Improving Symbol Detection on Engineering Drawings Using a Keypoint-Based Deep Learning Approach

Benedikt Faltin, Phillip Schönfelder, Markus König
Ruhr University Bochum, Germany
benedikt.faltin@ruhr-uni-bochum.de

Abstract. Since the position and orientation of objects in engineering drawings are indicated by symbols, the automated processing of these documents relies on the exact localization and classification of the symbols. However, if drawings are stored as pixel-based image files such information is not directly available. While the detection of symbols is the main focus of most symbol recognition approaches, few of them infer the orientation of the detected symbols, additionally. However, they rely on the results of a detection network and feed them to a regression network. The problem with these approaches is the need for two specialized networks, resulting in error aggregation. As a remedy, state-of-the-art methods for human pose detection may be utilized to solve the outlined problem. In this study, different keypoint-based models are compared, namely (1) Keypoint R-CNN, (2) YOLOv7-Pose, and (3) a custom two-stage approach as a baseline. The keypoint-based networks show improved results on the test data.

1. Introduction

Technical drawings are essential in various engineering fields, such as mechanical, electrical, and civil engineering, where they serve as a universal language to communicate design specifications. These drawings contain various standardized symbols that may represent objects and dimensions, or more abstract objects like auxiliary symbols that support the drawing's interpretation. However, if the drawings are stored in a pixel-based format, automatically digitizing and processing these drawings can be challenging. In that case, the information regarding the type, position, and orientation of the symbols is not directly available, which emphasizes the need for automatic symbol detection methods for engineering drawings.

Previous studies, as shown in Section 2, are mostly limited to detecting the symbols, and neglect inferring their orientation. To overcome this issue, few studies propose a two-stage approach consisting of a detection and a regression stage. However, such processing pipelines are cumbersome to train, may lead to error aggregations, and cause problems in downstream tasks. Therefore, we show how state-of-the-art human pose detection methods, such as Mask R-CNN (He et al. 2017) and YOLO-Pose (Maji et al. 2022), can be leveraged to address the symbol pose estimation task. These single-stage methods rely on keypoint-based models that can detect and classify symbols, as well as locate the symbols' keypoints, and thus, infer the symbols' orientation. The contributions of this paper are, first, a proposal for utilizing keypoint-based object detection for symbol pose estimation, and second, a comparison of two keypoint-based detection models with a two-stage baseline model.

This paper is structured as follows: Section 2 presents prior works in symbol detection on technical drawings. Section 3 describes the methodology of our study, including the data set, experimental setup, and evaluation metrics. In Section 4, we present our findings, including a detailed analysis of the performance of different keypoint-based detection models and an interpretation of the results. The paper concludes with a summary of the results and suggestions for further research.

2. Related literature

The automated detection of symbols in engineering drawings and architectural drawings is an active research area with many proposed solutions. Most involve some heuristics (Ablameyko et al. 2007), and more recent approaches mainly use deep learning methods (Moreno-García et al. 2019).

In the field of engineering, piping and instrumentation diagrams (P&IDs) are commonly used to represent complex systems. Elyan et al. (2018) combine various heuristic methods to localize symbols in P&IDs. Their approach includes thresholding, blurring, circle Hough transform, and text/graphics separation. The identified symbols are then classified with Convolutional Neural Networks (CNN), random forests, and support vector machines. In Elyan et al. (2020), the same task is addressed with YOLO (Redmon et al. 2016), which yields a robust bounding box-based symbol detector for the document type at hand (cf. Figure 1). While Nurminen et al. (2020) (cf. Figure 1) and Gupta et al. (2022) also use YOLO versions for symbol detection in P&IDs, they present approaches for enlarging the training data set with synthetically generated images.

A related research field is the detection of symbols in architectural drawings, which mainly represent building components or interior. A prominent public data set in this regard is SEYSD (Delalandre et al. 2010) which is used for evaluation in various studies. For instance, Rezvanifar et al. (2020) leverage YOLOv2 (Redmon & Farhadi 2017) to detect doors, windows, appliances, and furniture in the drawing files (cf. Figure 1). Mishra et al. (2021) train a Cascaded Mask R-CNN (Cai & Vasconcelos 2018) for symbol detection in floor plans and propose their dataset for the task called Synthetic Floor Plan Images (SFPI). Another study published by Ziran et al. (2020) suggests using Faster R-CNN (Ren et al. 2017) for the task.

It is evident that, while object detection in engineering and architectural drawings is a research field often addressed, most existing studies suggest the use of bounding box-based object detection only. Hence, the symbols of interest are localized on the drawing and assigned symbol types, however, the symbols' orientations remain unrecognized. This is where the symbol pose estimation task could benefit from using keypoint-based detection methods.

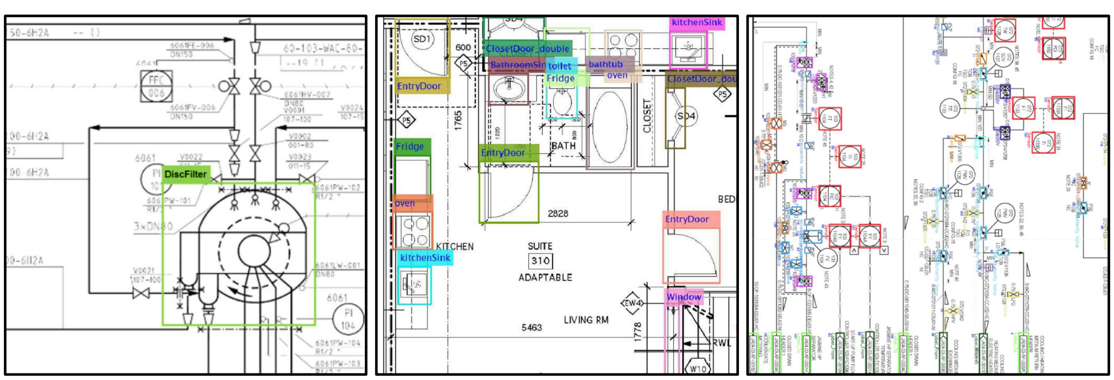

Figure 1: Example demonstration of prior symbol detection studies. From left to right: Nurminen et al. (2020), Rezvanifar et al. (2020), Elyan et al. (2020).

Keypoint-based detection methods are commonly used in the research field of pose estimation, which aims to estimate the position and orientation of an object by detecting its keypoints. A prominent application is human pose detection, including hand pose detection, and facial pose detection (Lui et al. 2022). Other studies have adapted the concept of keypoint-based inference to their specific use case. For instance, Zhao et al. (2017) propose a keypoint-based classification method for aircraft types, in which a CNN is utilized to detect the main landmarks of the aircraft. These landmarks are matched to a set of predefined templates to infer the aircraft type. Another application of keypoint-based detection is demonstrated by Xu et al. (2020), who use keypoint detection to infer the pose of space objects. First, the objects in the image are detected by Faster R-CNN, then cropped from the image, and finally, the keypoints are inferred using a custom CNN.

Building upon the state-of-the-art of engineering symbol detection methods, our study shows how keypoint-based detection may be used to address localization and orientation inference of engineering symbols with deep learning models. This detection approach may also be leveraged to boost performance and streamline the inference process.

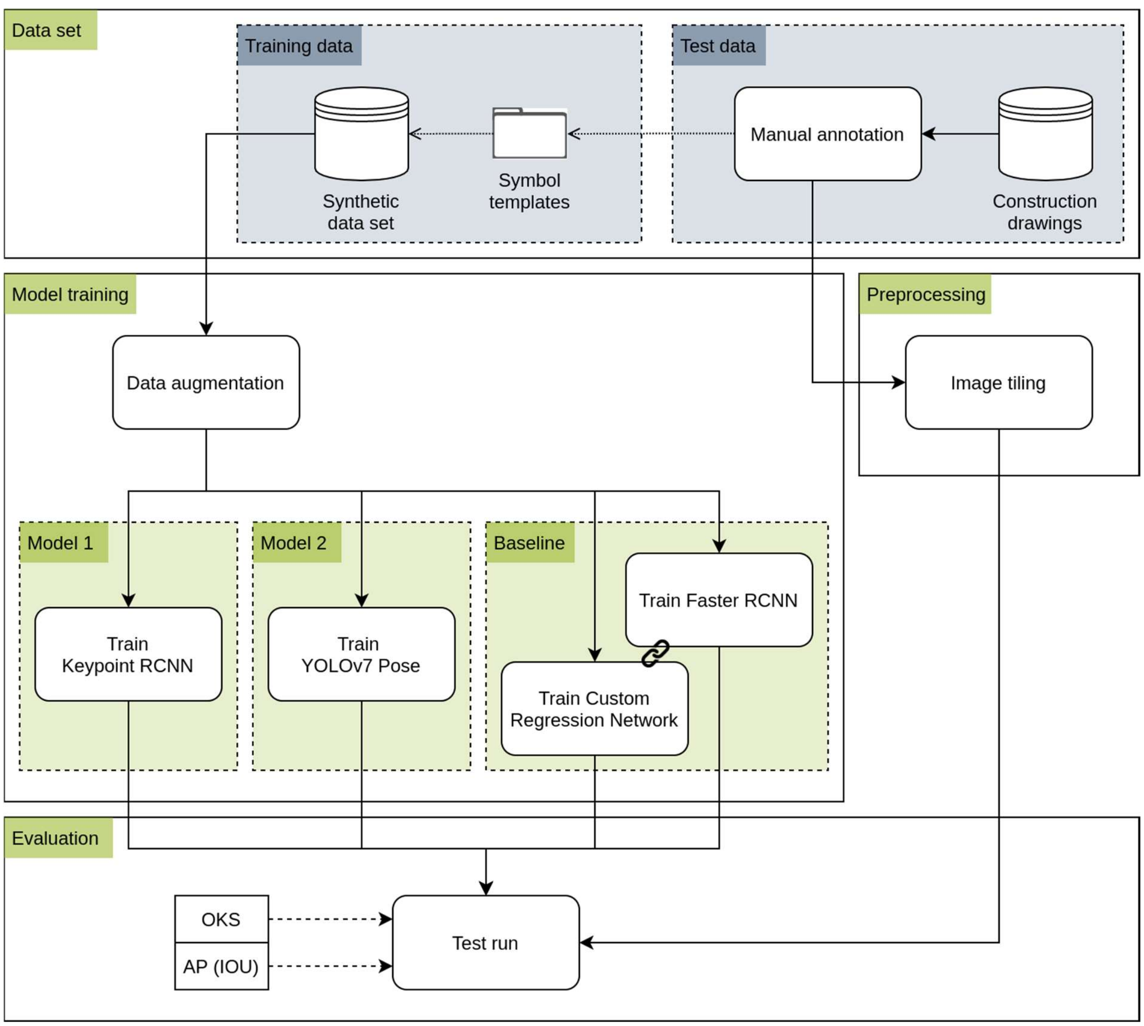

Figure 2: Overview of the methodology of this study on pose estimation for engineering symbols.

3. Methodology

The proposed approach is evaluated by comparing the performance of different neural network architectures. Therefore, two state-of-the-art architectures are used: Keypoint R-CNN, a specialization of Mask R-CNN, and YOLO-Pose, based on YOLOv7 (Wang et al. 2022). The performances of these architectures are compared to a baseline model consisting of the object detection network Faster R-CNN and a custom regression model. The networks are trained on a set of synthetically generated training images and tested on a set of real-world drawings. Figure 2 summarizes the methodology of this paper.

3.1 Data set

A symbol usually comprises two components: first, a marker, which indicates the position and orientation of the symbol, and second, the accompanying text, which contains additional information such as dimensions or references. To accurately assess a symbol, the marker and the text component must be reliably recognized and linked to each other. Therefore, to establish the connection between the two components, the whole extent of the symbol must be recognized.

Using the example of a section symbol, Figure 3 (left) shows a bounding box-based representation of a symbol, including its marker, text, and extent. Now, to infer the exact position of the marker, in addition to recognizing its bounding box, the position of its keypoints must be determined. However, this requires additional annotation effort, an extension of the inference pipeline, and, moreover, may lead to error aggregation. Therefore, in this study, we propose transforming the bounding box-based symbol detection problem to a keypoint-based pose estimation task.

Keypoints mark salient points of objects, such as corners, edges, or other distinctive features. This can be seen in Figure 3 (right), where the most descriptive points are annotated and enumerated. The keypoints (1) and (2) denote the position of the marker axis. Points (3), (4), and (5) determine the position of the triangle, indicating the symbol's direction. Lastly, point (6) lies in the center of the reference character. It is noted that this idea may be extended to other types of symbols. With these keypoints, the position and orientation of the symbol marker are immediately inferred, while simultaneously determining the text position. Since all points belong to the same object instance, the connection is also explicitly provided.

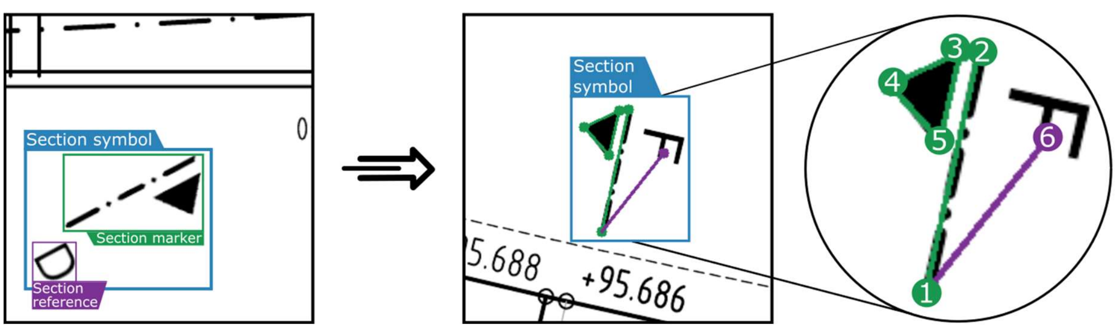

Figure 3: Difference between bounding box-based (left) annotation and keypoint-based (middle) annotation. For the example of the section symbol, six keypoints are annotated.

Therefore, in this study, state-of-the-art neural networks are trained for keypoint detection. To overcome data scarcity and avoid time-consuming manual annotation, we follow the approach proposed by Vilgertshofer et al. (2019) and generate the training images synthetically. Following a copy-and-paste strategy, a symbol template is pasted onto a randomly cropped region from real drawings. For testing, a set of 53 real drawings is used, that are annotated manually.

3.2 Model architectures

Three different deep learning models are compared for detecting the symbols and the respective keypoints, namely Keypoint R-CNN, YOLOv7-Pose and a two-stage model, consisting of Faster R-CNN and a custom regression model. In this section, the inner architectures of the models are briefly explained (cf. Figure 4).

Since Keypoint R-CNN is based on Mask R-CNN, it consists of a convolutional backbone for feature extraction. The feature maps extracted by the backbone are passed through a region proposal network to identify areas in the image that contain an object. The identified regions are then aligned with the feature maps via the Region Of Interest (ROI) alignment layer. The output of this process is subsequently processed by two distinct branches: the mask branch predicts the object masks, while the box branch outputs the bounding boxes and the object classes. To facilitate keypoint detection, He et al. (2017) modify the mask branch to output an individual mask per keypoint.

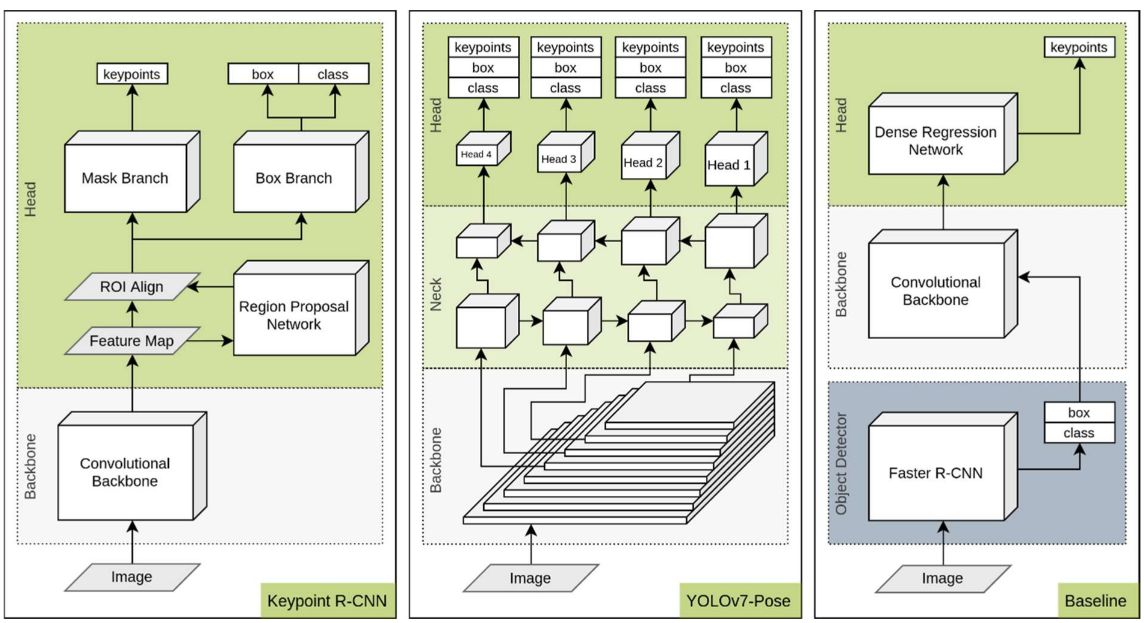

Figure 4: Simplified schematics of the used deep learning architectures.

In comparison, YOLOv7-Pose is based on YOLO-Pose, which itself is an extension of YOLOv5[1]. The YOLOv7 model extracts the feature maps through a backbone. Features from different layers are fed into a stacked set of convolutional blocks. The idea is to consider features from different scales. Therefore, YOLOv7 comprises four independent prediction heads that output the object bounding boxes as well as object classes. The detection heads of the underlying YOLO architecture are modified for the YOLO-Pose architecture. In addition to the bounding boxes and object classes, the model predicts each keypoint's position and visibility.

Lastly, the custom baseline model is a combination of two models. First, Faster R-CNN is employed for object detection and classification. The predicted bounding boxes are used as input for a custom regression model. Therefore, the image is cropped to the region of interest that extends the bounding box by a margin of 50 pixels. The regression network consists of six convolutional blocks, each containing two convolutional layers and a max pooling layer. The output of the last convolutional block is fed through two dense layers that output the coordinates of the keypoints.

4. Experimental results

To assess the performance of the different models a set of 53 real-world infrastructure drawings is used for testing. The drawings are labeled manually and then cropped into tiles with the required image size to be compatible with the neural networks. The test data set consists of a total of 411 tiles. For training, 5,000 images are synthesized by the copy-and-paste technique. To further increase the diversity of the training data, several data augmentation techniques, e.g., flipping, mosaic, and scaling are applied.

Keypoint R-CNN and YOLOv7-Pose are trained end to end on the synthetically generated images. Regarding the two-stage approach, Faster R-CNN is trained for the object detection task, while the custom regression network is trained for the pose estimation task independently. Training is stopped when a model reaches its minimum validation error: Faster R-CNN is trained for 30 epochs, Keypoint R-CNN for 15 epochs, YOLOv7-Pose for 300 epochs, and the regression network is trained for 50 epochs.

The performance of the models is evaluated following the COCO guidelines[2] (Lin et al. 2014). A prediction is assigned to the ground truth object when the Intersection over Union (IoU) of the respective bounding boxes exceeds a certain threshold. However, for the evaluation of the keypoint-based approaches, relying solely on the bounding box is not feasible since the location of the keypoints is not taken into account. To overcome this, the Object Keypoint Similarity (OKS) metric is used for keypoint-based approaches. The OKS for a symbol with N keypoints is defined by

$$OKS = \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right), \quad (1)$$

where the difference between the predicted and the ground truth keypoints is given by the Euclidean distance d. Since small deviations in the position of the keypoints in large objects are insignificant, the variable s takes the size of the object into account. Although keypoints

[1] https://github.com/ultralytics/yolov5 (Accessed: 30 May 2023)
[2] https://cocodataset.org (Accessed: 30 May 2023)

are generally defined at salient points of an object, the exact position of the keypoint can sometimes be ambiguous. This is illustrated in Figure 3 (right), where keypoint (6) is placed at the center of the reference. Due to the uncertainty about the exact position of the midpoint, keypoint (6) tends to scatter. This problem can have a negative impact on the objectiveness of the OKS metric. Therefore, each keypoint is assigned a constant value k, which reduces the impact of variance for the respective point position. In this study, k is assigned a higher value of 0.1 for keypoint (6) and a lower value of 0.025 for points (1) to (5). These values are chosen based on the idea of using small values for keypoints with low scatter and are not empirically determined.

To ensure the fairness of the comparison, all models are evaluated on the object detection task as well as the symbol pose estimation task. Note that the keypoint-based models solve both tasks in a single inference call, whereas the baseline model utilizes two separate networks. The results are shown in Table 1 (left), respectively. Exemplary pose estimation results are displayed in Figure 5.

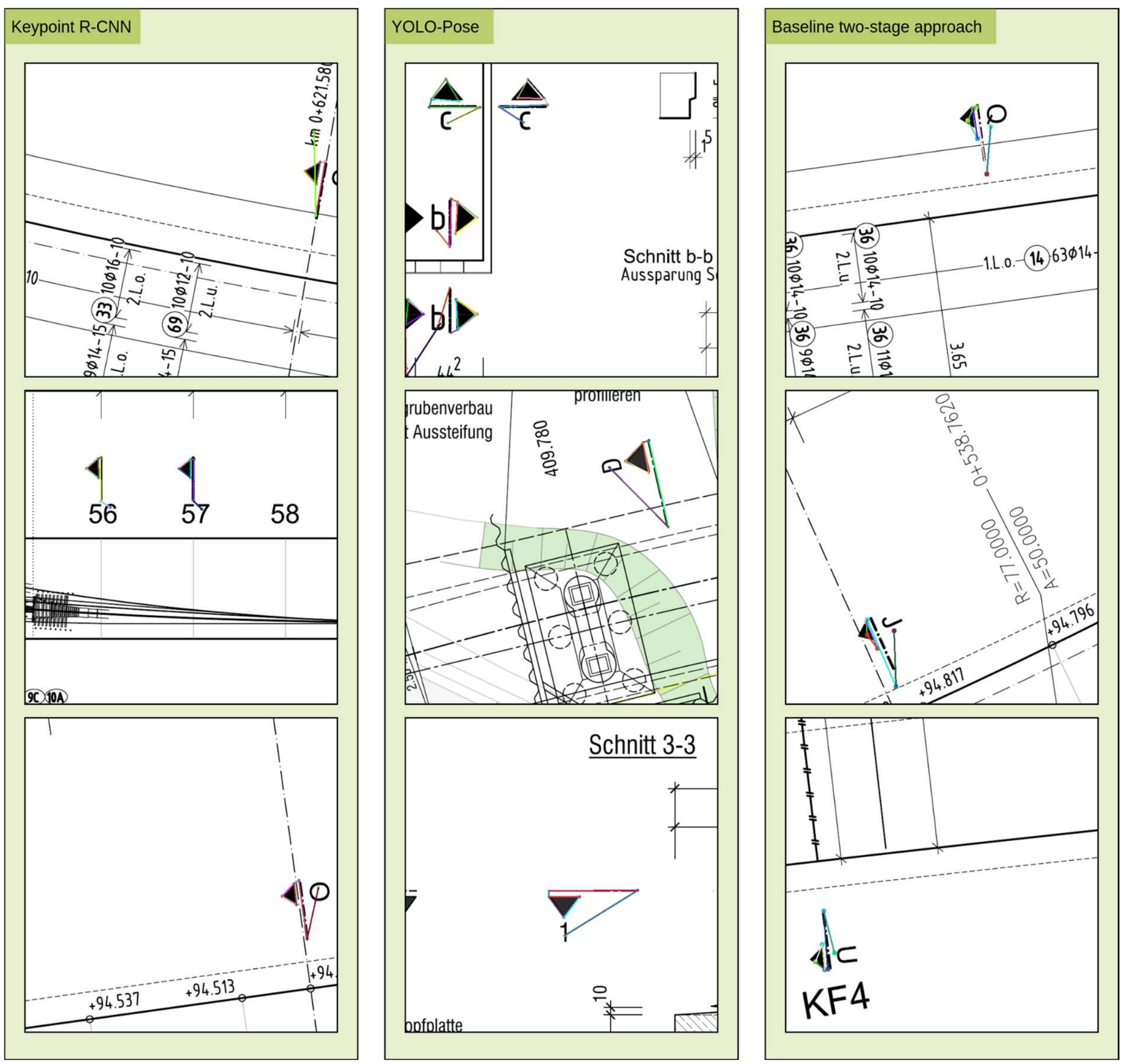

Figure 5: Test results of symbol and keypoint detection, resulting in pose estimations. The columns correspond to Keypoint R-CNN (left), YOLOv7Pose (middle), and the baseline model (right).

Table 1: Object detection performance scores (left) and keypoint detection performance scores (right) achieved by the models on the real-world bridge construction drawings. Scores are given in mean Average Precision (mAP). Details about the score computations are given in (Padilla et al. 2022).

	$mAP_{0.50}^{IoU}$	$mAP_{0.75}^{IoU}$	$mAP_{0.50:0.95}^{IoU}$	$mAP_{0.50}^{OKS}$	$mAP_{0.75}^{OKS}$	$mAP_{0.50:0.95}^{OKS}$
CustomNetwork	43.5%	28.0%	24.5%	0.3%	0.0%	0.0%
YOLOv7Pose	65.6%	12.3%	28.9%	68.0%	49.8%	43.8%
Keypoint R-CNN	75.2%	23.6%	**32.4%**	81.6%	56.5%	**51.8%**

5. Discussion

As can be observed in Table 1 (left), the keypoint-based models outperform the object detection model regarding the bounding box predictions. This is in accordance with the results noted by He et al. (2017). The better performance might be due to the multi-task loss, where the model is forced to learn more meaningful features during training, to improve for both the object detection task as well as the keypoint detection. In terms of $mAP_{0.50:0.95}^{IoU}$ Keypoint R-CNN (32.4%) performs best, followed by YOLOv7-Pose (28.9%).

Regarding the pose estimation performance, Table 1 (right) shows that Keypoint R-CNN outperforms the other two models by a large margin. Notably, the custom network performs poorly. This has two main reasons: On the one hand, the inaccuracies induced by the Faster R-CNN model lead to error aggregation. On the other hand, the evaluation in terms of the OKS metric disregards all predictions that fall below the threshold. As can be seen in Figure 5 (right), the custom network is able to detect the keypoints, but not as precisely as the other two models. In particular, despite a somewhat decent performance, the custom model does not make any predictions with OKS higher than 0.5. However, a prediction rated with an OKS score below 0.5 might still be considered appropriate for a given task (cf. Figure 6). Since the choices of the OKS threshold and the k values are somewhat arbitrary, the specific numbers require careful interpretation. Nonetheless, this does not change the fact that the Keypoint R-CNN and YOLOv7-based models are far superior to the baseline model.

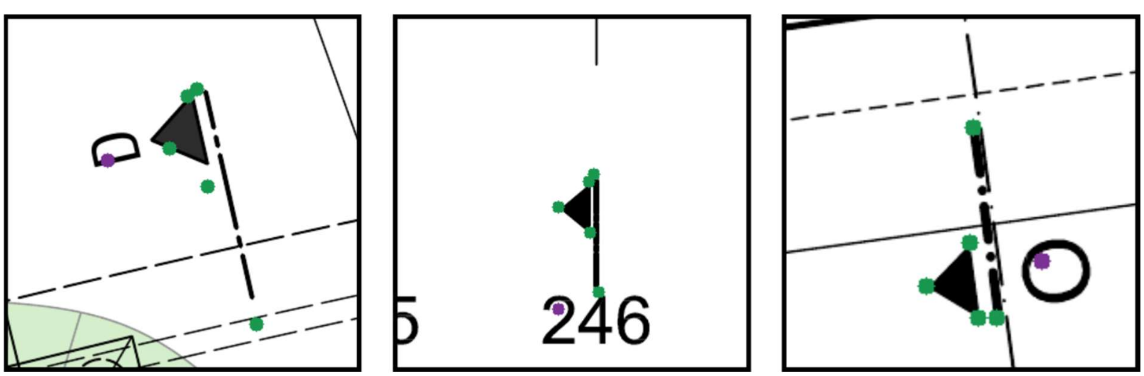

Figure 6: Keypoint prediction with an OKS of 0.45 (left), 0.75 (middle), and 0.95 (right).

6. Conclusion

In conclusion, this paper demonstrates the efficacy of keypoint-based object detection models, such as Keypoint R-CNN and YOLOv7-Pose, for symbol pose estimation in technical drawings. The study compares these models with a two-stage baseline approach. The results indicate that the keypoint-based models outperform the baseline approach in terms of both keypoint detection accuracy and bounding box detection accuracy. In addition, the keypoint detection also allows for the automatic linkage of symbol marker and reference, while determining the exact position of the symbol. This improves upon other object detection-based approaches, such as Faltin et al. (2022), which require detecting three objects per instance and the rule-based linkage. However, the proposed method and its results are limited to one drawing style and may not generalize to other domains or drawing collections. Therefore, further research is needed to explore the performance of the models on different symbols and drawing styles.

Acknowledgements

This research as part of the project "BIMKIT – Bestandsmodellierung von Gebäuden und Infrastrukturbauwerken Mittels KI zur Generierung von Digital Twins"[3] is funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant number 01MK21001J and is supported by the German Aerospace Center (DLR), Cologne.

The authors would like to express their gratitude towards Markus Scheffer from SD Engineering – A Socotec Company, who generously provided the drawing data set.

References

Ablameyko, S.V., Uchida, S. (2007). Recognition of engineering drawing entities: review of approaches, Int. J. Image Grap., 7(4), pp. 709–733. doi:10.1142/S0219467807002878.

Cai, Z., Vasconcelos, N. (2018). Cascade R-CNN: Delving Into High Quality Object Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, Salt Lake City, UT, USA, pp. 6154–6162. doi:10.1109/CVPR.2018.00644.

Delalandre, M., Valveny, E., Pridmore, T., Karatzas, D. (2010). Generation of synthetic documents for performance evaluation of symbol recognition & spotting systems, Int. J. Doc. Anal. Recognit. (IJDAR), 13, pp. 187–207. doi:10.1007/s10032-010-0120-x.

Elyan, E., Garcia, C.M., Jayne, C. (2018). Symbols Classification in Engineering Drawings. In: Proceedings of the International Joint Conference on Neural Networks, 2018, Rio de Janeiro, Brazil, pp. 1–8. doi:10.1109/IJCNN.2018.8489087.

Elyan, E., Jamieson, L., Ali-Gombe, A. (2020). Deep learning for symbols detection and classification in engineering drawings, Neural Netw., 129, pp. 91–102. doi:10.1016/j.neunet.2020.05.025.

Faltin, B., Schönfelder, P., König, M. (2023). Inferring Interconnections of Construction Drawings for Bridges Using Deep Learning-based Methods. In: Proceedings of the European Conference on Product and Process Modelling (ECPPM) – eWork and eBusiness in Architecture, Engineering and Construction, 2022, Trondheim, Norway, pp. 343–350. doi:10.1201/9781003354222-44.

Gupta, M., Wei, C., Czerniawski, T. (2022). Automated Valve Detection in Piping and Instrumentation (P&ID) Diagrams. In: Proceedings of the 39th International Symposium on Automation and Robotics in Construction (ISARC), 2022, Bogotá, Colombia, pp. 630–637. doi:10.22260/ISARC2022/0088.

[3] https://www.bimkit.eu (Accessed: 30 May 2023)

He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, Venice, Italy, pp. 2980–2988. doi:10.1109/ICCV.2017.322.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P. (2014). Microsoft COCO: Common Objects in Context. In: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.). Proceedings of the European Conference on Computer Vision, 2014, Zurich, Switzerland. Cham: Springer International Publishing (Lecture Notes in Computer Science (LNCS)), pp. 740–755. doi:10.1007/978-3-319-10602-1_48.

Liu W., Bao Q., Sun Y., Mei T. (2022). Recent Advances of Monocular 2D and 3D Human Pose Estimation: A Deep Learning Perspective. ACM Comput. Surv. 55(4), p. 80. doi:10.1145/3524497.

Maji, D., Nagori, S., Mathew, M., Poddar, D. (2022). YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, New Orleans, LA, USA, pp. 2636–2645. doi:10.1109/CVPRW56347.2022.00297.

Mishra, S., Hasmi, K.A., Pagani, A., Liwicki, M., Stricker, D., Afzal, M.Z. (2021). Towards Robust Object Detection in Floor Plan Images: A Data Augmentation Approach, Appl. Sci., 11(23), p. 11174. doi:10.3390/app112311174.

Moreno-García, C.F., Elyan, E., Jayne, C. (2019). New trends on digitisation of complex engineering drawings, Neural. Comput. Appl., 31(6), pp. 1695–1712. doi:10.1007/s00521-018-3583-1.

Nurminen, J.K., Rainio K., Numminen, J.-P., Syrjänen, T., Paganus, N., Honkoila, K. (2020). Object Detection in Design Diagrams with Machine Learning. In: R. Burduk, M. Kurzynski, M. Wozniak (Eds.). Progress in Computer Recognition Systems. Cham: Springer International Publishing (Advances in Intelligent Systems and Computing (AISC)), pp. 27–36. doi:/10.1007/978-3-030-19738-4_4.

Padilla R., Passos, W.L., Dias, T.L.B., Netto, S., da Silva, E.A.B. (2021). A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit, Electronics, 10(3), p. 279. doi:10.3390/electronics10030279.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, Las Vegas, NV, USA, pp. 779-788. doi:10.1109/CVPR.2016.91.

Redmon, J., Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, Honolulu, HI, USA, pp. 6517–6525. doi:10.1109/CVPR.2017.690.

Ren, S., He, K., Girshick, R., Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Trans. Pattern Anal. Mach. Intell., 39(6), pp. 1137–1149. doi:10.1109/TPAMI.2016.2577031.

Rezvanifar, A., Cote, M., Albu, A.B. (2020). Symbol Spotting on Digital Architectural Floor Plans Using a Deep Learning-Based Framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, Seattle, WA, USA, pp. 568–569. doi:10.1109/CVPRW50498.2020.00292.

Vilgertshofer, S., Stoitchkov, D., Borrmann, A., Menter, A., Genc, C. (2019). Recognising railway infrastructure elements in videos and drawings using neural networks. In: Proceedings of the Institution of Civil Engineers – Smart Infrastructure and Construction, 172, 2019, Chania, Crete, Greece, pp. 19–33. doi:10.1680/jsmic.19.00017.

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. doi:10.48550/arXiv.2207.02696.

Xu J., Song B., Yang X., Nan X. (2020). An Improved Deep Keypoint Detection Network for Space Targets Pose Estimation, Remote Sens., 12(23), p. 3857. doi:10.3390/rs12233857.

Zhao A., Fu K., Wang S., Zuo J., Zhang Y., Hu Y., Wang H. (2017). Aircraft Recognition Based on Landmark Detection in Remote Sensing Images, IEEE Geosci. Remote. Sens. Lett., 14(8), pp. 1413–1417. doi:10.1109/LGRS.2017.2715858.