

DETECTION AND CLASSIFICATION OF SYMBOLS IN PRINCIPLE SKETCHES USING DEEP LEARNING

Bickel, Sebastian;
Schleich, Benjamin;
Wartzack, Sandro

Friedrich-Alexander-Universität Erlangen-Nürnberg

ABSTRACT

Data-driven methods from the field of Artificial Intelligence or Machine Learning are increasingly applied in mechanical engineering. This refers to the development of digital engineering in recent years, which aims to bring these methods into practice in order to realize cost and time savings. However, a necessary step towards the implementation of such methods is the utilization of existing data. This problem is essential because the mere availability of data does not automatically imply data usability. Therefore, this paper presents a method to automatically recognize symbols from principle sketches, which allows the generation of training data for machine learning algorithms. In this approach, the symbols are created randomly and their illustration varies with each generation. . A deep learning network from the field of computer vision is used to test the generated data set and thus to recognize symbols on principle sketches. This type of drawing is especially interesting because the cost-saving potential is very high due to the application in the early phases of the product development process.

Keywords: Artificial intelligence, Machine learning, Early design phases

Contact:

Bickel, Sebastian
Friedrich-Alexander-Universität Erlangen-Nürnberg
Engineering Design
Germany
bickel@mfk.fau.de

Cite this article: Bickel, S., Schleich, B., Wartzack, S. (2021) 'Detection and Classification of Symbols in Principle Sketches Using Deep Learning', in *Proceedings of the International Conference on Engineering Design (ICED21)*, Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.118

1 INTRODUCTION

Symbols are very common way of representing information in our daily life. No matter if it is a traffic sign, an icon in a program or a symbol on a PC case, all symbols have the same goal. However, they require semantic and domain-specific knowledge of the user. The type, shape, size, and complexity of symbols strongly depends on the context and the degree of information transfer. For example, the specific symbols in technical drawings are only understandable for an educated engineer. A different person cannot understand the additional information provided by the symbols about tolerances, surface finishes or centerlines and therefore cannot fully comprehend the drawing. The reading and understanding of symbols on technical drawings has similarities with a language that has to be learned. Thus, there is no doubt that there is a high potential for the utilization of data in the evaluation of single symbols and especially in diagrams or graphs. For better exploitation of these drawings, the symbols must be recognized automatically to make a subsequent analysis. For a data-driven economy, this step is essential in order to be able to utilize existing data. The motivation for this paper is to assist the whole product development process through the automatic detection of symbols. In order to exploit the best potential in the product design process, support should be available as early as possible. This is due to the high cost responsibility of the early phases, which in turn offers enormous savings opportunities.

Therefore, the idea is applied to principle sketches. These are one of the first representations of conceptual products and should be automatically recognized and reused. This offers the opportunity to search in old drawings for a specific combination of symbols or to conduct a plausibility check in newly drawn principle sketches. Furthermore, the detected symbols can be converted into graphs and used for various applications, such as similarity search or graph-based tolerance specification (Goetz et al, 2018). This can provide essential advantages in the early phases of the development process, because it can link the product concept to already designed parts.

The first necessary step, however, is the automatic recognition of symbols in principle sketches. A recognition algorithm still requires a dataset to train the model. Especially for the detection of specific symbols or objects, the established benchmark datasets, which are commonly used for training, reach their limitations. To solve the data generation problem and to recognize symbols, an approach is presented in the following.

2 STATE OF THE ART

Nearly every engineering discipline has an own way of representing important information and knowledge through graphs, diagrams, or sketches. In Figure 1, a short overview of different sketches sorted by their engineering department is shown. These drawings and sketches can differ in format, size, complexity, and in their way of generation. The qualitative range of created drawings in mechanical engineering can vary between a quick hand drawing of a principle sketch up to a large technical drawing derived semi-automatically from the 3D-CAD model.

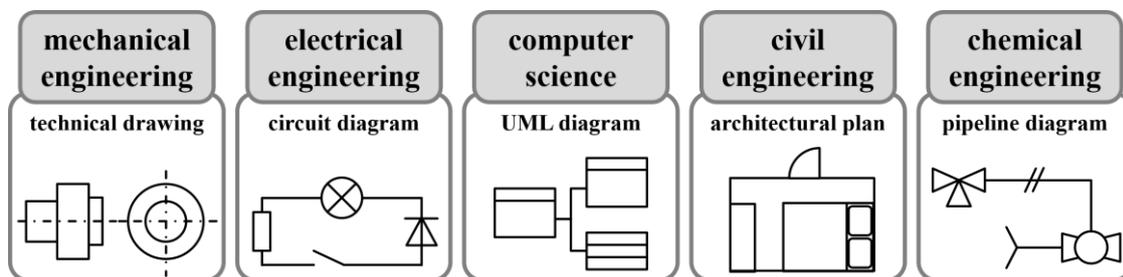


Figure 1: Overview of different sketches in engineering disciplines with examples

2.1 Product development process

In engineering design research, a variety of design processes have been developed and stated over the last decades. In general, these models are intended to support product development by providing guidance as to which steps are useful at what point in time and which methods can help to achieve the goal. Very common models are Pahl/ Beitz, Munich procedural model (Lindemann, 2008) and the VDI 2221/23.

According to VDI 2221, at first, solutions for the partial-functions are created under consideration of e.g. physical, chemical or electrical effects. Then, these effects are translated into effective principles, which form the basis for the combination into an overall solution. The result of this sequence is a set of principle solutions, which ensure that the functions of the product are fulfilled.

The developed solutions are then examined and evaluated in the fourth process phase. A comparison with the requirements assists in the analysis. After performing the review, the result of the process step is a selected solution concept in the form of a principle solution. (VDI 2221, 2019)

2.1.1 Principle solutions and principle sketches

In the context of technical products, a principle solution describes a rough and basic solution for a design task. The necessary effects for fulfilling the function are taken into account based on physical effects or the arrangement of bodies. The detailing of the principle solution then takes place in the embodiment design phase. Principle solutions can be provided for partial functions as well as for the overall function. They are usually documented in a principle sketch, which is a specific class of a technical drawing. This type of documentation is often used for basic innovations or new designs. (Roth, 2001)

According to VDI 2222 (1997) there are a multitude of representation modes for principle solutions, e.g. largely standardized symbolic sketches, free line sketches as abstraction of real structures, 3D freehand sketches, unscaled rough drafts or changes in existing drawings (with and without CAD).

In mechanical engineering, principle sketches are often used to illustrate principle solutions. In Figure 2 an example of a principle sketch is shown. This Figure also depicts the difference between a principal sketch and a technical drawing. The sketch illustrates a mechanism, which translates translational movement into rotation. The technical drawing shows the detailed design of the component.

It consists of a piston (c) and a matching housing, which serves as a guidance for the piston (d). The piston is connected to the bearing (b) via a connecting rod (a). Beside the principle sketch, the finished connecting rod is displayed. This comparison shows how much the components are simplified in principle sketches. However, they also offer great advantages through the simple representation of interrelationships, which in turn enables the rapid creation of variants and different solutions. To make the sketches usable in a data-driven development, they must be recognized using methods from Computer Vision to automatically capture the symbols and therefore the product-related information in the sketches.

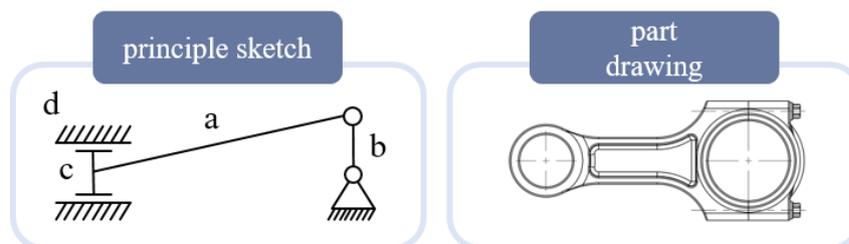


Figure 2: Example of a principle sketch and the resulting technical drawing of a designed component according to (Roth, 2000)

2.2 Computer vision

The research area of computer vision is interdisciplinary and follows the goal of training machines to extract information from images or videos. The technologies for computer vision have received a big push in recent years, based on more available computing power, higher image and video resolution, the application and development of Convolutional Neural Networks (CNN) and the increasing efforts in autonomous driving.

The inspiration for this field of research is the human vision, which the computer tries to imitate. It therefore includes methods and algorithms from other disciplines, such as image processing, pattern recognition, artificial intelligence, machine learning and computer graphics (Wiley and Luca, 2018). This also explains the wide range of applications for computer vision, for example: automation of assembly lines, robotics or remote sensing.

Due to the high level of interdisciplinarity and the wide range of applications for computer vision, there are different ways of structuring the methods. According to popular contests, such as

Everingham et al (2015), Russakovsky et al (2014), and Sinha et al (2017), the task spectrum of computer vision can be distinguished into three areas: classification, detection and segmentation.

Classification

The goal of classification is to assign an image to defined classes. These must be determined before the examination, which indicates it as a supervised learning procedure. Depending on the number of classes, different probabilities are calculated for each class. The label with the highest score is then assigned to the picture. This class label describes the main content of the image, e.g. dog, car or building. The location of the specific item in the picture is not retrieved. (Sinha et al 2017)

Object detection

Object detection is closely related to classification but is able to detect and highlight the position of objects in the images via bounding boxes. In contrast to classification, multiple bounding boxes can be allocated per image, which enables to extract more semantic information from the pictures. However in this case, the data generation is more complex because the position of the bounding box must be defined as input in addition to a label.

Segmentation

In comparison to the two tasks presented before, segmentation is a classification on a pixel level. By this exact assignment of each pixel to a class or background, the outlines of different objects in an image can be recognized and assigned. This ability eliminates the disadvantages of the bounding box, which is always a rectangle and cannot display the exact outline. This fine-grained resolution naturally requires more computational power and better-prepared input data. But object recognition can be used as a preliminary to segmentation (Leibe et al. 2004).

2.3 Engineering symbol recognition by computer vision

The idea of transforming the information automatically from technical drawings or graphs into a general readable format has been existing for over 30 years. For example in the late 1980's Okazaki et al. (1988) stated a new method to detect symbols in circuit diagrams through loop-structure based symbol recognition. Another example is a system for interpreting line drawings, which was developed by Kasturi et al (1990). These systems have in common that their methods and algorithms are heuristic-based and therefore hard to adapt to new drawings or symbols.

This changed with the upcoming trend of neural networks in the beginning of the twenty-first century and the special form of convolutional neural networks (CNN). CNNs are an evolution of neural networks that tries to imitate the human eye. The trend to utilize CNNs for the classification of images has been transferred to the field of symbol recognition. Denga et al. (2020) published a new method for classifying tolerance symbols in technical drawings. This framework applies several steps, including the image pre-processing, the callout location and extraction, followed by the symbol and character segmentation and in the final process step the symbol recognition.

Elyan et al. (2018) developed another solution for piping and instrumentation diagrams. Beside CNNs, two other machine learning classification algorithms were implemented. A random forest classifier and a support vector machine were selected and all three methods showed good overall classification accuracy. In contrast to the above-mentioned procedures, Fu and Kara (2010) created a method for detecting hand-sketch and computer generated diagrams. The user produces the training data, which is then synthetically enlarged. They were able to show that hand sketches could be transformed to Simulink or SimMechanics models.

In summary, a large problem with the development of methods for symbol recognition is the generation of the data. Either these problems are covered by provided datasets or using benchmark datasets. However, benchmark datasets are not specifically designed for the field of mechanical engineering and are therefore not well usable. Therefore, the symbols contained are rather general and not suitable for explicit design applications.

In the case of data provided e.g. by companies, a different problem is predominant. Here the labelling of the data is partly very complex and time-consuming. This is because the drawings must be searched for symbols manually and then labelled with bounding boxes. In addition, there is an extensive data preparation necessary, caused by possibly bad scan quality or errors in the creation of the drawings. Furthermore, these datasets are often subject to a non-disclosure agreement and are not open source.

The goal of this paper is to present a method that independently generates data for training a CNN that is able to detect objects, which are the symbols in principle sketches.

3 GENERAL APPROACH

The approach presented in this paper has the objective of automatically recognising symbols in principle sketches. In contrast to the methods explained in chapter 2, this model uses self-generated, synthetic data and the latest R-CNN version to train the recognition algorithm. As a result of the defined representation limits of principle sketches, shown in chapter 2.1.1, it is possible to train with synthetic data. In the following chapters a general overview will be given, followed by the data generation and training of the model. Finally, the results of the trained model are presented using the training data and sample datasets.

3.1 Overview

Figure 3 demonstrates the procedure of the symbol detection. It starts with a principle sketch as an input and results in bounding boxes fitted to the image. These bounding boxes can then be transformed to a graphed-based representation, which is the basis for future applications.

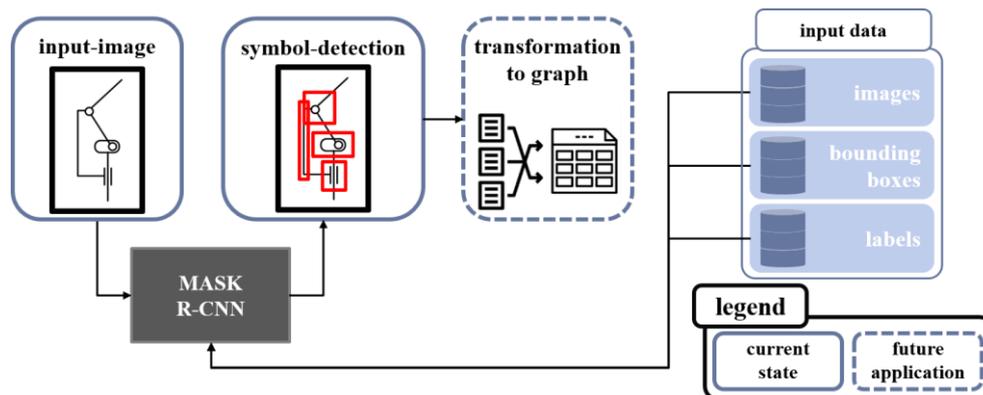


Figure 3: Overview of the symbol detection procedure

However, the first step is to detect the according symbols in a sketch or image. The classification of the different objects is achieved with the MASK R-CNN algorithm. The recognition requires the provision of training data beforehand. There are two issues with the data: First, a sufficient collection of drawings must either be already available, or collected and prepared. The second problem is the necessary labels and bounding boxes for the detection algorithm. Even if a sufficient amount of data would be available, bounding boxes must be drawn manually and be labelled afterwards. This procedure cannot be executed realistically, therefore a method was developed to solve both problems, which is explained in the section below.

3.2 Data generation

The data generation process is roughly divided into four steps. The general procedure is depicted in Figure 4. First a collection of symbols has to be created, which can be adjusted by specific factors. Potential parameters are the size, line width, individual length dimensions or the type of hatching. These values are randomly generated, but for every specific symbol, different rules apply to assure, that the general form of the symbol is still intact. Consequently, according to general parameters, different aspect ratios are defined. The ratios can also randomly changed, but only in a predefined margin.

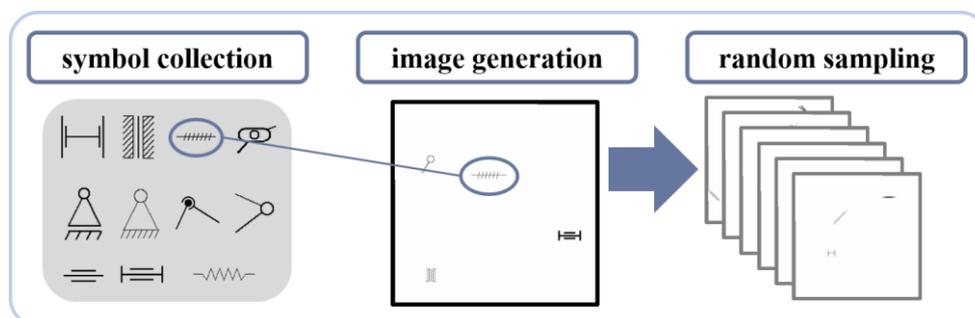


Figure 4: procedure for the data generation

In addition to the visualization of the symbols, a bounding box and a suitable associated label must also be produced during generation. The symbols currently drawn by the method include: fixed bearing, floating bearing, piston, linear guide, spring, joint, ball joint, double joint, rotating pair, floating rotating pair, thread. The created symbols are shown on the left side of Figure 4.

Afterwards the symbols can be generated automatically. For the dataset presented in this paper, an image with a white background and a resolution of 600x600 was selected as a start reference. The position of the symbols as well as the parameters of the respective symbol are selected randomly. In this way, any number of images can be generated for the training dataset. The variant representation of a symbol is exemplarily depicted for the fixed bearing, the spring and the double joint in Figure 5. This procedure solves the problems mentioned above, because the bounding boxes and labels are automatically produced for each image.

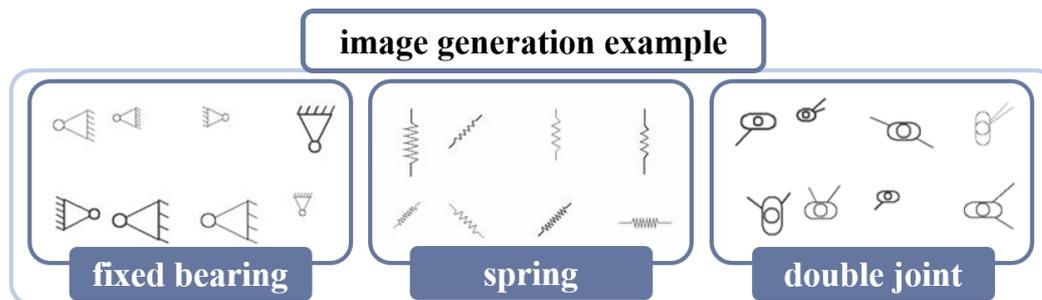


Figure 5: examples of randomly generated symbols

3.3 Training

From the variety of possible computer vision algorithms, the MASK R-CNN has been chosen for the task to recognize the symbols, which is the latest version of the capable R-CNN architecture. R-CNN stands for Region-based Convolutional Neural Network and was developed by Girshick et al. (2014). This approach was improved and extended by the same author two years later, to the Fast R-CNN (Girshick, 2015). Shortly after the release of Fast R-CNN, a new version has been published. It is called Faster R-CNN and was developed by Ren et al. (2015).

This new generation exchanges the selective search algorithm to a learning object detection algorithm. The method is separated into two modules. The first one is a fully convolutional network that proposes the regions. The second module is the same R-CNN detector from the Fast R-CNN. The first, new module is named Region Propose Network (RPN) and is an important part of the latest version, the MASK R-CNN. It generates the proposal of possible regions of objects. The second stage calculates the class labels and bounding boxes and, in contrast to the previously mentioned algorithms, it is able to create a binary mask for each RoI. Both stages are connected via a backbone network.

The backbone can consist of either resnet or resNeXt networks. Another possibility is to expand the resnet networks with Feature Pyramid Networks (FPN). This top-down network architecture allows gains in accuracy and speed. More information on the FPN-network can be found in (Lin et al. (2017)). A more detailed explanation of the MASK R-CNN is stated in (He et al. 2017)

3.3.1 Training settings

The hardware used for training the MASK R-CNN model was a workstation PC, with an Nvidia Titan V GPU with 12 GB, 32 GB RAM and an Intel Xeon W-2125 CPU. The training data was generated as stated in section 3.2. In total 200.000 images and bounding boxes were created randomly and used as input for the network.

The dataset is split into 180.000 images for training and 20.000 for validation. Another 500 independently generated images are used for the test-dataset. Figure 6 demonstrates the general procedure. The synthetic produced dataset is used for the training. Afterwards the finished model can be evaluated with the training-, validation- and test-metrics. If the evaluation values are not satisfying, the model must be optimized and trained with new parameters. If the model provides good accuracy values, an unknown test-dataset can be evaluated on the model. It consists of a small number of manually created principle sketches and allows to check the trained model. The sketches are sampled from construction catalogs or self-designed sketches, as seen in Figure 8 on the upper part of the Figure.

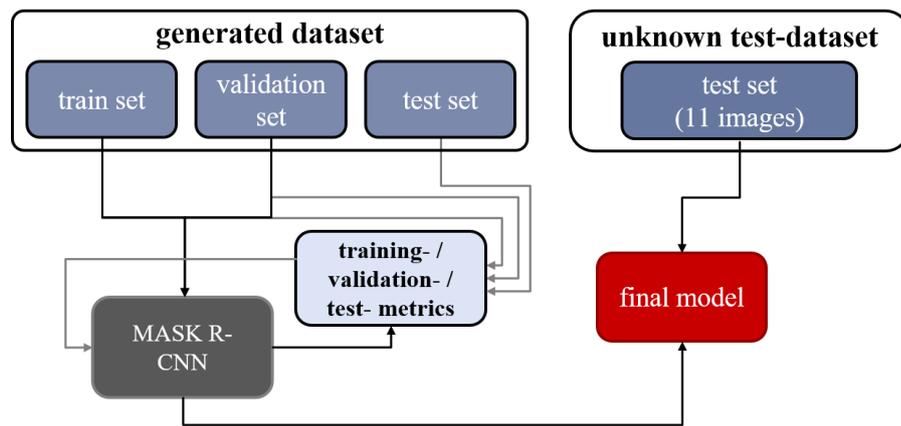


Figure 6: Overview of the evaluation process for the trained symbol detection model

The settings of the MASK R-CNN were slightly adjusted, so images could be used with a different resolution than the COCO dataset (Lin et al. 2014). The batch size was successively increased until the GPU was working stable at maximum capacity. Five pictures per GPU turned out to be the highest number. Furthermore, different backbone networks were examined, the resnet50 and the resnet101. Although the learning rate for the training of the model was adapted and iteratively optimized. Per epoch 1000 training steps were carried out, while the whole training process has 40 epochs. This results in a training time of nearly 1 day per network. The starting weights are taken from a trained set with the COCO dataset.

3.4 Results

For the evaluation of the calculated results, the metrics of the COCO Dataset are used (Lin et al. 2014). They consist of the intersection over union (IoU), the mean average precision (mAP) and the mean recall (mRc). All metrics and equations are explained in the following.

IoU analyses the overlap between the predicted bounding box (Bp) and the ground truth bounding box (Bgt). If the IoU is above a certain threshold, the detection is valid, otherwise, the detected bounding box is not taken into account.

$$\text{IoU} = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})} \quad (1)$$

The precision measures the ability of the model to detect the relevant objects. The true positive (TP) results are divide by the true positives and the false positives (FP).

$$\text{precision} = \frac{TP}{TP+FP} \quad (2)$$

In contrast to precision, the recall indicates how well the relevant objects are recognized. For the calculation, the true positives are divided by the true positives and false negatives (FN).

$$\text{recall} = \frac{TP}{TP+FN} \quad (3)$$

The metrics stated in the COCO dataset expand these equations. For calculating the average precision (AP) a 101-point interpolated AP definition is used. The average precision can be computed at different IoU thresholds. For the evolution of this model, the IoU threshold was set to 0.50. The mean average precision (mAP50) is the mean of all AP values at the fixed threshold for all classes. The mean Recall (mRc50) is accordingly the average over all recall values at the fixed IoU value. These two metrics are used for the evaluation of the trained model.

The first examination considers the influence of different backbone networks on accuracy and recall. The results for the different backbone networks are plotted in Figure 7. Furthermore, the mAP50 and mRc50 values are calculated for the training-, validation- and test-dataset. For the calculation of the evaluation metrics, 1000 images are used for the train- and validation- and 500 for the test results. The dataset was generated by the procedure stated in section 3.2.

However, with an IoU of 0.5, no differences between the two backbone networks were noticeable. Consequently, the IoU value was increased to 0.75, so the two results are now named mAP75 and mRc75.

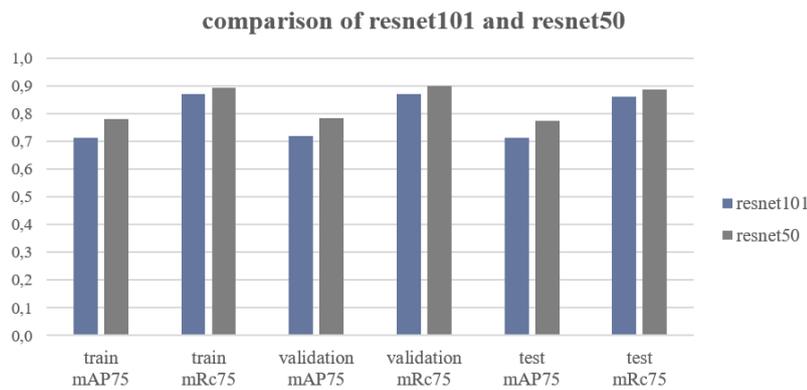


Figure 7: Comparison of different backbone networks

The resnet50 results are plotted in dark grey, whereas the resnet101 values are displayed in dark blue bars. In general, the results are very good. The mAP75 and mRC75 generally provide high precision and recall for both networks. But the bar diagram shows that the resnet101 works slightly better for the given task. For every dataset, the resnet101 network received higher precision and recall values, in comparison to the resnet50 network. For this reason, the resnet101 was picked as the backbone network.

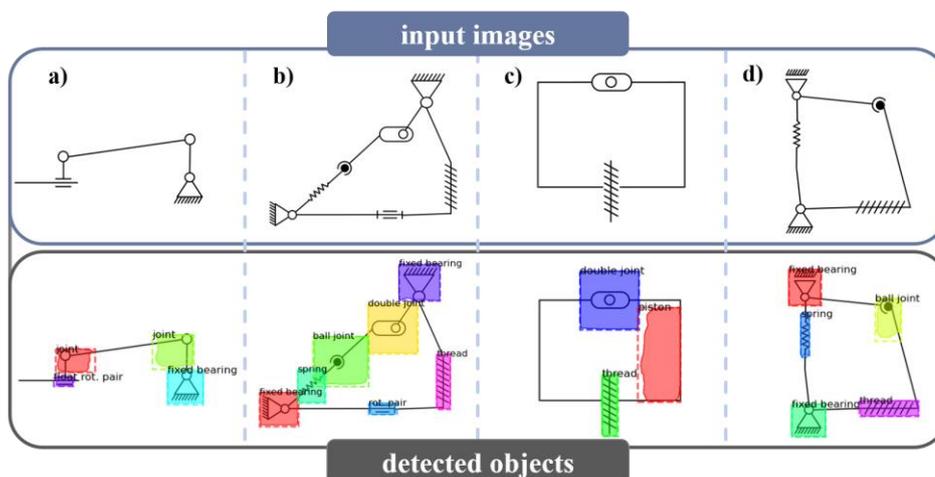


Figure 8: Examples of resulting symbol detection for the unknown test-dataset

Therefore, this layout is applied on the unknown test-dataset, which contains 11 images. One image is a collection of all symbols in the dataset and it is used to check if every symbol is correctly classified, without the appearance of intersections or joints. If this first check is not successful, the whole model must be optimized or newly trained. The remaining images are adaptations from sample principle sketches. Some were taken from the construction catalog (Roth, 2001), e.g. image a) and c) in Figure 8. The images had to be redrawn and manually annotated. Also, random sketches were created with the requirement to represent as many symbols as possible. Examples are the sketches b) and d) in Figure 8. These sketches were in reality not very reasonable, but they are an important component for the investigation of the recognition accuracy.

Figure 8 illustrates the detected results in the lower section of the graphic. The calculated masks and their according labels are displayed for every prediction of the model. The comparison shows that the procedure works generally well. The majority of the symbols connected to a sketch are recognized and no symbol in the images was missed. The resulting masks and bounding boxes of sketch a) and b) demonstrate this outcome. In these two drawings, all symbols are recognized correctly. However, the procedure does not yet work without errors, as shown in sketch c) and d). Here a connection was falsely detected as a piston. In sketch d) a floating bearing is misclassified as a fixed bearing. These errors still occur in some cases, but this should be corrected by more variance in the generation of the dataset.

For the small dataset with 11 images, a mAP50 of 0.545 and a mRc50 of 0.663 could be obtained. The progress of precision and recall can be observed in Figure 9. It becomes obvious that with a decreasing IoU value, both values increase significantly. That means, with an IoU value of 0.25 mAP and mRc are both above 0.8.

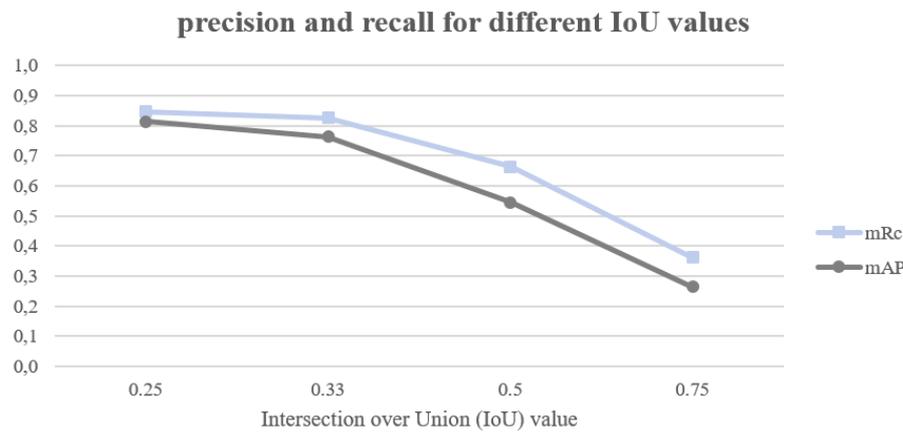


Figure 9: Precision and recall values for different IoU values on the unknown test-dataset

Because the unknown test-dataset is manually labelled, the corresponding values for precision and recall are not as good as those of the training dataset and are decreasing rapidly with a higher IoU value. Additionally, the small number of images in the dataset accelerates this process. For an application in a real engineering design activity, the precision and recall values should be improved. In the current state, the model could be used to search for an occurrence of specific symbols, because the high mAP und mRc with low IoU values indicate, that the right class is detected, but the position is not sufficient enough. An application with a high need for the correct position detection in the sketch is currently not very suitable. For further improvement of the process, it was observed that a change in resolution of the test images has a great influence on the detection of the symbols. Modifying the resolution when generating the synthetic dataset should solve this problem. Furthermore, a broader variety of the design parameters, especially the rotation and size of the different symbols should increase the evaluation values as well. Another important factor is the MASK R-CNN itself, where the possibilities for partial parameter optimization still exist.

4 CONCLUSION AND OUTLOOK

In this paper, a method for the detection of symbols in principle sketches is presented. The approach structures into the synthetic generation of the data, the training of the detection model and the evaluation. Example images of the training dataset and the resulting classification are shown, as well as the calculated bounding boxes and masks. It was proven that the presented method provides good results with the resnet50 and resnet101. Even the application on the unknown test-dataset displayed promising precision and recall values.

For future work, the recognition should be more independent of the resolution of the drawings. This has been discovered during the investigation and is revealed by the fact that the trained model wants to divide symbols that are too large into smaller ones. Therefore, the next step is to increase the size of the symbols in the generated dataset to possibly solve the problem. Furthermore, new symbols need to be added gradually during the creation of the training data. Other interesting research topics for this new method are the recognition of hand drawing sketches and to automatically scan old paper-based documentation. Both ideas would result in an updated data generation model.

The next big leap for this procedure is the transfer of the recognized symbols into graphs. For this purpose, a solution must be found to automatically transfer the masks or bounding boxes into a graph-based representation. Based on these graphs, further actions are conceivable, such as a similarity search, an optimization of the graph or a system that provides suggestions for the completion of the sketch. The development of an app for automatic recognition and use of sketches on a tablet can also be considered for a further practical application.

ACKNOWLEDGMENTS

The authors would like to thank the NVIDIA Corporation and the academic GPU Grant Program for the donation of a Tesla GPU.

This research work is part of the FAU “Advanced Analytics for Production Optimization” project (E|ASY-Opt) and funded by the Bavarian program for the “Investment for growth and jobs” objective

finance by the European Regional Development Fund (ERDF), 2014-2020. It is managed by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology. The authors are responsible for the content of this publication.

REFERENCES

- Deng, X., Li, T., Xu, Y., Cao, Y., Kong, C., Zhang, E., (2020), The Computer Vision-based Tolerancing Callout Detection Model. *Procedia CIRP*. 92. 134-139. <https://doi.org/10.1016/j.procir.2020.05.189>
- Elyan, E., Moreno-García, C., Jayne, C., (2018), Symbols Classification in Engineering Drawings. International joint conference on neural networks 2018 (IJCNN), 8-13 July 2018, Rio de Janeiro, Brazil <https://dx.doi.org/10.1109/IJCNN.2018.8489087>.
- Everingham M., Eslami S.M.; Van Gool L., Williams C., Winn J., Zissermann A. (2015), The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision* Volume 111 (2015) 1, pp. 98–136. <https://doi.org/10.1007/s11263-014-0733-5>
- Fu, L., Kara, L., (2011). From engineering diagrams to engineering models: Visual recognition and applications. *Computer-Aided Design*. 43. 278-292. <https://doi.org/10.1016/j.cad.2010.12.011>
- Girshick R., Donahue J., Darrell T., and Malik J. (2014), Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 580-587, <https://dx.doi.org/10.1109/CVPR.2014.81>
- Girshick R. (2015), “Fast R-CNN,” *IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 1440-1448, <https://dx.doi.org/10.1109/ICCV.2015.169>
- Goetz S., Schleich B., Wartzack S. (2018), A new approach to first tolerance evaluations in the conceptual design stage based on tolerance graphs, *Procedia CIRP*, Volume 75, Pages 167-172 <https://doi.org/10.1016/j.procir.2018.04.030>.
- He, K.; Gkioxari, G.; Dollár, P.; Girshick, R., et al. (2017) Mask R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, pp. 2980-2988, <https://dx.doi.org/10.1109/ICCV.2017.322>
- Kasturi R., Bow S. T., El-Masri W., Shah J., Gattiker J. R. and Mokate U. B. (1990), “A system for interpretation of line drawings,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 978-992, <https://dx.doi.org/10.1109/34.58870> .
- Lawrence S., Giles C. L., Ah Chung Tsoi and Back A. D.(1997), “Face recognition: a convolutional neural-network approach,” in *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98-113, <https://dx.doi.org/10.1109/72.554195>
- Leibe B., Leonardis A., Schiele B. (2004), Combined object categorization and segmentation with an implicit shape model. *Proceedings 8th Eur. Conference. Computer Vis. (ECCV)*.
- Lin T.-Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick C. L. (2014) Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) *Computer Vision – ECCV 2014*. vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48
- Lin T.-Y., Dollár P., Girshick R., He K., Hariharan B., and Belongie S. (2017), Feature Pyramid Networks for Object Detection, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 936-944, <https://dx.doi.org/10.1109/CVPR.2017.106>
- Lindemann, U. (2008), *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht an-wenden*. Berlin u. a.: Springer-Verlag, 2008, <https://dx.doi.org/10.1007/978-3-642-01423-9>
- Okazaki A., Kondo T., Mori K., Tsunekawa S. and Kawamoto E. (1988), An automatic circuit diagram reader with loop-structure-based symbol recognition, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 331-341, <https://dx.doi.org/10.1109/34.3898> .
- Ren S., He K., Girshick R., and Sun J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, <https://dx.doi.org/10.1109/TPAMI.2016.2577031>
- Roth K. (2000), *Konstruieren mit Konstruktionskatalogen: Band 1: Konstruktionslehre*. 3., Aufl. 2000. Springer-Verlag Berlin Heidelberg <https://doi.org/10.1007/978-3-642-17466-7>
- Roth K. (2001), *Konstruieren mit Konstruktionskatalogen: Band 2: Kataloge*. 3., Aufl. 2000. Springer-Verlag Berlin Heidelberg <https://doi.org/10.1007/978-3-642-17467-4>
- Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A. C., Fei-Fei L. (2014), ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 211–252. <https://doi.org/10.1007/s11263-015-0816-y>
- Sinha R. K., Pandey R., Pattnaik R. (2017), Deep Learning For Computer Vision Tasks: A Review, *International Conference on Intelligent Computing and Control (I2C2)* <https://doi.org/10.1155/2018/7068349>
- VDI 2222 (1997), Methodic development of solution principles. Düsseldorf, VDI
- VDI 2221 (2019), Design of technical products and systems - Model of product design. Düsseldorf, VDI
- VDI 2223 (2004), Systematic embodiment design of technical products. Düsseldorf, VDI
- Wiley V., Lucas, T. (2018), Computer Vision and Image Processing: A Paper Review. *International Journal of Artificial Intelligence Research* Volume 2, pp. 28–36. <https://doi.org/10.29099/ijair.v2i1.42>