte hinterlegten Beziehung zwischen Ampeln und Haltelinie genutzt werden, ohne auf eine fehleranfällige Rückprojektion angewiesen zu sein. Wir evaluieren das Gesamtsystem anhand ausgewählter Szenarien in Karlsruhe und Sindelfingen und zeigen damit die Einsatzbereitschaft in realen automatisierten Fahrzeugen.

Schlüsselwörter Deep learning, autonomes Fahren, Ampeln, Detektion, Assoziation

Abstract In this work, we present a system architecture that has been successfully tested on public roads in Germany to robustly perceive traffic lights assigned to stop lines and filter the resulting release state. Traffic light detections are associated with traffic lights from an HD map [1] to assign them to the stop line relevant to the vehicle. This allows the use of the relationship be-

tween traffic lights and stop lines stored in the map, without relying on an error-prone back-projection. We evaluate the overall system using selected scenarios in Karlsruhe and Sindelfingen, demonstrating its readiness for deployment in real automated vehicles.

Keywords Deep learning, autonomous driving, traffic lights, detection, association

1 Einleitung

1.1 Stand der Technik

Ermöglicht durch die hochgenaue 3D Kartierung und Lokalisierung haben automatisierte Fahrzeuge den für Sie relevanten Ampelzustand bereits vor der Entwicklung von Deep-Learning-basierter Objektdetektoren erfolgreich durch die Rückprojektion und anschließender Klassifizierung geschätzt [2]. Moderne Ansätze verwenden Deep-Learning-Detektoren, trainiert auf speziellen Ampeldatensätzen [3]. Arbeiten die das gesamte System und nicht nur die Wahrnehmung im Rahmen des hochautomatisierten kartenbasierten [1] Fahrens entwerfen und evaluieren gibt es in deutlich geringerem Umfang [2].

1.2 Ziel der Arbeit

Die Rückprojektion von in 3D kartierten Ampeln in das Kamerakoordinatensystem und anschließende Klassifikation des Ampelzustandes ist sehr anfällig für Lokalisierungs-, Kalibrierungs- oder Kartierungsfehler. Der Beitrag dieser Arbeit ist der Entwurf und die Umsetzung eines robusten Systems zur Freigabe von durch Ampeln geregelten Haltelinien.

2 Methode

Die Systemarchitektur wird in die Module *Detektion*, *Klassifikation*, *3D Erweiterung*, *Assoziation*, und *Filterung* der Sektionen 1-5 unterteilt. Der

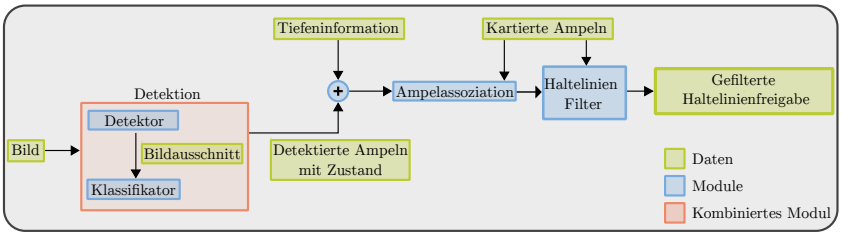


Abbildung 1: Die Systemarchitektur eines robusten Freigabesystems für Ampelhaltelinien.

Datenfluss von Eingabe-Bildern bis zu den gefilterten Haltelinienfreigaben der HD-Karte wird in Abb. 1 in Relation zu den Funktionsmodulen gesetzt um eine Meta Architektur zu bilden.

2.1 Ampeldetektion

Um die instabile Rückprojektion von kartierten Ampeln zur Bestimmung des Ampelzustands zu umgehen ist eine Ampeldetektor mit ausreichend hoher Genauigkeit und Trefferquote notwendig. Wir verwenden einen Faster-RCNN [4] Detektor mit einem ResNet50 [5] Rückgrad, trainiert auf dem Microsoft COCO Datensatz [6], welcher Ampeln in ausreichend hoher Anzahl annotiert hat. Dieser zweistufige Box Detektor liefert gute Detektionsergebnisse auch für kleine Objekte bei Bildern mit Auflösungen von über 3000 Pixel Bildbreite. Die Inferenzzeit beträgt 70 ms bei Nutzung von PyTorch [7] und einer RTX 6000 Ada GPU. Die Boxen der Ampeldetektionen werden an den feingranularen Typ- und Zustandsklassifikator innerhalb des kombinierten Moduls *Detektion* im selben Prozess überreicht.

2.2 Klassifikation

Wir generieren Bildausschnitte für jede detektierte Ampel basierend auf den Detektionen. Diese Bildausschnitte werden durch ein doppelköpfigen Klassifikator in Zustand und Typ eingeordnet, entsprechend der *pictogram* und *state* Attribut-Klassen des DriveU Traffic Light Dataset (DTLD) [8]. Der Klassifikator basiert auf einem modifizierten

Abbildung 2: Die in Typ und Zustand klassifizierten Ampeldetektion. Die Farbe der Box entspricht dem Ampelzustand. Unbekannte Zustände sind in pinker Farbe gekennzeichnet. Am besten digital zu betrachten.

EfficientNet [9] Modell welches auf den Bildausschnitten der annotierten Boxen des DTLD Datensatzes trainiert wurde.

Tabelle 1: Die durch den doppelköpfigen Klassifikator eingeordneten Zustand- und Typklassen.

Ampelzustand	Ampeltyp
off	circle
red	arrow_left
yellow	arrow_right
red_yellow	arrow_straight
green	arrow_straight_left
unknown	arrow_straight_right
	tram
	bicycle
	pedestrian
	pedestrian_bicycle
	unknown

2.3 Erweiterung in den 3D Raum

Die Boxen der Ampeldetektionen werden durch Tiefeninformationen in den 3D Raum erweitert. Die Tiefeninformationen können dabei dynamisch von verschieden Sensorquellen und Tiefenschätzungsmethoden gewonnen werden. Wenn genügend Lidar-

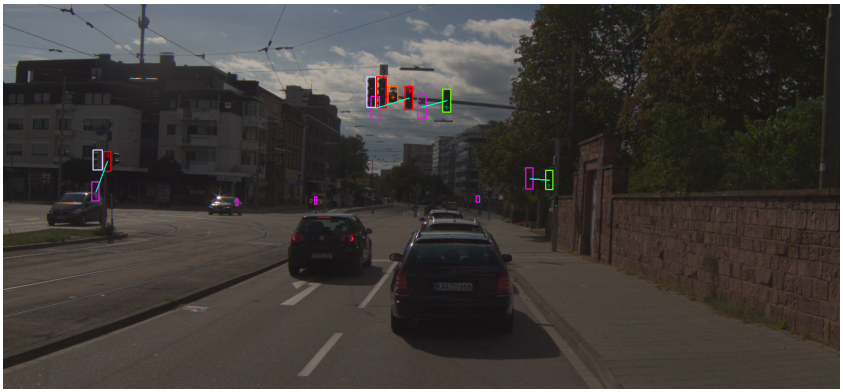


Abbildung 3: die Ampeln der HD-Karte werden mit den in den 3D Raum erweiterten Detektionen assoziiert (türkise Verbindungslinie). Die Ampeln der HD-Karte werden zur Visualisierung des Lokalisierungsfehlers in das Kamerabild rückprojiziert (Pink).

punkte in die Box der Detektion fallen, werden aus der Menge der Lidarpunkte durch Gewichtung und Clustering ein skalarer Tiefenwert berechnet. Wenn die Box ausserhalb des Lidarstrahlbereichs liegt, kann eine dichte Tiefenkarte geschätzt werden, basierend auf Stereokameras oder Monokularen Kameras um den skalaren Tiefenwert der Box zu bestimmen. Durch Ausnutzung der normierten Ampeldimensionen haben sich heuristische Methoden zur Tiefenschätzung der Ampel als möglich erwiesen. Wenn fehlende Lidardaten und eine geringe Rechenleistung das System einschränken, kann eine objektbedingte Tiefenschätzung direkt durch einen 2D Detektor gelernt werden [10]. In dem evaluierten System kommt ein 128 Zeilen Lidar zum Einsatz mit einer heuristischen Tiefenschätzung für den vom Lidar nicht abgedeckten extremen Nahbereich.

2.4 Assoziation

Die nun in den 3D Raum erweiterten Detektion werden den kartierten Ampeln in einem bipartiten Graph zur Assoziation gegenübergestellt. Wir formulieren das Assoziationsproblem als Minimum Cost Flow Problem [11, 12]. Gegenüber der weiter verbreiteten Lösung mit dem

Hungarian Algorithmus für optimale 1 zu 1 Assoziation, stellt das Minimum Cost Flow Problem eine Verallgemeinerung dar. Es ist hiermit auch die Assoziation von meheren Detektionen zu einem Kartenelement oder umgekehrt möglich. Dies erlaubt es Doppeldetektionen robust abzufangen, welche die Ergebnisse bei herkömmlicher 1 zu 1 Assoziation stark beeinträchtigen. Weitere Vorteile sind die intuitive Einführung eines Assozationskostenmaximums und die Möglichkeit, beispielsweise die global besten 5 Assoziationen zu liefern,formuliert in einem einzigen Optimierungsproblem.

Die Kostenfunktion berücksichtigt den Ampeltyp, die Dimensionen der Ampel und ihre 3D Position. Der bipartite Graph wird, wie bei dem Minimum Cost Flow Problem üblich, als Flussgraph einer Quelle zu einer Senke dargestellt, der dabei die bipartiten Knoten der detektierten und kartierten Ampeln passieren muss. Der Quellfluss kann als die Anzahl der kartierten Ampeln in einem Suchradius, die Anzahl der Detektionen oder das geringere von beiden gewählt werden, wobei wir in unseren Versuchen letztere Option verwenden. Ein Kostenmaximum filtert Assoziation mit zu hohen Kosten. Die Typen *bicycle*, *pedestrian* aus Tabelle 1 und deren Kombination werden nicht mit Fahrzeugampeln der Karte assoziiert. Als *tram* klassifizierte Typen werden weiterhin verwendet, aber mit höheren Assozationskosten verbunden, da abgeschaltete Fahrzeugampeln häufig fälschlicherweise dem Typ *tram* zugeordnet werden.

2.5 Filter

Für jede kartierte *Haltelinie* werden die assozierten Zustände *aller* der Linie zugehörigen Ampeln durch einen gleitenden Fensterfilter geglättet. Dies sorgt für eine robuste Haltelinienfreigabe auch bei Verdeckungen oder instabilen Detektionen und Klassifizierungen durch schlechte Sichtbedingungen. Wenn nicht genügend aktuelle *green* oder *off* Zustände im Fenster vorhanden sind wird die Haltelinie geschlossen. Durch den robusten Entwurf können auch instabile Wahrnehmungsergebnisse zu korrekten Haltelinienfreigaben führen, indem bei unbekanntem und somit geschlossenem Freigabezustand das Fahrzeug verzögernd in bessere Sichtverhältnisse rückt.

3 Ergebnisse

Das System wurde im realen Strassenverkehr in Karlsruhe und Sindelfingen im automatisierten Testbetrieb eines Versuchsfahrzeugs evaluiert. Die präsentierte Versuchsreihe zeigt ein korrekte Freigabe in über 96% der Haltelinienüberfahrungen der Erprobungsstrecken, auch bei Erprobungen in Regen oder der Dämmerung.

3.1 Klassifikationsergebnisse

Der gewählte Detektor erziehlt auf dem COCO Validierungsdatensatz eine mAP von *46.7*. Diese Detektionsperformance ist die Basis für alle weiteren Teile des Systems. Der auf Boxendetektionen aufbauende doppelköpfige Klassifikator wurde auf dem DTLD [8] Datensatz trainiert. Die initialen Gewichte für das Modell wurden auf dem ImageNet-1k Datensatz [13] vortrainiert. Die Hyperparameter des Modells und des Trainingregimes sind in Tabelle 2 aufgeführt.

Parameters	**Value**
Model	Efficient Net [9]
Optimierer	ADAM [14]
Kostenfunktion	Kreuzentropie
Lernrate	$1e^{-3}$
Epochen	60
Bildgröße	224^2
Batch size	64

Tabelle 2: Die Hyperparameter des Klassifikators.

Mit diesen Parametern erreicht das Netzwerk eine Zustandsklassifikator eine Genauigkeit von 96% auf dem DTLD Datensatz. Das Netzwerk wurde besonders wegen der schnellen Trainingszeit und der effiziente Inferenzzeit von unter 30ms ausgewählt.

3.2 Systemevaluation im Strassenverkehr

Wir befahren in einem automatisierten Versuchsfahrzeug eine wie in Abb. 4 und Abb. 5 zu sehende kartierte Route in Karlsruhe und Sindelfingen innerhalb von vier Monaten mehrfach ab um die System-

Abbildung 4: Die Evaluierte Route in Karlsruhe mit einer Länge von über 7 km. Luftbilder von Bing Maps © 2024 Microsoft Corporation

funktion zu bewerten. Dabei zählen wir die korrekten Ampelhaltelinienzustände bei An- und Überfahrung der Haltelinie und die entsprechend inkorrekten Ampelhaltelinienzustände. Die Befahrungen haben an sonnigen Tagen, an bewölkten Tagen und auch bei Dämmerung und Starkregen stattgefunden. folgende Probleme haben dabei zu Systemfehlern geführt:

- Regen und schlechte Sicht sorgen für fehlende Detektionen von insbesondere abgeschalteten (*off*) Ampeln und damit zu einem unbekannten Zustand der Ampeln oder falschen Assoziationen mit leuchtenden Ampeln.
- hoch montierte, rot leuchtende Ampeln die teilweise wie in Abb. 6 abgeschnitten werden, werden detektiert, jedoch als *off*

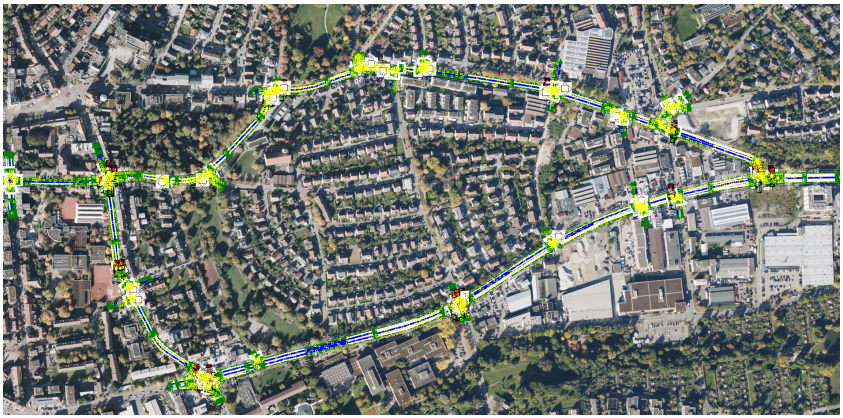


Abbildung 5: Die Evaluierte Route in Sindelfingen mit einer Länge von über 4 km.Luftbilder von Bing Maps © 2024 Microsoft Corporation

Abbildung 6: hoch montierte (links oben), rot leuchtende Ampeln die teilweise im Bild abgeschnitten werden. Detektiert aber fälschlicherweise als *off* klassifiziert, da keine rote Leuchte zu sehen ist. Die Haltelinie wird durch zwei Ampeln geregelt, bei einer von zwei falsch klassifizierten Ampeln kann der Filter falsche Ergebnisse liefern.

klassifiziert.

- Ampeln die Aufgrund der Fahrzeugausrichtung nicht in das Frontkamerabild fallen, können durch diese nicht beobachtet werden.

Route	korrekt	inkorrekt	# Haltelinien	Präzision (%)
KA	155	5	160	96.88
SiFi	49	2	51	96.08
Gesamt	**204**	**7**	**211**	**96.68**

Tabelle 3: Anzahl korrekter und inkorrekter Haltelinienzustände entlang der Routen in Karlsruhe (KA) und Sindelfingen (SiFi).

Eine falsche Ausrichtungen des Fahrzeugs kann durch Erweiterung der Detektion und Assoziation auf das 360 Grad Ringkamera System des Fahrzeugs ausgeglichen werden. Die Problematik abgeschnittener Ampeln kann durch einen angepassten Sensoraufbau, wie vertikale Orientierung der vorderen Kameras oder durch eine angepasste Haltedistanz zu Ampeln und deren Haltelinien durch die Trajektorienplanung gelöst werden. Den Einfluss der Trainingsdaten auf die Detektionsrate bei schlechten Sichtbedingungen muss weiter untersucht werden. Eine aktive Lernstrategie kann hier angewandt werden um fehlende Bedingungen in den Trainingsdatensätzen auszugleichen.

4 Zusammenfassung

Der vorgeschlagene Entwurf und dessen Umsetzung eines robusten Systems zur Freigabe von durch Ampeln geregelten Haltelinien konnte den Haltelinienzustand in 96% der 211 evaluierten Haltelinienüberfahrungen in Karlsruhe und Sindelfingen schätzen und hat den automatisierten Betrieb eines Versuchsfahrzeugs ermöglicht. Das System zeigt ein robustes Verhalten gegenüber Lokalisierungs-, Kalibrierungs- oder Kartierungsfehlern und die verbleibenden beobachteten systematischen Schwachstellen des Systems wurden erörtert.

Literatur

1. F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2018.
2. N. Fairfield and C. Urmson, "Traffic light mapping and detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
3. K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
4. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.
5. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
6. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.
7. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
8. A. Fregin, J. Muller, U. Krebel, and K. Dietmayer, "The driveu traffic light dataset: Introduction and comparison with existing datasets," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
9. M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2019.
10. J.-H. Pauls, R. Fehler, M. Lauer, and C. Stiller, "Combining 2d and 3d datasets with object-conditioned depth estimation," in *IEEE Intelligent Vehicles Symposium (IV)*, 2022.
11. A. V. Goldberg and R. E. Tarjan, "Finding minimum-cost circulations by successive approximation," *Mathematics of Operations Research*, 1990.
12. A. V. Goldberg and M. Kharitonov, *On implementing scaling push-relabel algorithms for the minimum-cost flow problem*, 1992.

13. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.
14. D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

AI scratching your car: Using diffusion models for training data generation in automotive damage detection

Julian Strietzel[1], M. Saquib Sarfraz[1,2], and Rainer Stiefelhagen[1]

[1] Karlsruhe Institute of Technology
[2] Mercedes-Benz Tech Innovation

Abstract Demand for reliable data remains a major issue in training machine learning models in computer vision. Frequently, datasets are of insufficient scale, imbalanced, not diverse, and of poor quality, potentially resulting in biased, inaccurate, non-robust, and badly generalizing models. Moreover, real-world training data can raise privacy concerns or be extremely expensive to gather, necessitating alternative solutions.
This paper investigates the use of diffusion models for generative data augmentation in semantic image segmentation, specifically in the domain of vehicle damage detection. We propose a new approach that utilizes an existing diffusion model ControlNet to generate useful synthetic data depicting realistic vehicles with damages such as scratches, rim damages, dents and etc. Based on this we provide an analysis and show how such a generative data augmentation may help in scenarios where training data is scarce and of low quality.

Keywords Generative data augmentation, diffusion models, ControlNet, damage detection

1 Introduction

A major challenge in Deep Learning for Computer Vision persisting is the scarcity and quality of training data, which is crucial for training robust and generalizing models. Acquiring a large quantity of detailed and balanced images for training is often time-consuming, expensive,

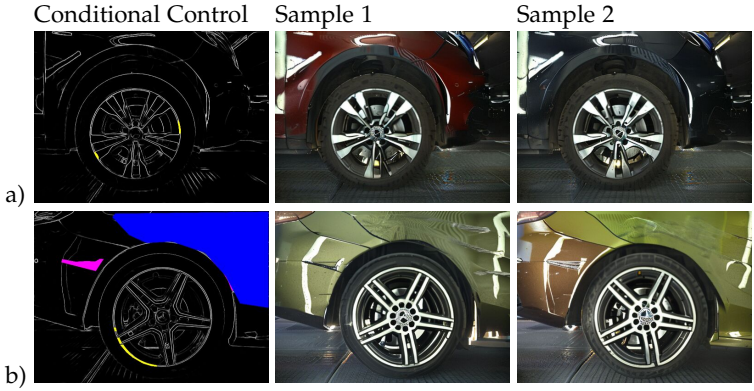


Figure 1: Edge detection maps with color patches as labels used as conditional control input and respective generated images from our trained ControlNet. With a) rim damages (yellow), b) deformations (blue), scratches (pink), and rim damages (yellow).

and sometimes impossible. This paper aims to address the challenge of limited training data in the domain of vehicle damage detection by investigating the use of ControlNet [1] for Generative Data Augmentation (GDA) [2] in scarce and low-quality data scenarios. We trained ControlNet based on StableDiffusion (SD) to generate synthetic images of damaged cars from labeled edge-detection maps and use these generated images to train segmentation models for damage detection (see fig. 1).

In automotive damage detection, semantic segmentation models may be used to recognize various types of exterior damages on images for efficient vehicle inspection. In practical industrial setting usually images of passing cars are taken autonomously from multiple angles by vehicle scanners and sent to cloud processing, to recognize several damage classes. Due to the nature of the damages it is quite impractical, even impossible in some cases to collect such data manually.

Guiding this are the questions about (1) how synthetic training data transfers to real-world evaluation, (2) its capacity to tackle challenges of scarcity, quality, and bias in training data, and (3) the effect on model generalizability. In the process, we evaluate parameters, design decisions, and training data compositions in extensive ablation studies and

experiments. The use of GDA has not been explored yet on this problem and may increase the potential of synthetic data in vehicle damage detection scenarios.

2 Related Work

Data augmentation addresses challenges like data scarcity, lack of diversity, and overfitting by creating new label-preserving examples from existing datasets [3]. Common augmentation approaches include image manipulation, image erasing, image mixing, auto-augment, feature augmentation, and neural style transfer [3]. GDA involves supplementing training data with synthetic examples to improve model performance, especially when only little training data is available and overfitting is of concern [2]. Classic methods in computer vision include CGI placement [4], model renderings [5–7], and degrading techniques [8]. Training data generation may help increase diversity, generalization capabilities, and robustness including to adversarial attacks.

Synthetic data from Denoising Diffusion Probabilistic Models (DDPM) [9] prove to be effective for GDA in ImageNet [10] classification [11], even achieving new state-of-the-art scores using supplemented real training data [12]. Synthetic data is also employed to fight representation bias [13, 14] and privacy concerns [15] in medical image data, with curation of synthetic data proving important [14]. For segmentation model training, mostly Generative Adversarial Networks (GANs) [16] have been employed to generate samples, using a decoder to extract pixel-wise annotations from latent space [17, 18], and showing performance gains mainly in out-of-domain data. Only recently DDPMs have been explored for GDA, mostly following a similar approach, extracting labels from attention-maps [19] or training a grounding model to align pixels with textual representations [20]. Apart from their superior sample quality [21], DDPMs for GDA face challenges including dataset memorization [22], diversity [23] and do not generally outperform GANs from scratch [24].

ControlNet is a neural network structure aimed to further condition the output of DDPMs [1]. ControlNet is a copy of an arbitrary neural network block, running in parallel to the original network, incorpo-

rating an encoding of additional control input and feeding its guided output back to the main structure. During training of ControlNet, the original model is locked to preserve its distilled knowledge, only the parallel, duplicated blocks are trained for guidance. Using this architecture we can control DDPMs to exactly match input conditions, like edge detection maps, human poses, or drawings, even with comparably low training data available.

3 Methodology

The core part of this paper is the implementation of ControlNet generating pre-labeled data as GDA for segmentation model training for damage detection. We consider four commonly occurring damage classes: deformation, dent, rim damage and scratch.

In this section, we discuss how to guide and train ControlNet to generate pre-labeled samples for semantic segmentation training.

Conditional Control We utilize ControlNet's conditional control feature to generate precise images representing specific views of cars and damage positions. Edge detection maps, identifying image boundaries, serve as the control input. This approach offers a balance between detailed output descriptions and the freedom to generate varied results and has proven to work well with ControlNet [1].

For pre-labeled sampling, we need to include detailed label information in the conditional control. We propose to include the labels using color patches on the black-and-white edge detection maps (see fig.1). This approach of labeling the conditional control to generate pre-labeled training data has, to our knowledge, never been evaluated before.

It has the following advantages: (1) Efficient placement and generation from existing labels, (2) effective guidance for ControlNet, (3) easily differentiable by eye, (4) covering edge detection maps on relevant positions, and (5) referenceable in text prompts by naming the respective color.

Text prompts play a crucial role in text-to-image generation, functioning as fundamental guidance, and imparting context and semantic information to the generative process. When using ControlNet, the

textual prompt introduces background information guiding the interpretation of the conditional control. The integration of semantically relevant textual information to image generation results in more precise and sophisticated outcomes, potentially improving its capability for GDA.

Text prompts are generated following a specific prompting schema: (1) The applying short description of the relevant damage: *Rim damage at the yellow marking, Scratch at the pink marking, Dent at the green marking, Deformation at the blue marking*, (2) a background prompt to define the image, context, and style: *side of a car in a workshop, high quality, detailed, and professional image*.

Training Generative Model We train the generative model on the available real-world training data, to generate damaged cars matching the conditioning. Apart from its comparably low requirements on training data, ControlNet training is exhibiting a sudden convergence phenomenon [25], which we take into account asjusting the virtual batch size using gradient accumulation to reach around 10k steps during training.

Fine-Tuning In visual evaluation, samples of different damage classes showed significant deviations in image quality, suggesting that the generative model might be improved by fine-tuning it on specific damage classes. We filtered the 17k dataset to include only images containing instances of the respective class and trained a generative model for each one. We name these fine-tuned models (damage-class-) *specific*.

4 Experiments

Experiments have been split into (1) tuning and evaluating the image generation process and (2) optimizing training of segmentation models from (partly) synthetic data. The used dataset includes 17k hand-labeled images from a vehicle scanner containing from 1k to 9k instances per damage class (see appendix).

4.1 Image Sampling

Firstly, we conducted experiments to evaluate the quality of synthetic images for different parameters and ablations. To measure improvements, we employ an existing segmentation model, trained on the existing real17k dataset and evaluate it on our generated datasets. We expect correlation between sample quality and the model's ability to recognize synthetic damages, measured by the evaluation F-Scores.

We evaluated the prompting schema defined in section 3: As the models had been trained with the fixed schema, additional background and negative text prompts, as well as no prompts at all resulted in worse samples. This suggests that our prompts need to stick to the scenario or have to be trained on a more diverse prompt landscape. We also evaluated a pre-trained ControlNet from a large edge detection dataset on damage generation, which was not able to extract meaningful results from the guidance input, though.

4.2 Training Data Compositions

Evaluating a model trained from synthetic data on the real evaluation dataset, a first approach showed a significant domain gap between real and synthetic images, with F-Scores of less than 0.1. In the second part of our experiments, we therefore evaluate (only partly) synthetic training data compositions to train segmentation models based on their downstream performance on real evaluation data.

Damage-Specific Training Data To evaluate samples from damage specific generative models, we combined 2k samples for each class with 2k samples from a general model to a new dataset - *specific10k* - for segmentation model training. Compared to 10k samples from the general model, we seem to slightly improve F-Scores on average, reflecting the the results from class-specific sample evaluation. Furthermore, we are able to improve over *real17k* training data in the deformation class, representing the most scarce and low-quality training data, increasing the F-Score by $\sim$ .03 when using specific training data.

We also supplemented fake training data to real17k, instead of using it isolated, and employed a quality filter to curate the samples for training. We used an IOU threshold of 25% per image from evaluating

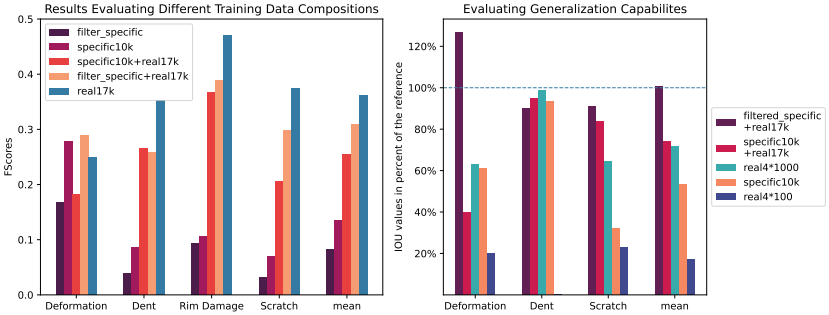


Figure 2: Left: F-Scores of segmentation models trained using different (synthetic) datasets evaluated on real test data. *Specific10k* refers to samples from damage-specific generative models, with *real17k* referring to the original training data as reference. *Filtered* referencing to a quality threshold. **Right**: Generalization IOU values of trained segmentation models (synthetic and limited training data) on out-of-domain data in proportion to the same reference model.

the images by a pre-trained segmentation model as in image sampling experiments (section 4.1). Using both supplementation and filtering greatly reduced the synthetic-to-real-data gap, resulting in only .04 difference in macro average scores. This was primarily due to a further improvement in deformation accuracy (see fig. 2), where we increased the existing lead over real training data. Notably, we still decrease overall performance by supplementing fake data to our training, especially in well-represented classes. This suggests a key difference in data distribution regardless of the visual quality of samples, but we show potential application and benefit of GDA for very scarce classes.

Comparison on limited data To find a threshold of data availability where synthetic samples outperform real training data, we limit available real training data (to 25, 100, 250, and 1000 examples per class). As expected, limiting the availability of real training data negatively impacts overall segmentation performance. Decrease differs from class to class, with scarce classes (dent and deformation) benefiting from more balanced training data. Synthetic data outperforms very limited real datasets (25 samples per class) across all damage classes and enhanced datasets are competitive to larger real datasets (up to 1k), especially in

rim damage.

In this damage detection scenario, the targeted threshold seems to be between 25 and 100 images per class.

Generalizability To assess generalizability, we evaluate models trained with GDA on a new dataset of 250 labeled samples from unseen locations and vehicle scanners (out-of-domain dataset) and compare them to limited real datasets. Especially in the deformation class, the GDA-trained model (filtered, specific, and supplemented samples) significantly outperforms the reference model in the out-of-domain setting by 25 % (see fig. 2). Even on average, filtered supplemented data outperforms real training data, with even non-filtered outperforming up to 1k real samples per class, and generated samples only still significantly dominating over 100 real samples. Synthetic data improves generalization performance compared to limited datasets, particularly outperforming the original training data in scarce classes. This underlines the potential of synthetic data especially when it comes to generalization, where GDA shows more competitiveness than during in-domain evaluation.

5 Discussion

We show how ControlNet with StableDiffusion can be effectively used to generate pre-labeled, high-fidelity images for GDA in image segmentation tasks. In the context of vehicle damage detection the model demonstrates the ability to accurately place damages on vehicles.

Our experiments and ablation studies have revealed several key factors that can contribute to optimizing ControlNet generative performance. Parameter tuning and input specifications significantly improved sample fidelity. The ablation studies further provided valuable insights into the role of various techniques guiding and training ControlNet: We discovered the necessity of fine-tuning and additional text prompts, incorporating quality guidance. Finally, tuning damage-class-specific generative models for specific damage classes is beneficial, compared to a general multi-class generative-model. We showed how our synthetic data can be effectively used to train segmentation models for damage detection: Synthetic data alone can enhance seg-

mentation performance for very scarce classes and generally outperform limited real data when only a few samples (less than 50) are available. Especially scarce classes can benefit from *additional* synthetic training data. Furthermore, GDA can, with some limitations, be used to increase the generalization capabilities of our segmentation models, where supplemented fake data is outperforming the real dataset, especially limited to a few hundred examples only. Key findings from this study align with prior research GDA.

As a key takeaway we note that filtered, specific generated datasets supplementing real data can increase in-domain and especially out-of-domain performance significantly for scarce data classes. However, synthetic-only data remains no match to real large scale datasets, due to a significant distribution shift between real and fake samples. This is emphasized by a significant performance drop when GDA-trained models are evaluated on real-world test data, indicating how synthetic data may not fully capture the nuances and variations present in real-world data. We show that GDA in this use-case is mostly not effective for well represented data classes.

Future Work To guide effective utilization of GDA we suggest employing synthetic data when real data is very scarce. When using ControlNet for GDA, stronger guidance and increased steps benefit sample fidelity. Furthermore, analyzing the characteristics of synthetic compared to real samples and employing inpainting for GDA might be promising directions.

Acknowledgment We would like to express our gratitude towards Mercedes-Benz Tech Innovation GmbH for providing valuable compute, data, and hardware that significantly contributed to the research presented in this paper. Their cooperation and support for this research played a crucial role in enabling us to carry out the experiments and analysis necessary.

We would also like to thank our colleagues at the Autonomous Systems Karlsruhe Team, for their valuable insights, constructive feedback, and collaboration throughout the research process. Their expertise and dedication have significantly contributed to the quality of this work.

6 Appendix

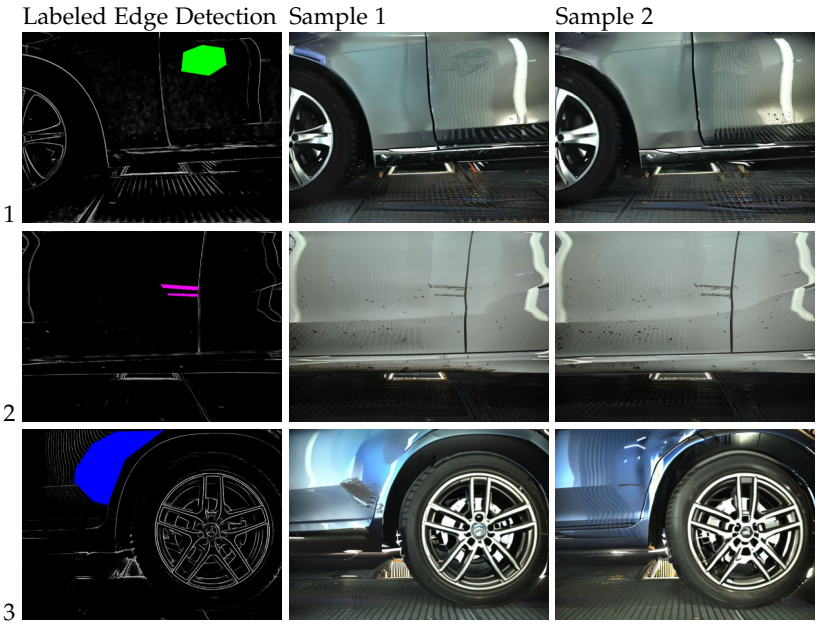


Figure 3: Sampled images of different damage classes from shown labeled edge detection maps as conditional control input for ControlNet.

Segmentation Model We use a U-Net [26] from the segmentation model library [27]. The reference model is trained on the real17k training data and does not represent the performance of similar models in production. **ControlNet** version 1.0 with StableDiffusion 2.1 is trained on the same T4 for 15 epochs on a virtual batch size of 32 for about 8,000 steps, taking around 100 hours per model. Virtual batch size is reached by using gradient accumulation of 32.

6.1 Datasets

The used dataset - real17k - contains 17467 manually labeled images of cars with different damages (9234 rim damages, 8685 scratches, 1803 dents, 972 deformations). The test dataset contains 739 images of which 177 include rim damages, 168 scratches, 104 dents, and 11 deformations. The in-domain images are taken from an automatic vehicle scanner at entry points to a workshop. They all come from the same location, taken with the same equipment, lighting conditions, background, and surroundings. The weather is similar with only a few images containing rainy or snowy conditions.

The **limited datasets** real4*x with $x \in 25, 100, 250, 1000$ contain a random sample from the real17k dataset. They are used to simulate a scenario, where only a limited but balanced amount of data is available.

The **out-of-domain** dataset to test generalization contains 170 images, with some being from different locations and types of vehicle scanners. The dataset contains 169 images of which 31 deformations, 32 dents, 0 include rim damages, and 128 scratches.

Synthetic Datasets The **specific10k** dataset of generated synthetic data contains 2k images per class generated from class-specific generative models and 2k images from a general model. The **filtered** dataset contains all images from specific10k, that passed a quality threshold established as an IOU greater than .25 in evaluation using a real-world pre-trained segmentation model: Deformation 339 (from 2940 samples containing instances in total), Dent 392 (3095), Rim Damage 876 (3804) & Scratch 490 (5164).

References

1. L. Zhang and M. Agrawala, "Adding Conditional Control to Text-to-Image Diffusion Models," Feb. 2023, arXiv:2302.05543 [cs]. [Online]. Available: http://arxiv.org/abs/2302.05543
2. C. Zheng, G. Wu, and C. Li, "Toward Understanding Generative Data Augmentation," May 2023, arXiv:2305.17476 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2305.17476
3. T. Kumar, A. Mileo, R. Brennan, and M. Bendechache, "Image Data Augmentation Approaches: A Comprehensive Survey and

Future directions," Mar. 2023, arXiv:2301.02830 [cs]. [Online]. Available: http://arxiv.org/abs/2301.02830

4. V. Shakhuro, B. Faizov, and A. Konushin, "Rare Traffic Sign Recognition using Synthetic Training Data," in *Proceedings of the 3rd International Conference on Video and Image Processing*, ser. ICVIP '19. New York, NY, USA: Association for Computing Machinery, Feb. 2020, pp. 23–26. [Online]. Available: https://dl.acm.org/doi/10.1145/3376067.3376105
5. W. Armstrong, S. Drakontaidis, and N. Lui, "Synthetic Data for Semantic Image Segmentation of Imagery of Unmanned Spacecraft," in *2023 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2023, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/10115564/
6. J. Adams, J. Sutor, A. Dodd, and E. Murphy, "Evaluating the Performance of Synthetic Visual Data for Real-Time Object Detection," in *2021 6th International Conference on Communication, Image and Signal Processing (CCISP)*, Nov. 2021, pp. 167–171.
7. M. Pergeorelis, M. Bazik, P. Saponaro, J. Kim, and C. Kambhamettu, "Synthetic Data for Semantic Segmentation in Underwater Imagery," in *OCEANS 2022, Hampton Roads*, Oct. 2022, pp. 1–6, iSSN: 0197-7385.
8. X. Nie, M. Yang, and R. W. Liu, "Deep Neural Network-Based Robust Ship Detection Under Different Weather Conditions," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct. 2019, pp. 47–52.
9. J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020, event-place: Vancouver, BC, Canada.
10. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y
11. M. B. Sariyildiz, K. Alahari, D. Larlus, and Y. Kalantidis, "Fake it till you make it: Learning transferable representations from synthetic ImageNet clones," Mar. 2023, arXiv:2212.08420 [cs]. [Online]. Available: http://arxiv.org/abs/2212.08420
12. S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, "Synthetic Data from Diffusion Models Improves ImageNet Classification," Apr. 2023, arXiv:2304.08466 [cs]. [Online]. Available: http://arxiv.org/abs/2304.08466

13. L. W. Sagers, J. A. Diao, M. Groh, P. Rajpurkar, A. Adamson, and A. K. Manrai, "Improving dermatology classifiers across populations using images generated by large diffusion models," in *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, 2022. [Online]. Available: https://openreview.net/forum?id=Vzdbjtz6Tys

14. M. Akrout, B. Gyepesi, P. Holló, A. Poór, B. Kincső, S. Solis, K. Cirone, J. Kawahara, D. Slade, L. Abid, M. Kovács, and I. Fazekas, "Diffusion-based Data Augmentation for Skin Disease Classification: Impact Across Original Medical Datasets to Fully Synthetic Images," Jan. 2023, publication Title: arXiv e-prints ADS Bibcode: 2023arXiv230104802A. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2023arXiv230104802A

15. S. Ghalebikesabi, L. Berrada, S. Gowal, I. Ktena, R. Stanforth, J. Hayes, S. De, S. L. Smith, O. Wiles, and B. Balle, "Differentially Private Diffusion Models Generate Useful Synthetic Images," Feb. 2023, arXiv:2302.13861 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2302.13861

16. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, Dec. 2014, pp. 2672–2680.

17. Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler, "DatasetGAN: Efficient Labeled Data Factory with Minimal Human Effort," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 10 140–10 150, iSSN: 2575-7075.

18. D. Li, H. Ling, S. W. Kim, K. Kreis, S. Fidler, and A. Torralba, "BigDatasetGAN: Synthesizing ImageNet with Pixel-wise Annotations," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Jun. 2022, pp. 21 298–21 308. [Online]. Available: https://ieeexplore.ieee.org/document/9878775/

19. W. Wu, Y. Zhao, M. Z. Shou, H. Zhou, and C. Shen, "DiffuMask: Synthesizing Images with Pixel-level Annotations for Semantic Segmentation Using Diffusion Models," Mar. 2023, arXiv:2303.11681 [cs]. [Online]. Available: http://arxiv.org/abs/2303.11681

20. Z. Li, Q. Zhou, X. Zhang, Y. Zhang, Y. Wang, and W. Xie, "Guiding Text-to-Image Diffusion Model Towards Grounded Generation," Jan. 2023, arXiv:2301.05221 [cs]. [Online]. Available: http://arxiv.org/abs/2301.05221

21. P. Dhariwal and A. Nichol, "Diffusion Models Beat GANs on Image Synthesis," in *Advances in Neural Information Processing Systems*,

M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf

22. M. U. Akbar, W. Wang, and A. Eklund, "Beware of diffusion models for synthesizing medical images – A comparison with GANs in terms of memorizing brain tumor images," May 2023, arXiv:2305.07644 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2305.07644
23. M. F. Burg, F. Wenzel, D. Zietlow, M. Horn, O. Makansi, F. Locatello, and C. Russell, "A data augmentation perspective on diffusion models and retrieval," Apr. 2023, arXiv:2304.10253 [cs]. [Online]. Available: http://arxiv.org/abs/2304.10253
24. M. U. Akbar, M. Larsson, and A. Eklund, "Brain tumor segmentation using synthetic MR images – A comparison of GANs and diffusion models," Jun. 2023, arXiv:2306.02986 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2306.02986
25. L. Zhang and M. Agrawala, "ControlNet/docs/train.md at main · lllyasviel/ControlNet." [Online]. Available: https://github.com/lllyasviel/ControlNet/blob/main/docs/train.md
26. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, ser. Lecture Notes in Computer Science, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.
27. P. Iakubovskii, "Segmentation Models," 2019, publication Title: GitHub repository. [Online]. Available: https://github.com/qubvel/segmentation_models

Image stitching using gradual image warping in autonomous driving

Christian Kinzig, Jiang Yifan, Martin Lauer, and Christoph Stiller

Karlsruhe Institute of Technology (KIT),
Institute of Measurement and Control Systems,
Engler-Bunte-Ring 21, 76131 Karlsruhe, Germany

Abstract To improve object recognition and tracking in autonomous driving, we first create a seamless panorama. Object recognition can benefit from image stitching, especially at the borders of individual images when an object is only partially visible. This also prevents duplicate detection of the same objects in overlapping image areas that are to be filtered for tracking. In this process, a homography is determined for the overlapping image area, whereby the entire image is transformed using classical image stitching methods. As a result, the deformations propagate to further images that are to be added to the panorama. To avoid this problem, we integrated a step-by-step image warping approach into our existing stitching pipeline. This ensures that after attaching one image to another, the outermost right and left borders of the panorama are no longer deformed. Furthermore, the panorama width remains constant regardless of the calculated homography. We have evaluated our approach on the nuScenes dataset and the Waymo Open Dataset for perception. In addition to a qualitative assessment, we evaluate the resulting panoramas in terms of the deformation of the individual images as well as the deformation of labeled object instances.

Keywords Autonomous driving, panorama, image stitching, homography, warping, deformation

Acknowledgements This research is accomplished within the project UNICAR*agil* (FKZ 16EMO0287). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

1 Introduction

The UNICAR*agil* [1,2] project, in which four autonomous vehicles were built entirely from scratch, investigated how and whether camera images should be stitched together to form a panorama before object recognition. One of the resulting articles [3] shows that object recognition performs just as well on panoramic images as on individual images without the need for retraining. In addition, in another article [4] we demonstrate that object detection on panoramic images improves compared to single images after retraining in this domain.

To stitch two images together, in a simple procedure, a homography is determined between pairs of feature matches in the overlapping image area to transform one of the images. However, this procedure for stitching images has the disadvantage that the resulting deformations increase with each additional images added to the panorama. For this reason, we have implemented a gradual image warping method based on the approach in [5]. Our main contribution is the elimination of deformations at the outermost right and left borders of the panorama, allowing any number of images to be stitched together horizontally. In this way, the transformations of all individual images can be calculated independently of each other. At the same time, the resulting panorama has a constant image width, which makes it more suitable as training data, as less zero padding needs to be applied. Furthermore, we decided to realize the local alignment not as a grid but as vertical image slices in order to reduce the computational effort.

2 Related Work

In the work by Zaragoza et al. in [6], a global homography between two images is first estimated, then equally sized grid cells in the image are transformed by local homographies to improve the alignment of the images to each other. In contrast, Chang et al. introduced a three-step process to preserve perspective by combining transformations from homography and similarity transformation in [7]. Based on this, Xiang et al. achieve smoother transitions by using weighted combinations of homography and similarity transformation in [5]. Chen and Chuang specifically aim for natural image stitching in [8] by using APAP [6] in

combination with a global similarity transformation to adjust scale and rotation for each image to be stitched. In [9], Zhang et al. developed a method specifically designed to return a rectangular panoramic image to reduce deformations in the images.

3 Implementation

The presented approach extends our image stitching method presented in [3] and [4]. Our image stitching pipeline is shown in Fig 1, with the modifications highlighted in blue. Thus, the deformation of the panorama towards the outermost right and left borders is gradually eliminated. The core components for gradual image warping can be divided into three consecutive steps, where first a homography in overlapping image areas is determined. In the second step, we divide the camera image into vertical sections and determine a transformation for each part of the image from the resulting homography. In the last step, we apply the resulting transformations to each image section and combine them to create a panorama.

3.1 Homography Estimation

The homography between two individual images p and q is determined by features in the overlapping image area. Consequently, the transformation of a feature point in a camera image p into another image q is given by (1).

$$\mathbf{H} \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = s \begin{pmatrix} u_q \\ v_q \\ 1 \end{pmatrix}, s \in \mathbb{R} \tag{1}$$

$\mathbf{H}$ stands for the homography and s for the scaling factor. As in [3], we do not perform feature extraction as well as subsequent feature matching. Instead, we use depth information as in a LiDAR point cloud, which we project into the overlapping image areas. Compared to image features, however, we have the disadvantage that an error occurs when projecting into cameras due to the parallax and the rotation of the LiDAR. This error can be reduced if the cameras are triggered as

soon as the LiDAR points in their direction. The Waymo Open Dataset for perception [10] also provides synchronized LiDAR data regarding the movement of the ego vehicle and the movement of other traffic participants. To calculate the homography, we use a method based on the RANSAC algorithm. In addition, learned methods as in [11] to determine a homography between two images would also be possible.

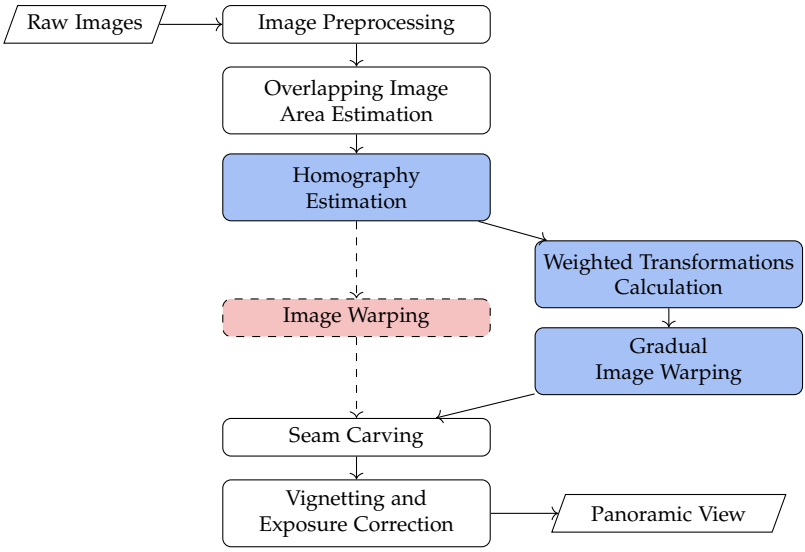


Figure 1: Workflow of our modifications shown in blue to the image stitching pipeline in [3] and [4] by integrating gradual image warping. Consequently, the image warping module shown in red is replaced.

3.2 Weighted Transformations Calculation

Using the homography determined in 3.1, the respective overlapping area is transformed. The opposite overlapping image area adjacent to the next camera image is not transformed. If there is no further camera image to be stitched, we assume a quarter of the image width that is not transformed. The remaining image area in between is warped gradually. First, a configurable parameter k, is used to define how many vertical image sections the center image area is divided into. This

determines how well the individual sections merge into one another. Similar to the approach in [5], we determine weighted transformations for each image section from two individual transformations, as shown in (2).

$$\mathbf{T} = \alpha\mathbf{H} + \beta\mathbf{I}_3 \tag{2}$$

However, we use the homography $\mathbf{H}$ determined in 3.1 and the identity matrix $\mathbf{I}_3$. For the first vertical image section adjacent to the overlapping image area associated with the homography, $\alpha = 1 - \frac{1}{k+1}$ and $\beta = \frac{1}{k+1}$. As the horizontal distance to the overlapping area increases, β gradually increases and α gradually decreases, so that the last vertical image section matches the overlapping area at the far end of the image. This allows any number of camera images to be stitched together horizontally without the deformations in the panorama becoming progressively larger towards the outside. In addition, the identity means that smoother video sequences can be created from the panoramas with a constant image width.

3.3 Gradual Image Warping

Once the transformation matrices have been determined for each individual image section, the image can be warped gradually. However, the transformations are still in image coordinates (u, v). In order to process smaller amounts of data and thus improve the runtime, we first transform each matrix $\mathbf{T}$ into the coordinate system of the respective image section (u_i, v_i) and denote the resulting transformation matrix as $\mathbf{T}_i$. The transformation into the vertical image sections can be described by a translation Δu. Finally, each transformation $\mathbf{T}_i$ is applied to the corresponding image section. These are subsequently projected onto the overall panoramic image.

4 Evaluation

First, we qualitatively compare our approach of gradual image warping with our previous approach as baseline described in [3] and [4].

Furthermore, we evaluate our approach also in comparison to our previous method using a quantitative measure of image deformation. In addition, we separately compare the deformations of labeled object instances. The two publicly available datasets nuScenes [12] and the Waymo Open Dataset for perception [10] are used in our evaluation.

(a) Individual images from which the panorama is composed using a spherical camera model.

(b) Image stitching using the method in [3] and [4].

(c) Image stitching with gradual image warping.

Figure 2: Comparison on image stitching using data from the nuScenes dataset [12].

4.1 Qualitative Comparison

To give an first impression of how our method performs, we use the two panoramas in Fig. 2 and 3 to show the comparison with the use of a homography per overlapping area. Fig. 2 compares both methods using an example from the nuScenes dataset [12] whereas Fig. 3 uses data from the Waymo Open Dataset for perception [10]. Both figures show that gradual image warping can better compensate for strong deformations. This applies in particular to the images from the outermost cameras. The curvature at the top and bottom of the images is due to the use of a spherical camera model, which can be seen in Fig. 2(a)

and 3(a). Particularly with video sequences, strong deformations are noticeable as jumps in the panoramas, as these do not remain constant. Gradual image warping ensures that the deformations are substantially smaller and more consistent. This could improve object tracking, especially if it is assumed that a detected object is in a similar position in the subsequent panoramic image.

(a) Individual images from which the panorama is composed using a spherical camera model.

(b) Image stitching using the method in [3] and [4].

(c) Image stitching with gradual image warping.

Figure 3: Comparison on image stitching using data from the Waymo Open Dataset [10].

4.2 Image Deformation Evaluation

To quantitatively evaluate gradual image warping, we determine the deformations in the warped images compared to the original images. In this case, the term original image refers to images that have already been processed but not warped for image stitching. Pre-processing consists of compensating for lens distortion and converting the image from a pinhole camera model to a spherical camera model.

To measure the deformation, we analyze points $\mathbf{p}_i$ evenly distributed over the images with a distance of 20 pixels. Then these points are

deformed to $\mathbf{p}_{warped,i}$ by image warping either with a single homography or with gradual image warping. First, we determine the average displacement $\overline{\mathbf{d}}$ between all N points in the deformed image and those in the original image, since a constant translation has no influence on the deformation. Accordingly, the average displacement $\overline{\mathbf{d}}$ is calculated separately for the directions u and v in (3).

$$\overline{\mathbf{d}} = \begin{pmatrix} \overline{d}_u \\ \overline{d}_v \end{pmatrix} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{p}_i - \mathbf{p}_{warped,i} \tag{3}$$

We then calculate the displacement between the points in the original images and in the warped images, taking into account the average displacement. This results in our error metric $\mathbf{E}_i$ in (4).

$$\mathbf{E}_i = \begin{pmatrix} E_{u,i} \\ E_{v,i} \end{pmatrix} = \left| \mathbf{p}_i - \mathbf{p}_{warped,i} - \overline{\mathbf{d}} \right| \tag{4}$$

The evaluation of the image deformation is performed on 10 sequences of the nuScenes dataset [12] and on 6 sequences of the Waymo Open Dataset for perception [10]. This results in an evaluation of 404 panorama images for nuScenes and 551 for Waymo. The results are displayed as two-dimensional box plots in Fig. 4 for the nuScenes dataset [12] and in Fig. 5 for Waymo Open Dataset for perception [10]. Both graphs clearly show that the deformations for gradual image warping are much smaller compared to the use of a single homography. The difference in deformation is most noticeable in the u direction. The smaller parallax in the Waymo Open Dataset results in significantly reduced warping on average. However, the outliers to the maximum are also higher in this case. The reason for this are the motion-compensated lidar point clouds. With high ego velocity or fast moving objects in the overlapping image area, significantly fewer point correspondences are available to calculate a homography.

4.3 Object Instances Deformation Evaluation

Especially in object recognition with machine learning, it is crucial that the results obtained on datasets can also be reproduced in the real

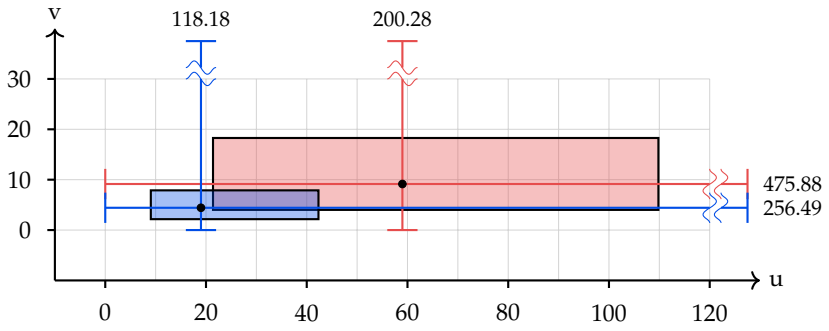


Figure 4: Image deformation analysis over 10 sequences of the nuScenes dataset [12]. 2D box plot of the deformations of the individual images in u- and v-direction in pixels with the method in [3] and [4] (red) compared to gradual image warping in (blue).

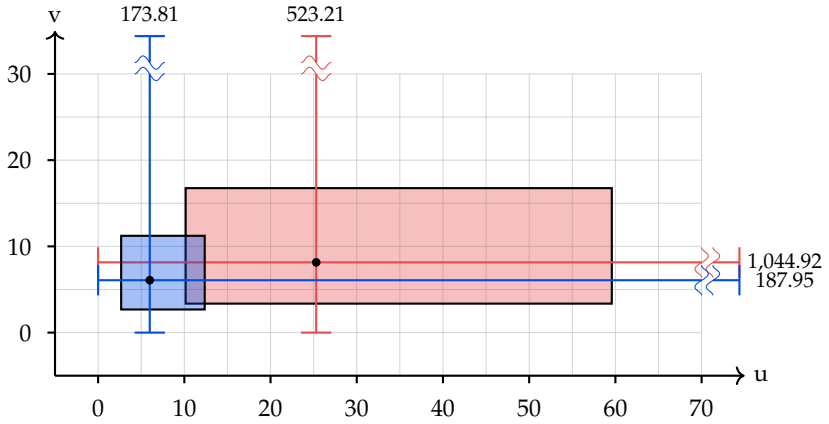


Figure 5: Image deformation analysis over 20 sequences of the Waymo Open Dataset for perception [10]. 2D box plot of the deformations of the individual images in u- and v-direction in pixels with the method in [3] and [4] (red) compared to gradual image warping in (blue).

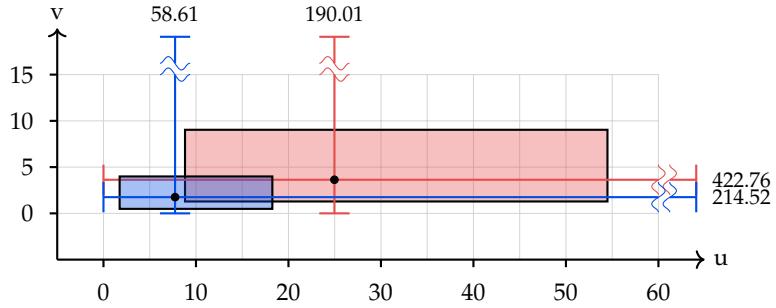


Figure 6: Deformation analysis of the object instances over 10 sequences of the nuScenes dataset [12]. 2D box plot of the deformations of the object bounding boxes in u- and v-direction in pixels with the method in [3] and [4] (red) compared to gradual image warping in (blue).

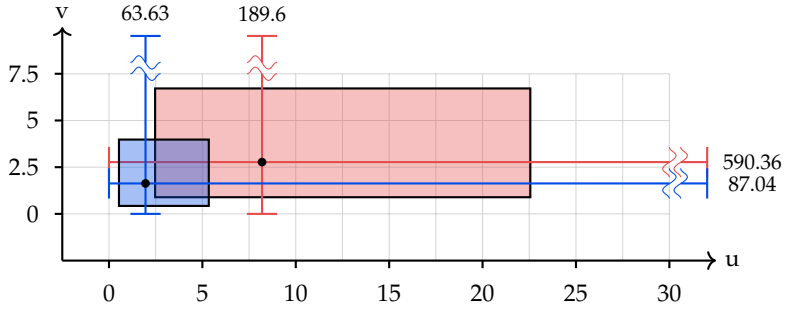


Figure 7: Deformation analysis of the object instances over 20 sequences of the Waymo Open Dataset for perception [10]. 2D box plot of the deformations of the object bounding boxes in u- and v-direction in pixels with the method in [3] and [4] (red) compared to gradual image warping in (blue).

world. Object recognition based on panoramic images has already been investigated in [4], where the network used was pre-trained on raw camera images. Consequently, it is not desirable for the objects in the panoramas to be deformed. For this reason, we run the same evaluation as in section 4.2 for the deformation of all object instances separately. In this case, an average displacement is determined for each object instance and not for each individual image. In the nuScenes dataset [12], the 2D bounding boxes are evaluated with the object classes *car*, *truck*, *bus*, *construction*, *cycle*, *trailer*, *pedestrian* and *cyclist*. In the Waymo Open Dataset for perception [10], we evaluate the panoptic labels with the classes *car*, *truck*, *bus*, *other large object*, *trailer*, *pedestrian*, *pedestrian object*, *bicycle*, *motorcycle*, *cyclist*, *motorcyclist*. The results are shown analogously as two-dimensional box plots for both evaluated datasets in Fig. 6 and 7. As in 4.2, a comparable reduction in image deformations due to gradual image warping can be recognized for the object instances.

5 Conclusion

In this article, we presented a method for improved image stitching using gradual image warping in autonomous driving. To achieve this, the images are warped in vertical sections to gradually compensate for the initial deformation caused by the estimated homography. In the evaluation, we were able to show successfully that the deformations in the panoramic images are significantly compensated for with our approach. We demonstrated this effect not only qualitatively but also quantitatively by evaluating deformations in 955 images from the nuScenes dataset [12] and the Waymo Open Dataset for perception [10]. Since our approach is primarily designed for improving object detection, we specifically measured deformations of object instances labeled in the data. Also in this case, gradual image warping shows clearly reduced image deformations. As a positive side effect, the image width of the resulting panoramas now remains constant. In upcoming research, we plan to investigate object detection capabilities on panoramic images created with gradual image warping.

References

1. T. Woopen *et al.*, "UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts," in *27. Aachen Colloquium Automobile and Engine Technology*, 2018, pp. 663–694.
2. M. Buchholz *et al.*, "Automation of the UNICARagil vehicles," in *29th Aachen Colloquium Sustainable Mobility*, 2020, pp. 1531–1560.
3. C. Kinzig *et al.*, "Real-time seamless image stitching in autonomous driving," in *2022 25th International Conference on Information Fusion (FUSION)*, 2022, pp. 1–8.
4. ——, "Panoptic segmentation from stitched panoramic view for automated driving," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 3342–3347.
5. T.-Z. Xiang, G.-S. Xia, and L. Zhang, "Image stitching with perspective-preserving warping," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III-3, pp. 287–294, 2016.
6. J. Zaragoza *et al.*, "As-projective-as-possible image stitching with moving dlt," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2339–2346.
7. C.-H. Chang, Y. Sato, and Y.-Y. Chuang, "Shape-preserving half-projective warps for image stitching," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3254–3261.
8. Y.-S. Chen and Y.-Y. Chuang, "Natural image stitching with the global similarity prior," in *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 186–201.
9. Y. Zhang, Y.-K. Lai, and F.-L. Zhang, "Content-preserving image stitching with piecewise rectangular boundary constraints," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 7, pp. 3198–3212, 2021.
10. P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2443–2451.
11. H. Le *et al.*, "Deep homography estimation for dynamic scenes," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7649–7658.
12. H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.

Bildverarbeitung ist definitionsgemäß die Wissenschaft von der Verarbeitung von Bildern. Damit verknüpft das Fachgebiet die Sensorik von Kameras – bildgebender Sensorik – mit der Verarbeitung der aufgenommenen Sensordaten – den Bildern. Aus dieser Verknüpfung resultiert der besondere Reiz dieser Disziplin. Bildern begegnet der Mensch ständig, schon weil das Sehen die wichtigste Informationsquelle als Handlungsgrundlage für den Menschen bildet.

Der vorliegende Tagungsband des „Forums Bildverarbeitung", das am 21. und 22. November 2024 in Karlsruhe als gemeinsame Veranstaltung des Instituts für Industrielle Informationstechnik am KIT und des Fraunhofer-Instituts für Optronik, Systemtechnik und Bildauswertung stattfand, enthält die schriftlichen Aufsätze der eingegangenen Beiträge. Darin wird über aktuelle Trends und Lösungen der Bildverarbeitung in den methodischen Schwerpunkten Messtechnische Anwendungen, Robotik, Bildgewinnung, Bildverarbeitung, Unsicherheiten bei maschinellem Lernen, Wahrnehmung von Personen, Künstliche Intelligenz als Mess- und Prüfmittel, Fahrzeuge berichtet.

ISSN 2510-7224
ISBN 978-3-7315-1386-5

Gedruckt auf FSC-zertifiziertem Papier