Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 253 (2025) 1256–1265

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Conference on Industry 4.0 and Smart Manufacturing

Automatic Gear Tooth Alignment 2.0: Improved Image Segmentation for Better Rotation Angle Deviation Determination

Florian Grimm[a,*], Daniel Kiefer[a], Tim Straub[a], Günter Bitsch[a], Clemens van Dinther[b]

[a]Reutlingen University, Alteburgstr. 150, 72762 Reutlingen, Germany
[b]Karlsruhe Institute of Technology, Kaiserstr. 89-93, 76133 Karlsruhe, Germany

Abstract

The maintenance of special tools is an expensive business. Either manual inspection by an expert costs valuable resources, or the loss of a tool due to irreparable wear is associated with high replacement costs, while reconditioning requires only a fraction. In order to avoid higher costs and drive forward the automation process in production, a German gear manufacturer wants to create an automatic evaluation of skiving gears. As a sub-step of this automated condition detection, it is necessary for wheels to be automatically aligned within a vision-based inspection cell. In extension to a study conducted last year [1], further image preprocessing steps are implemented in this publication and a new alignment algorithm from the autoencoder family is evaluated. By using an additional synthetic dataset, previous limitations could be clarified. The results show that thorough data preparation is beneficial for all solution approaches and that neural networks can even beat a brute force algorithm.

Keywords: image regression; segmentation; autoencoder; gear tooth alignment; deep learning;

This work is part of "MetaLearn" and has been carried out with the support of the German Federal Ministry of Education and Research (BMBF)

1. Introduction

The use of artificial intelligence (AI) in industrial environments has risen sharply in recent years [2]. More and more companies are integrating machine learning algorithms into their daily workflows, whether to automate repetitive tasks, simplify existing jobs or optimize business processes. The range of applications is manifold [3]. For example, quality checks of manufactured products are carried out using visual inspections and classification networks to determine whether a product is OK or not [4, 5]. Successful approaches in the field of order scheduling within pro-

* Corresponding author. Tel.: +49 (0) 7121 271 1498
E-mail address: florian.grimm@reutlingen-university.de

1877-0509
10.1016/j.procs.2025.01.187

duction [6, 7] or the use of AI in the area of process optimization, such as optimized parameter determination when setting up machines or the identification of the most suitable workstation based on historical data [8] have also been made. Overall, it is clear that AI is an elementary component of modern industry in view of the rapid developments [9] and that the application of machine algorithms to company data offers enormous potential [10].

In some cases, the use of AI is being considered to solve tasks that are either expensive or impossible to solve by mechanical means. One example is determining whether used tools can continue to be used in production or need to be replaced. Companies that use highly specialized and expensive equipment have a particular interest in the timely detection of tools that are too worn, as reconditioning is often cheaper than replacing them [11]. This study is about supporting a German gear manufacturer that uses the gear hobbing process in production. An automatic condition determination system is to be implemented so that unusable tools can be detected, replaced and repaired in good time in the future. This publication is a result of the cooperation project with this company.

1.1. Problem Description

As extensively described in [1] and [12], the company faces the problem of a reliable automatic classification, whether a hobbing wheel still can be used in production or better should be reconditioned in order to prevent a total loss of the tool. As possible solution to the problem, a vision-based control cell was constructed in order to capture multiple images of the hobbing wheel with different views on all individual teeth. After labeling the worn-out parts using polygons in label studio [13], a segmentation and classification network was trained to determine the varying degrees of abrasion in different regions of the hobbing wheel [12]. However, the possibility of an initial automatic placement of the tool for the image capturing process is still lacking of accuracy. In [1], different approaches, such as brute force, orb-feature detection and an image regression network are examined, to determine the deviation angle of a perfectly centered tooth and a recording section of an arbitrary placed tool. The results were surprising, as the image regression performed better than the brute force method. After a thorough analysis of the data, it was concluded that better preprocessing of the images could lead to better results, which leads to the first research question: ***How can the image preprocessing be improved to make all approaches more robust to distortions and impurities on the recorded hobbing wheel?*** During a literature review, an autoencoder approach called **u-net** has been found [14], that was specialized on segmentation of medical objects, having some ground truth binary masks as label for the training process. In comparison to the classical autoencoder implementation [15], the architecture features skip-connections, to improve the information flow between the encoder and the decoder networks. For tasks such as feature extraction or dimensionality reduction, these skip-connections are rather unpracticable, but it helps enormously during a segmentation task, as described later. However, in order to solve a regression task for the angle deviation determination, autoencoder can be used to create latent space representations of an image, that describe its properties and characteristics. Here, the second question arises: ***Are these feature vectors sufficient representative to be used as input for an actual regression task?*** Another limitation identified in [1] was the lack of knowledge about the true origin of the rotation, the center of the hobbing wheel. In order to identify the effects on the determination of the angular deviation, this time a synthetic dataset was additionally created where all parameters are known. All solution approaches were then applied to this dataset, which leads to the third research question: ***Can the previously examined limiting factors be specified more precisely when comparing the results of the real data with those of the synthetic data?***

1.2. Structure of this Study

This study is divided into three main sections. In Section 2: **Approach and Methodology**, a recapture of the last study [1], including a brief overview of the used dataset, the generated synthetic data and a detailed overview of all described methods and algorithms is given. In Section 3: **Results and Discussion**, all training and prediction results are listed, accompanied with performance plots and an analysis of the outcome. At the end, a **Conclusion and Outlook** is drawn in Section 4, where all findings are briefly summarized, limitations are discussed and further investigation goals are outlined.

The study contributes to the research fields of image processing and the application of artificial intelligence methods within industry. It tries to solve the task of automatic aligned gear tools inside a vision-based control cell, so that further quality assurance algorithms can be applied.


(a) (b) (c)

Fig. 1: Examples segmenting teeth with heavy impurities. On (a), the input image can be seen. (b) shows the results of the scikit-image and OpenCV combination. On (c), the u-net segmentation is illustrated.

2. Approach and Methodology

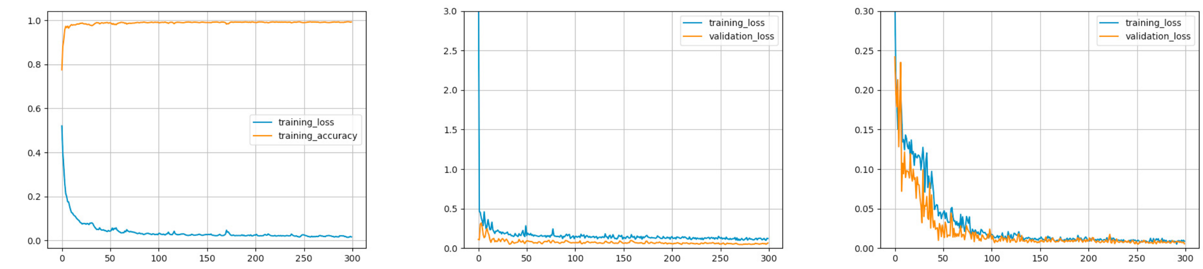
In this section, a brief recapture of the previously collected real-world dataset, the newly generated synthetic dataset, the environment settings and the company's workflow is given, followed by a description of the improved image preprocessing procedures of the collected dataset. Afterwards, an overview of the examined approaches is provided, with focus on the newly investigated autoencoder regression algorithm.

2.1. Recapture of Dataset, Setting and Workflow

In [1], a visual-based control cell has been introduced, which is able to capture high resolution (2448 × 2048) gray-scale images of hobbing wheel teeth using a camera that is mounted on a robot-arm. A turn-table, where the tool is placed, rotates counter-clockwise by 360° divided by the amount of teeth of the tool. In the current setting, hobbing wheels with in total 38 teeth are observed, leading to an angle between two teeth equal to approximately 9.47°. During the calibration process, 256 images were captured, that are available for this study. The workflow within the company is as follows: A worker dismounts a hobbing wheel from its current machine and places it in the control cell. After capturing the different views [12], a recommendation is sent to the corresponding employee, whether the tool is still usable or should be reconditioned. In order to get consistently oriented images that are in focus, the placement of the hobbing wheel in the control cell has to be as accurate as possible. Therefore, in this study, the automatic placement of tools is investigated.

2.2. Synthetic Dataset Creation

In addition to the captured dataset, a synthetic dataset is generated in order to identify the influence of inaccuracies in the determination of the rotational origin. As the authors had the taken images of the visual-based control cell at their disposal, but not the size of the hobbing wheel or the precision of the turn-table, a brute force algorithm was used to determine the origin in the last paper [1]. For this purpose, two consecutive images of teeth were taken and the origin of rotation was determined using brute force and the mean-squared-error via a set of gear centers (please refer to [1]). As this resulted in slightly different pixel values for the distance of the tooth to the center of the hobbing gear, the median of all calculated distances was used to minimize the influence of outliers. Either way, there was some inaccuracy and a thorough calibration is still pending. With a synthetic dataset this inaccuracy can be eliminated, as the origin of rotation is always exactly the same. This allows a theoretical best value of all angle determination approaches and the potential for further preprocessing to be identified. The dataset was generated by creating a gear drawing using a gear generator [16]. The resulting graphic was correctly aligned and rotated two full rotations with a step-size of 3.65° each around the center of the image and a third rotation with 6.00° per step, analogous to the real dataset, resulting in the same amount of images (256 in total). A 2048 × 2048 section was saved analogous to the camera recording after each rotation. A comparison of the masks of the synthetic and the real-world dataset can be seen in Figure A.4.



(a) (b) (c)

Fig. 2: Loss curves of the real-world dataset training processes. (a) Training of the u-net (b) Training of the feature vector regression. (c) Training of the image regression network.

2.3. *Preprocessing: Image Segmentation for Tooth Detection*

In the first approach of determining the angle deviation, the preprocessing steps were limited to a normalization of the image and median-filtering for minor distortions on the tooth. For heavier distortions, a segmentation approach was investigated, using the MASK-RCNN from [17]. Since the results of the created masks were good but slightly unsharp and welly (see Figure D.7), a special focus is set on other possibilities of image segmentation and mask creation.

2.3.1. *OpenCV and Scikit-Image Image Segmentation*

A straight-forward solution of binary masking in the field of image processing is finding contours within an image and filling the encircled area. All necessary methods are provided by two very common image processing libraries: OpenCV [18] and scikit-image [19]. As first step, a per image threshold is calculated via Otsu's method [20] of scikit-image. Then, OpenCV's findContours algorithm is applied to find the contours of the hobbing wheel's teeth. To prevent errors on the image edges, the top, left and right image edge's color is set to white. Thereby, distortions on the upper edge are also taken into account. Afterwards, all hierarchical children of the contours are filled with white color. With very few exceptions (see Figure 1), this approach produces precise and sharp binary masks for each tooth (see Figure D.8). Using a normal threshold procedure, this would have been not possible, since the distortions in the middle area of the tooth would have been converted to black instead of white.

2.3.2. *U-NET Image Segmentation*

In the field of artificial neural networks, the principle of autoencoders [15] can be applied to solve various tasks, such as anomaly detection, denoising of images or feature extraction. In 2015, Ronneberger et al. [14] introduced an adapted autoencoder network, using skip-connections between the encoder and decoder section to improve the information flow between the individual stages. The result was an architecture of a robust network, that reliably was able to segment different cells on medical images. This architecture has been chosen to perform a tooth segmentation in the training image dataset. The training has been performed over 300 epochs, using adam as optimizer function and binary crossentropy as loss function (see Figure 2a). For the architecture, please refer to [14].

2.4. *Angle Determination*

The main goal of this study is the correct angle deviation determination between a reference image of a perfectly aligned tooth in the center of an image, and an arbitrary position of the hobbing wheel, using the same camera position as for the reference image. To achieve this, several algorithms are implemented and their performance compared.

2.4.1. *Brute Force Baseline*

As baseline, a brute force algorithm is chosen. Compared to the previous implementation [1], the step-size has been set equal to 0.01° to gain a more fine granular result. This yields in larger runtimes, but with the improved image

Table 1: Result table with the absolute errors in degree per approach, dataset and fold. Lowest errors are marked as bold.

		Real-world Dataset			Synthetic Dataset		
Metric	Fold	Brute Force	Image Regression	Autoencoder Regression	Brute Force	Image Regression	Autoencoder Regression
mean	1	0.69	**0.68**	0.94	**0.2**	**0.2**	0.58
median	1	0.32	**0.3**	0.55	**0.0**	0.15	0.48
std dev	1	1.7	1.5	**1.32**	1.28	**0.16**	0.6
mean	2	0.86	**0.56**	0.96	**0.01**	0.18	0.56
median	2	0.36	**0.28**	0.57	**0.0**	0.16	0.66
std dev	2	2.03	**1.23**	1.47	**0.03**	0.15	0.28
mean	3	0.86	**0.58**	0.94	**0.03**	0.29	0.55
median	3	0.43	**0.27**	0.64	**0.0**	0.09	0.42
std dev	3	2.04	1.53	**1.37**	**0.07**	1.13	0.96
mean	4	1.04	**0.44**	0.77	**0.19**	0.43	0.54
median	4	0.49	**0.33**	0.34	**0.0**	0.14	0.24
std dev	4	2.29	**0.44**	1.21	1.29	1.45	**1.06**
mean	5	0.68	**0.6**	0.72	**0.02**	0.13	1.2
median	5	0.34	0.35	**0.31**	**0.0**	0.09	1.04
std dev	5	1.68	1.3	**1.1**	**0.06**	0.13	0.86

segmentation and the use of the binary masks as input instead of the captured images, a greater accuracy regarding the rotation angle determination is assumed.

2.4.2. Image Regression

As second algorithm, the image regression network introduced by [21] is retrained using the created binary masks of the hobbing wheel teeth. The network is trained with the same settings as in the last investigation, using adam as optimizer, the mean-squared-error as loss function and 300 epochs for the fitting process (see Figure 2c). The only adaption is that no resizing of the images has been performed before feeding the network.

2.4.3. Autoencoder Feature Regression

In extension to [1], a new approach for the angle determination is investigated. Since the u-net described in section 2.3.2 was able to generate adequate feature representations of the teeth's binary masks, it came the idea of using these features as input for a simple regression network. For this, the u-net architecture has been modified, adding two additional convolution blocks and leaving out the skip-connections to generate a classical autoencoder network. The results showed, that due to the relatively simple shape of the hobbing wheel teeth, the skip-connections were not essential for learning a sufficient representation of a binary mask. The network has been trained with the same settings as described in section 2.3.2. The subsequent regression network consists of four dense layers, followed by dropout layers for improving the robustness of the architecture and preventing it from overfitting (see Figure C.6). It uses adam as optimizer and the mean-squared-error loss as well as 300 epochs for model fitting (see Figure 2b).

3. Results and Discussion

In this section, the results of the different angle deviation determination methods are described. All computations are performed on a system with a i9-10900X as CPU, a NVIDIA RTX A4000 as GPU, and 128GB of RAM. The code has been implemented in Python, using Tensorflow [22] as deep learning framework and OpenCV [18] and scikit-image [19] as image processing libraries.

Concerning the outcome of the image segmentation procedure, it can be said that both work very well with varying limitations. The image processing approach using scikit-image and OpenCV produces sharp and clean high resolution binary masks. Only some heavy impurities can't be captured entirely. The u-net approach is able to cover these

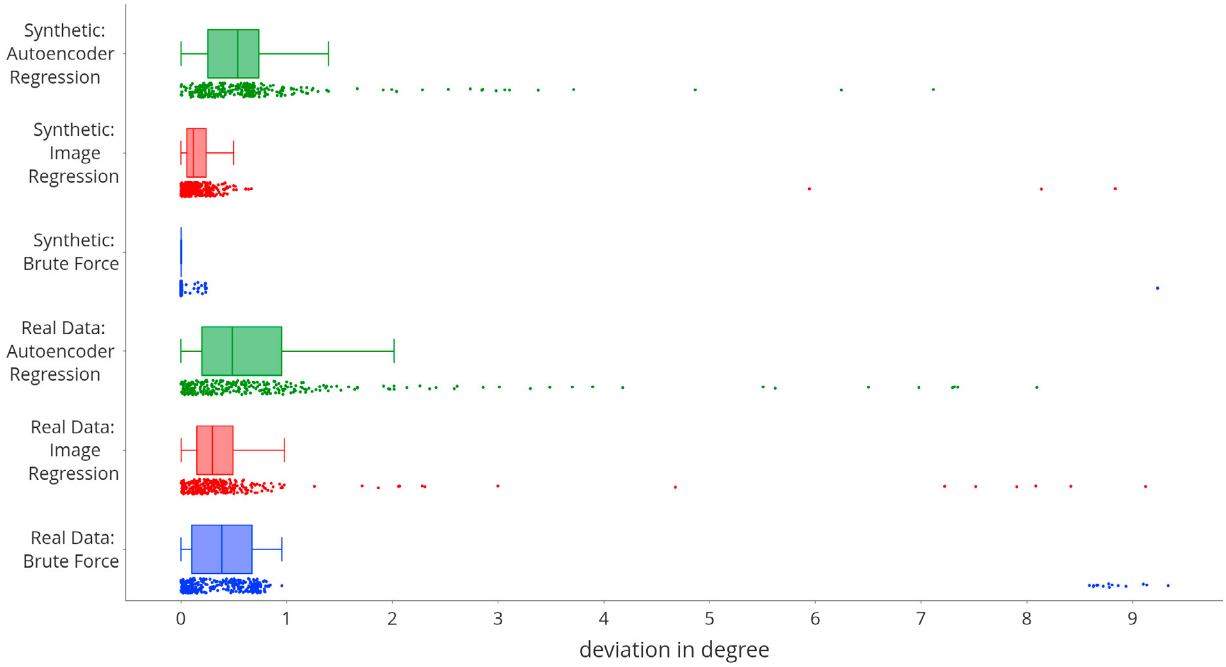


Fig. 3: Box-Plots of the absolute errors in degree for each approach and dataset.

impurities at least up to a certain degree, as depicted in Figure 1c, but it's outcome is currently limited to low-resolution images of 256 × 256 pixels. Compared to the used MASK-RCNN [17] approach used last time [1], both algorithms are superior in terms of accuracy of the recognized tooth shape (see Appendix D). Due to the resolution, the binary masks of the image processing segmentation procedure described in section 2.3.1 are used for all angle deviation determination approaches.

In order to make reliable statements about the results of the neural network training, a cross-validation with a k-fold of 5 was performed. In Table 1, the results of the folds over all approaches and datasets are listed. Apart from four exceptions, the best performing algorithm for the real-world dataset is the image regression network introduced by [21], beating again the brute force baseline as observed in the last investigation [1]. Only the autoencoder shows in three folds a more consistent prediction, resulting in a slightly smaller standard deviation, but when looking at the mean and median absolute error, it performs worse except for once. Looking at the results of the synthetic dataset, it is clear that the correct rotation around the known center of the hobbing wheel brings an enormous improvement. With the exception of two folds, the brute force algorithm is by far the most accurate. Surprisingly, however, all algorithms improve noticeably compared to the real dataset. Since the main difference between the two datasets lies in the different size of the teeth, it can be assumed that the higher number of pixels per tooth enables a more robust learning of the relevant features and thus a more precise determination of the angular deviation. A closer look at Fig. 3 shows that all algorithms suffer from extreme outliers. The brute force algorithm in particular has several predictions above 8°, explaining the high standard deviation observed before. These outliers represent the class of images, where two teeth are captured on one image (see B.5) and the algorithm calculates a lower mean-squared-error on the wrong tooth of the image. For the neural network approaches, the deviation is smaller but with regard of the maximal deviation possible (please refer to 2.1), it is still relatively high.

In order to draw a comparison between all presented methods, the predicted angular deviations were evaluated across all images and folds. The results are listed in Table 2 and it can be seen, that with an error mean of 0.57°, a median of 0.3° and a standard deviation of 1.25°, the image regression network performs best for the real-world dataset. Although the image preprocessing reduced the errors for the brute force method compared to the previous

Table 2: Result table with the overall absolute errors in degree per approach and dataset. Lowest errors are marked as bold. Per approach, the needed training time is appended.

	Real-world Dataset			Synthetic Dataset		
Metric	Brute Force	Image Regression	Autoencoder Regression	Brute Force	Image Regression	Autoencoder Regression
mean	0.83	**0.57**	0.87	**0.09**	0.25	0.69
median	0.39	**0.3**	0.49	**0.0**	0.12	0.54
std dev	1.95	**1.25**	1.29	**0.81**	0.83	0.84
times [m]	579.73	7.39	31.46	477.29	7.33	31.23

study [1], the standard deviation has increased. This is due to the problem that in some images two teeth are visible. As far as synthetic data is concerned, the brute force algorithm is by far the best.

Looking at the results from an industry perspective, it can be said that the errors for the real data are still relatively high. A deviation of 0.57° in average is most likely producing images, that are out of focus for the side views. Unfortunately, no general threshold value can be specified to determine how good the result is, as an acceptable threshold value depends on the underlying setting (e.g. depth of field of the camera). For sharply focused images of each individual tooth, further investigation and work on the problem is required in this case, as the control cell described is equipped with a camera whose focus range is very narrow. However, advances in image segmentation for the preprocessing are achieved, showing that not always AI is needed or superior in solving a task, as it can be seen in the creation process of the binary masks. In addition to that, the accuracy for the alignment process of gear teeth has been improved compared the previous study [1], confirming that good image processing is the key to many algorithms and good results.

4. Conclusion and Outlook

Automatic gear tool alignment remains a challenging task within industry. Beside hardware solutions using laser distance measurements for gear tooth detection, programmatic approaches represent a viable alternative for solving the problem. In this paper, further investigations on the determination of angle deviation between two rotated images are performed. With a centered tooth as ground truth, different approaches are tested to gather the correct rotation angle that has to be sent to the turn-table, in order to gain a perfectly oriented starting point for the following image capturing process. In extension to [1], a special focus is put on the preprocessing of the training image dataset. For this, two different segmentation algorithms are compared, of which one uses image properties and classical image processing methods, and the other one an autoencoder variation featuring skip connections for a better information flow. The task for the determination of the angle deviation is performed comparing the brute force algorithm and the image regression network of the last study [1] with a new variation of the u-net, leaving out the skip connections and taking the reduced feature space of the encoder as input for a regression network. The results show, that all algorithms benefit from the extended image preprocessing procedure, reducing the average of absolute error by 0.37°, the median by 0.18° and the standard deviation by 0.22° when comparing the best performing algorithm of this study and the last investigation [1]. Regarding the performance, the image regression network introduced by [21] again beats all other examined approaches (see Table 2). With a mean of absolute error of 0.57°, a median of 0.3° and a standard deviation of 1.25°, it prevails over the autoencoder regression and brute force approach. Concerning the execution times, the image regression network needs with $7m39s$ the lowest training time, placing it once again on the top. By using a synthetic data set, the limitations of the previous study could be clarified and it was found that the rotational origin of the hobbing wheel is the key to the brute force algorithm.

Although the image preprocessing has improved the results by a lot, there are still some limitations that occurred during the studies. As mentioned in [1], the determination of the origin for the rotation matrix still needs improvement. This time, the point of origin was determined individually for each image, but the variance of the results indicates that the turntable needs to be recalibrated, since a small shift of the teeth was observed. In addition to that, some heavy distortions could not be masked correctly, as described in section 3 and Figure 1. Regarding the outcome of the angle deviation determination, there are still a couple of outliers, that could not be processed by the presented approaches. This is mainly due to the fact that in some cases there are two teeth on the same image and it is quite difficult for the algorithms to decide which one should be centered. Concerning the outlook, a network refinement of the hyperparameters as well as variations to the network architecture can be performed as extension to the presented approaches and results. As several new images have been taken since the last publication, the training process of the neural networks can be fed with an extended dataset, which is likely to improve the network's performance and robustness. An additional investigation of the examined approaches on other datasets might also give insights about the performance of similar tasks under different circumstances.

These recommendations will be tested shortly in future work.

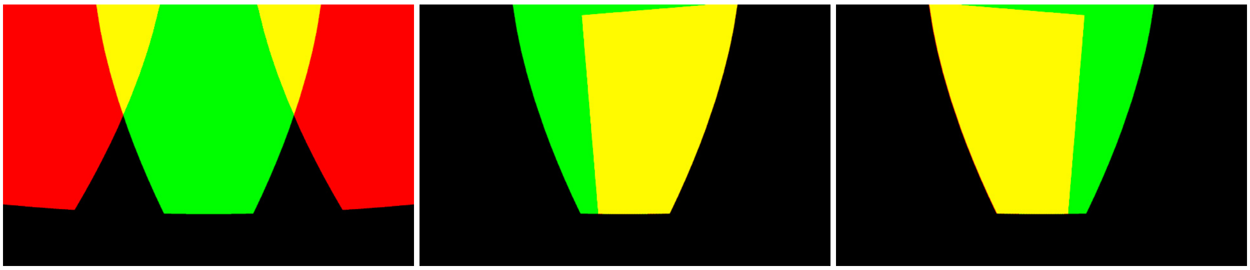
Appendix A. Dataset Comparison

(a) (b) (c)

Fig. A.4: Comparability between real-world dataset and synthetic dataset. On (a) the actual captured picture can be seen, on (b) its mask created by opencv and scikit-image and on (c) the synthetic correspondent of the same angle deviation is depicted. It can be seen, that the synthetic dataset features wider teeth, but has similar distances to its neighbor teeth.

Appendix B. Outlier Investigation

In the brute force algorithm, all outliers in both datasets, the synthetic and the real one, are above 9°. This is due to the case where two teeth lie almost symmetrically on the image and the distance between the two teeth and the ground truth is the same. The following graph illustrates this case:



(a) (b) (c)

Fig. B.5: Case where the maximal deviation of the ground-truth (green) is observed. On (a) the ground-truth and the actual investigated image (red) can be seen, on (b) the correct rotation for the left tooth is depicted and on (c) the correct rotation for the right tooth is illustrated. It can be seen that (b) and (c) are valid alignments, but within the labels, one would be a deviation of 0° and the other one the maximal deviation of 9.47°.

Appendix C. Regression Network for Encoded Images

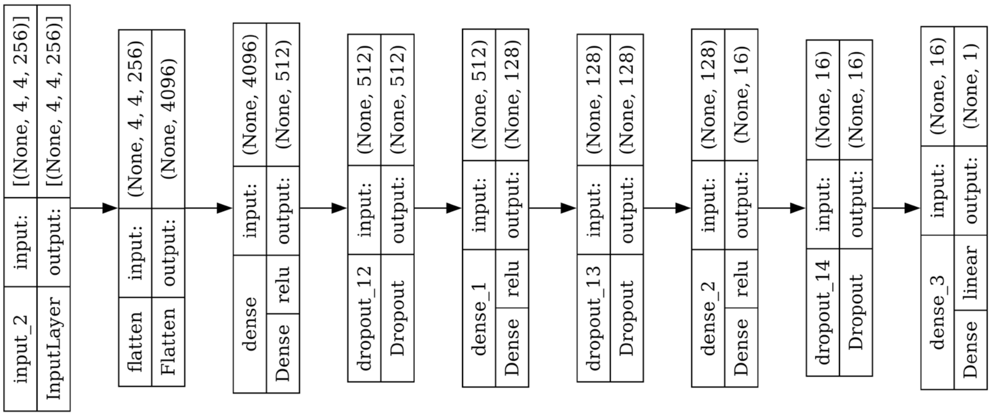
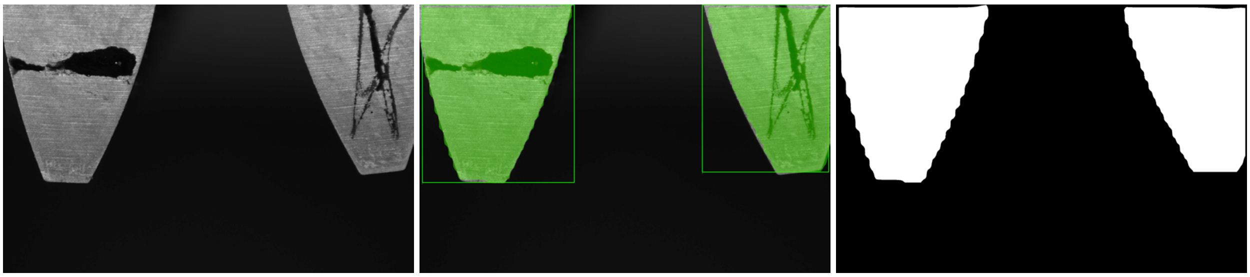


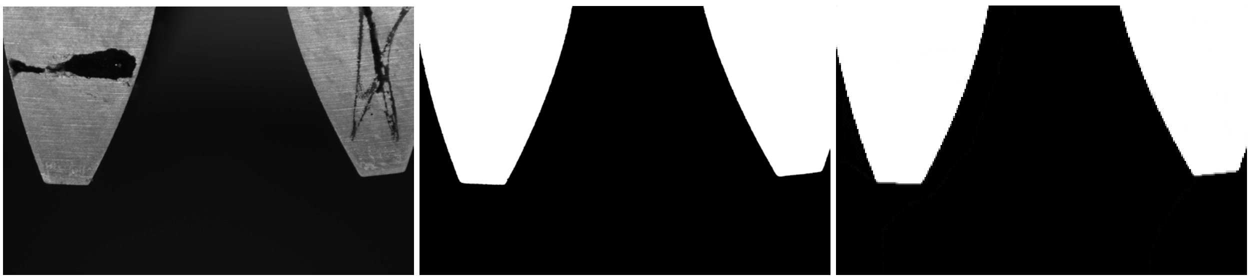
Fig. C.6: Network architecture of the encoded image feature regression.

Appendix D. Mask Segmentation Comparison



(a) (b) (c)

Fig. D.7: Process of the image segmentation using ”MASK-RCNN”. On (a), the input image can be seen. (b) shows the intermediate outcome of the network with the found bounding-boxes. On (c), the final processed results is illustrated.



(a) (b) (c)

Fig. D.8: Results of the improved image segmentation using scikit-image with OpenCV and an autoencoder approach. On (a), the input image can be seen. (b) shows the segmentation using scikit-image and OpenCV. On (c), the u-net segmentation is illustrated.

References

[1] F. Grimm, D. Kiefer, T. Straub, G. Bitsch, C. van Dinther, Automatic gear tooth alignment in vision based preventive maintenance, Procedia Computer Science 232 (2024) 1564–1572, 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023). doi:https://doi.org/10.1016/j.procs.2024.01.154.
URL https://www.sciencedirect.com/science/article/pii/S1877050924001546

[2] R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo, J. Barata, Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook, IEEE access 8 (2020) 220121–220139.

[3] I. K. Nti, A. F. Adekoya, B. A. Weyori, O. Nyarko-Boateng, Applications of artificial intelligence in engineering and manufacturing: a systematic review, Journal of Intelligent Manufacturing 33 (6) (2022) 1581–1601.

[4] J. Zipfel, F. Verworner, M. Fischer, U. Wieland, M. Kraus, P. Zschech, Anomaly detection for industrial quality assurance: A comparative evaluation of unsupervised deep learning models, Computers & Industrial Engineering 177 (2023) 109045.

[5] A. Arora, R. Gupta, A comparative study on application of artificial intelligence for quality assurance in manufacturing, in: 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, 2022, pp. 1200–1206.

[6] B. Waschneck, A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, A. Kyek, Optimization of global production scheduling with deep reinforcement learning, Procedia Cirp 72 (2018) 1264–1269.

[7] B. Waschneck, A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, A. Kyek, Deep reinforcement learning for semiconductor production scheduling, in: 2018 29th annual SEMI advanced semiconductor manufacturing conference (ASMC), IEEE, 2018, pp. 301–306.

[8] D. Kiefer, F. Grimm, T. Straub, G. Bitsch, C. van Dinther, A framework for explainable root cause analysis in manufacturing systems–explainable artificial intelligence for shopfloor workers (2024).

[9] P. Zheng, H. Wang, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarok, S. Yu, X. Xu, Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives, Frontiers of Mechanical Engineering 13 (2018) 137–150.

[10] J. F. Arinez, Q. Chang, R. X. Gao, C. Xu, J. Zhang, Artificial intelligence in advanced manufacturing: Current status and future outlook, Journal of Manufacturing Science and Engineering 142 (11) (2020) 110804.

[11] V. Shapovalov, A. Klochko, M. Gasanov, O. Antsyferova, A. Belovol, Optimizing the technology of reconditioning large high precision gear rims, Innovative Technologies and Scientific Solutions for Industries (1 (3)) (2018) 59–70.

[12] D. Kiefer, S. Wezel, A. Böttcher, F. Grimm, T. Straub, G. Bitsch, C. Van Dinther, Anomaly detection in hobbing tool images: Using an unsupervised deep learning approach in manufacturing industry, Procedia Computer Science 232 (2024) 2396–2405, 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023). doi:https://doi.org/10.1016/j.procs.2024.02.058.
URL https://www.sciencedirect.com/science/article/pii/S1877050924002357

[13] K. Wada, Labelme: Image Polygonal Annotation with Python. doi:10.5281/zenodo.5711226.
URL https://github.com/wkentaro/labelme

[14] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, 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.

[15] M. A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, AIChE journal 37 (2) (1991) 233–243.

[16] E. Design, Spur Gear Generator.
URL https://evolventdesign.com/pages/spur-gear-generator

[17] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[18] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools (2000).

[19] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, the scikit-image contributors, scikit-image: image processing in Python, PeerJ 2 (2014) e453. doi:10.7717/peerj.453.
URL https://doi.org/10.7717/peerj.453

[20] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man, and Cybernetics 9 (1) (1979) 62–66. doi:10.1109/TSMC.1979.4310076.

[21] S. Verma, Angle-Detector.
URL https://github.com/shivaverma/Angle-Detector

[22] 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, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL https://www.tensorflow.org/