

AUTOMATED EXTRACTION OF BRIDGE GRADIENT FROM DRAWINGS USING DEEP LEARNING

Hakan Bayer, Benedikt Faltin & Markus König
Ruhr University Bochum, Germany

ABSTRACT: Digital methods such as Building Information Modeling (BIM) can be leveraged, to improve the efficiency of maintenance planning of bridges. However, this requires digital building models, which are rarely available. Consequently, these models must be created retrospectively, which is time-consuming when done manually. Naturally, there is a great interest in the industry to automate the process of retro-digitization. This paper contributes to these efforts by proposing a multistage pipeline to automatically extract the gradient of a bridge from pixel-based construction drawings using deep learning. The bridge gradient, a key element of the structure's axis, is critical for describing the elevation profile and axis slope. This information is implicitly contained in the longitudinal view of bridge drawings as gradient symbols. To extract this information, the well-established object detection model YOLOv5 is employed to locate the gradient symbols inside the drawings. Subsequently, EasyOCR and heuristic rules are applied to extract the relevant gradient parameters associated with each detected symbol. The extracted parameters are then exported in a machine-interpretable format to facilitate seamless integration into other applications. The results show a promising 98% accuracy in symbol detection and an overall accuracy of 70%. Consequently, the pipeline represents a significant advance in automating the retro-digitization process for existing bridges by reducing the time and effort required.

KEYWORDS: Building Information Modeling, Computer Vision, Deep Learning, Symbol Detection, Optical Character Recognition, Construction Drawings

1. INTRODUCTION

Bridges play a central role in the transport network, as they are an essential element that creates connections and thus enables the transport of people and goods. To ensure their functionality and safe operation, regular inspections and effective maintenance management are of utmost importance. To support efficient maintenance planning, digital methods such as Building Information Modeling (BIM) can be employed. BIM refers to a digital collaboration method based on the creation and interdisciplinary exchange of digital building models (Borrmann et al., 2018). These models combine semantic information with geometric representations, acting as a single and central source of continuously enriching project information. Informed decisions can thus be made based on accurate and current data. Despite the potential BIM offers for all lifecycle phases, especially for the operation and maintenance (O&M) phase, it is most commonly used in the design phase of a construction project (Durdyev et al., 2022).

The limited utilization of BIM in the O&M phase arises from the requirement for digital as-built models of the structure, which are often not available. In many cases, this is due to the fact that the buildings were designed and constructed without the implementation of BIM. Therefore, these models have to be created retrospectively. Since this is a time-consuming manual task, there is a major research effort to assist or automate the process through the use of artificial intelligence (Schönfelder et al., 2023).

While many different sources of information can be used to create a digital building model of an existing structure, construction drawings are the most accessible. They not only contain geometric and semantic information about the building but also describe the internal structure of the components or building elements that are obstructed, e.g., buried underground. Therefore, drawings are an important source of information. This research contributes to the automatic creation of digital models from construction drawings by proposing a multi-stage pipeline for the automatic extraction of bridge gradient information from pixel-based construction drawings.

The bridge gradient illustrates the elevation profile of the bridge's axis and is, therefore, an essential information for a precise reconstruction of the superstructure's geometry. The course of the gradient is contained in the longitudinal view. It is implicitly described through gradient symbols that hold important details about elevation and slope at specific points along the bridge axis. Therefore, the pipeline encompasses several stages to automatically extract the gradient information from multiple locations and link the information. First, the pipeline utilizes state-of-the-art deep learning-based methods to detect the gradient symbols within the drawings. Subsequently, the text information associated with each symbol is extracted using optical character recognition

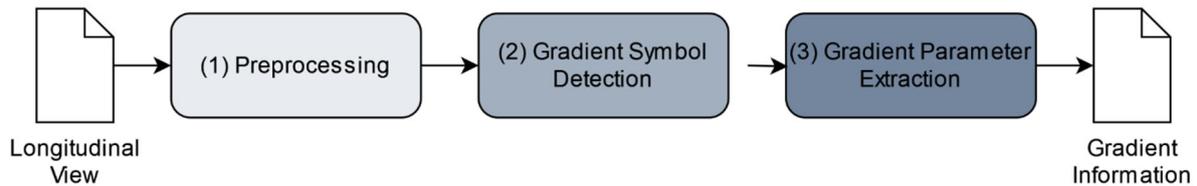


Fig. 1: Workflow diagram of the multi-stage pipeline for extracting the gradient of the bridge axis.

(OCR). Finally, the information is consolidated and harmonized into a structured data schema. The paper gives insight into the individual process steps and evaluates the performance of the proposed pipeline on a set of real-world bridge drawings.

The remaining sections of the paper are organized as follows: Section 2 reviews recent research publications on automatic reconstruction of digital models from drawings. The implemented pipeline is described in Section 3, with a detailed explanation of each step. Section 4 shows the results obtained when applying the proposed pipeline to real drawings. Finally, Section 5 discusses the results and limitations of this study and outlines a perspective for further research.

2. RELATED WORK

So far, little research has been published on digitizing technical drawings (Moreno-Garcia et al., 2019). Only a few of these publications deal with the (semi-)automatic analysis of drawings for infrastructures. Poku-Agyemang & Reiterer (2023) proposes a semi-automated process that utilizes the Douglas-Peucker algorithm to detect the corner points of illustrated components. These points are then used to reconstruct the component exterior edges, ultimately creating the bridge's geometry. A different semi-automatic framework is proposed by Akanbi & Zhang (2022). The proposed process involves converting the drawings into a vector-based data format, aligning the contained views, and finally using the extracted information to reconstruct the bridge's geometry. The reconstructed digital model is exported in the IFC (Industry Foundation Classes) data format. An approach based on deep learning methods is introduced by Mafipour et al. (2023). The authors present a pipeline employing YOLOv5 and CRAFT to automatically detect the individual components and texts in the drawings. The detected objects are then clustered based on their labels, as each component may appear in different views of the structure. These views are provided to an expert who uses this information to create the bridge based on a predefined parametric model manually. In contrast, Faltin et al. (2023) proposes a different approach for linking the views in construction drawings. The authors utilize FasterRCNN for detecting section symbols that illustrate the interconnections between views. Each detected symbol is uniquely identified through the section reference, allowing it to be mapped to the corresponding view using OCR on the view title. The interconnections are established across all views contained in a drawing set for a specific structure.

Overall, a larger number of publications exist on the reconstruction of high-rise building models from drawings. Wei et al. (2022) proposes a pipeline for detecting and reconstructing walls from floor plans. Firstly, the drawing is divided into patches, and the ResNet model is employed to identify patches containing walls. The walls are then detected in the positive patches using YOLOv3. All detections are merged to enable the utilization of Dynamo to create the digital model. Similarly, Zhao et al. (2020) uses the YOLO object detector to locate structural elements in the column structure and generate framework plan images. In a subsequent study, Zhao et al. (2021) continues the research by incorporating the superior Faster R-CNN model and introducing the creation of an IFC file from the extracted information. Kim et al. (2021) propose an approach for the detection of rooms, walls, and openings in floor plan images. The authors use a conditional generative adversarial network to create a heat map of the intersections and perform a style transfer to the floor plan image. Using the heat map of the connection points, the walls and openings are vectorized, which provides important information for recreating the building geometry.

However, to the best of the author's knowledge, no publication has been made that addressed the extraction of the bridge gradient from pixel-based drawings. This research aims to close the identified gap.

3. METHODOLOGY

A detailed explanation of the implemented methods is provided in the following section. Fig. 1 presents an overview of the proposed process. The input for the pipeline is a pixel-based longitudinal view of the bridge, as it

contains the required gradient information. In this paper, it is assumed that the view has already been extracted from the complete drawing set, which the engineer can do manually in advance. The pipeline first preprocesses (1) the view by dividing it into smaller patches to enable the symbol detection. The gradient symbols (2) are detected within these patches, and for each symbol, the associated gradient parameters (3) are extracted. Finally, the information is exported in a structured format to facilitate its utilization within BIM modeling software.

3.1 Dataset Creation and Annotation

To detect the gradient symbol in the drawings, the state-of-the-art object recognition network YOLOv5 (Jocher et al., 2022) is employed. Training the network requires a dataset of bridge construction drawings manually annotated with the gradient symbol. However, since the gradient symbol only occurs in small numbers in a longitudinal view, this results in a limited number of training data points. This is insufficient to receive a well-trained model. Therefore, the training data is synthetically generated, using a copy-and-paste strategy following Faltin et al. (2023). The real annotated data is only used to test the network.

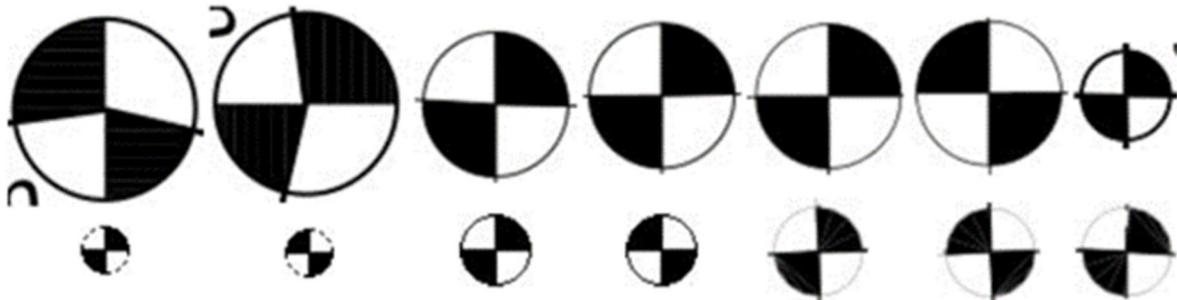


Fig. 2: Variation in created gradient symbols, sizes ranging from 85x85 to 20x20 pixels.

For the synthetic data, a template set of 14 unique gradient symbols is created and employed in the process (cf. Figure 2). These symbols vary in size from 85x85 to 20x20 pixels, shape, and texture, providing increased diversity in order to improve the models' ability to generalize.

The gradient symbol is chosen from the available template set and is randomly inserted into the background images. These background images are randomly cropped from bridge construction drawings which do not contain the gradient symbol and consist of various segments of construction drawings with different pixel sizes. To ensure compatibility with the YOLOv5 model, the background images are uniformly cropped to a standardized size of 640 x 640 pixels. This process is repeated multiple times to generate different combinations enhancing the diversity of the synthetic dataset. Fig. 3 presents some exemplary results of the method.

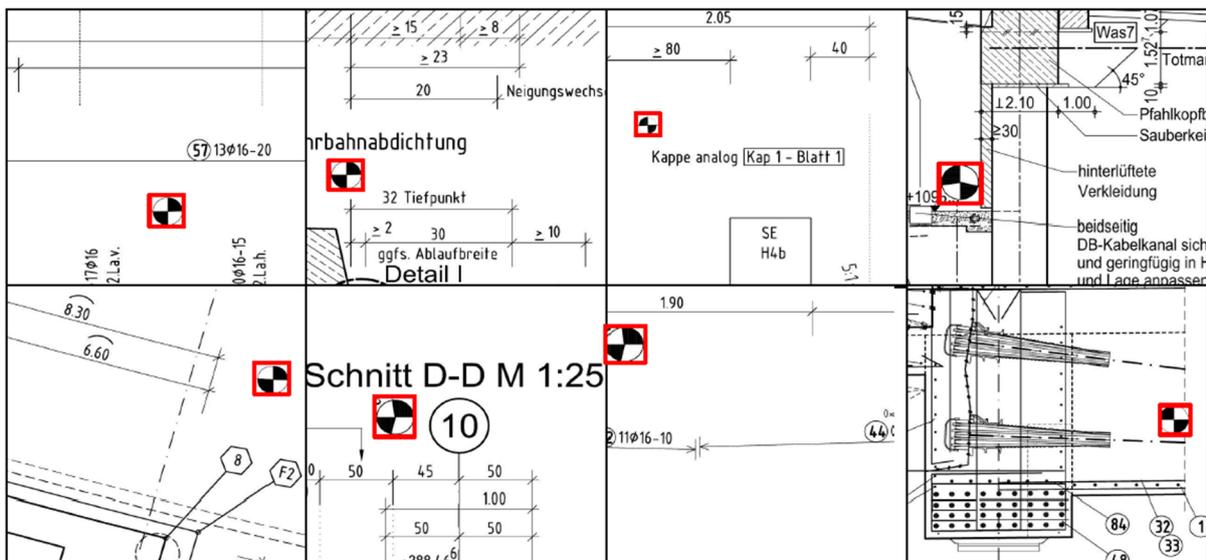


Fig. 3: Exemplary training images with a side length of 640 by 640 pixels synthetically generated using the proposed copy-paste method.

In total a number of 980 training images and 280 validation images are generated, resulting in the dataset named *Symbols*. Additionally, 62 real different images extracted from real bridge drawings are used for testing. According to Jocher et al. (2022) adding up to 10% of empty background images to the training and validation dataset enhances the object detection performance. Hence, 128 images are added to the training data, while 32 images are added to the validation data. This dataset is called *Symbols+BG*. Table 1 provides an overview of the final datasets.

Table 1: Overview of the generated data sets. Synthetic images are only used for training and validation.

	Content	No. of synth. training images	No. of synth. validation images	No. of real testing images
<i>Symbols</i>	Symbols only	980	280	62
<i>Symbols+BG</i>	Symbols and empty background	1108	312	62

3.2 Preprocessing and Gradient Symbol Detection

In order for the YOLOv5 model to handle the large image sizes of the input longitudinal views, a sliding window approach is employed. As shown in Fig. 4, as a first step a 640 by 640 pixel sized window is shifted across the image in increments of 330 pixels, ultimately covering the entire image. This step ensures that the gradient symbol is displayed in its entirety in at least one window, improving the detection performance. The trained YOLOv5 detector individually processes the windows, and the detected symbol is recorded. In cases where a symbol is detected in multiple overlapping windows, non-maximum suppression is employed to mitigate the possibility of double detections. For the final detections, rectangular regions are cropped from the original input image, extending 330 pixels in each direction (see Fig. 4), which is found to be sufficient. These cropped regions are further processed in the gradient parameter extraction, as explained in the following section.

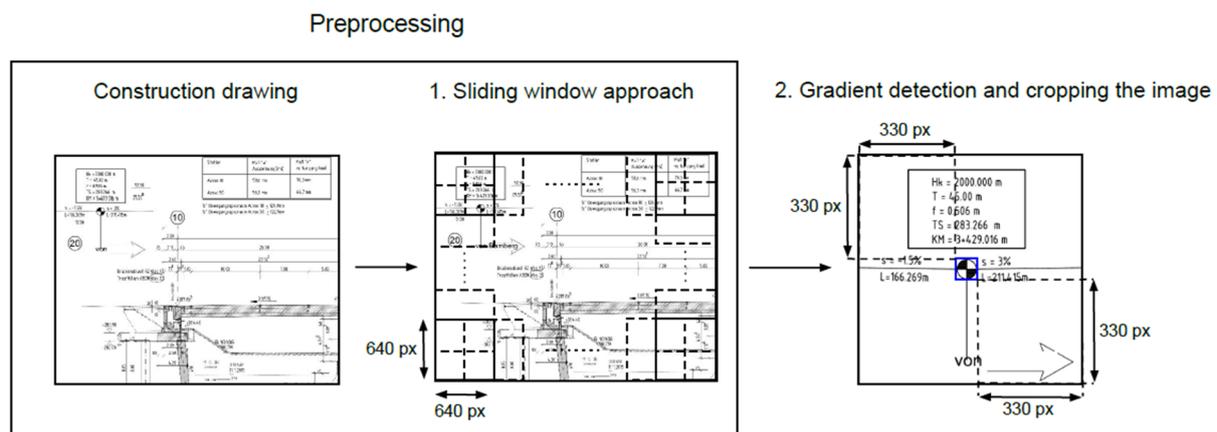


Fig. 4: Preprocessing of the construction drawing and detecting the gradient within the cropped area.

3.3 Gradient Parameter Extraction

After detecting the gradient symbols, the associated gradient parameters are extracted. To reliably recognize the parameters, the EasyOCR¹ text recognition model is used. EasyOCR combines the CRAFT (Baek et al. 2019) network and the text recognition network CRNN (Shi et al. 2018), since it is specifically designed for OCR. Therefore, EasyOCR provides robust capabilities for recognizing and extracting text from images. EasyOCR is employed in each region, as provided by the results from section 3.2. Within these regions, the text may appear rotated, reducing the EasyOCR model's recognition accuracy. To address this issue, the OCR engine is employed on the original region and a 90-degree rotated version. In addition, various image pre-processing techniques, e.g., adjustments to brightness and contrast, are applied to further improve the text recognition results.

An additional challenge is that not all text in the regions is relevant for the gradient reconstruction. Therefore, to ensure that only the relevant text is detected for further analysis, filtering, and string-matching techniques are implemented.

¹ EasyOCR: <https://github.com/JaidedAI/EasyOCR> (Accessed 1 August 2023).

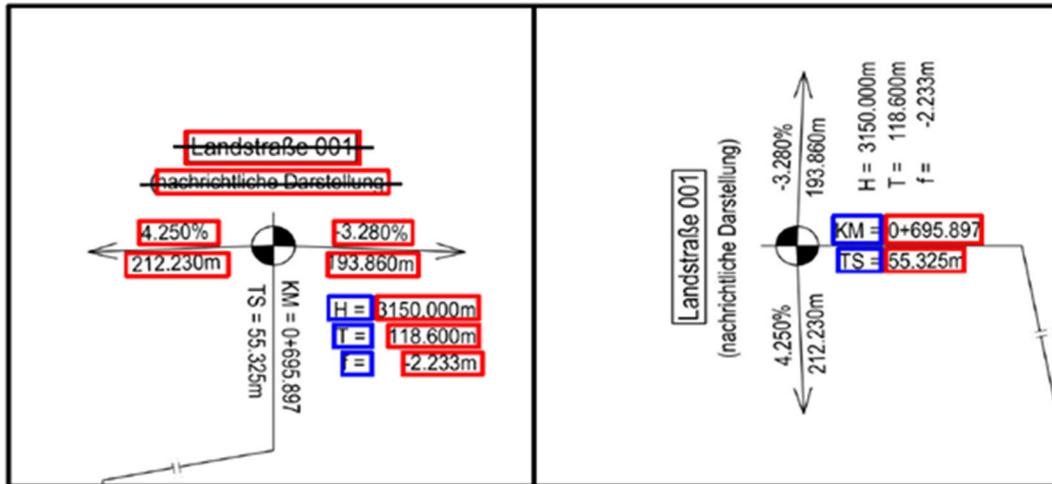


Fig. 5: Exemplary representation of the recognized texts in the original (left) and 90° rotated (right) patches. The recognized texts are marked in red, and the parameters are marked in blue. The crossed-out text passages cannot be assigned to any parameter and are, therefore, ignored.

In Fig. 5, the process of rotation, filtering, and string-matching is demonstrated. For instance, the text 'Landstraße 001' is irrelevant for the gradient detection and thus discarded. This filtering process is accomplished by identifying specific characters, such as KM , TS , or T , as depicted by blue in Fig. 5. Subsequently, the values that appear in the same horizontal line (cf. Fig. 5 marked with red rectangles) are matched with the identifying specific characters. In contrast, values located far away from the clusters are discarded. This process filters out unwanted text so that only relevant information is used in further analysis.

4. TRAINING & RESULTS

4.1 Training Process

Three different model sizes of YOLOv5 are trained and their performance is compared: YOLOv5m, YOLOv5l, and YOLOv5x. Each model trains on both the *Symbols* and *Symbols+BG* dataset using a NVIDIA A100 SXM4 40 GB graphics card. The models trained on Symbols are referred to as YOLOv5m, YOLOv5l, and YOLOv5x, while the ones trained on Symbols+BG are denoted as YOLOv5m_b, YOLOv5l_b, and YOLOv5x_b. During training a batch size of 56 is used for a maximum of 300 epochs. Backpropagation is performed using a learning rate of 0.01 with a stochastic gradient descent (SGD) optimizer (Robbins & Monro, 1951).

4.2 Symbol Detection Results

The YOLOv5 models are evaluated on 62 test images. Several key evaluation metrics are analyzed: precision, recall, intersection over union (IoU), mean average precision (mAP) at an IoU threshold of 0.50 (mAP@0.50), and mAP across IoU thresholds from 0.50 to 0.95 (mAP@0.50:0.95). The IoU measures the overlap between the predicted and ground truth bounding box. Precision measures the proportion of true positive detections out of all positive detections made by the model, indicating the accuracy of the model's predictions. On the other hand, recall represents the proportion of actual gradient symbols correctly identified by the models, measuring the model's ability to capture all instances of the target object. mAP@0.50 is a metric that enables the evaluation of the model's precision on average when there is at least a 50% IoU with the ground truth bounding boxes. In simpler terms, mAP@0.50 allows the assessment of how well the model performs in accurately detecting and localizing the gradient symbol, even when there is a moderate level of overlap between the predicted bounding boxes and the actual objects in the images. mAP@0.50:0.95 offers a more comprehensive evaluation by considering a range of IoU thresholds, enabling a better understanding of the model's performance across different levels of overlap. Since the gradient symbols are relatively small, this study considers an IoU of 50% sufficient.

The test results of different trained YOLOv5 models are shown in Table 2. It can be observed that the models achieve high precision scores, indicating their capability to accurately detect the gradient symbol in the real image dataset. Moreover, the models show varying levels of recall. YOLOv5m archives the highest recall score of 0.957, while YOLOv5l scores the lowest value of 0.882. Overall, the models successfully identify the gradient symbols

but with some variation in recall performance.

Table 2: Gradient detection results for different models. The bold fonts indicate the best results.

	YOLOv5m	YOLOv5l	YOLOv5x	YOLOv5m_b	YOLOv5l_b	YOLOv5x_b
Precision	0.998	1	0.984	1	0.999	0.999
Recall	0.957	0.882	0.906	0.913	0.942	0.957
$mAP_{0.50}^{IoU}$	0.965	0.942	0.966	0.958	0.977	0.976
$mAP_{0.50:0.95}^{IoU}$	0.878	0.865	0.882	0.874	0.89	0.894

For the $mAP@0.50$ all models demonstrate strong performance. YOLOv5l_b achieves the highest $mAP@0.50$ of 0.977, followed closely by YOLOv5x_b with 0.976. YOLOv5l achieves the lowest $mAP@0.50$ score of 0.942. Considering the $mAP@0.50:0.95$ all models have consistently high scores. YOLOv5x_b achieves the highest score of 0.894, closely followed by YOLOv5l_b with 0.89. YOLOv5l shows the lowest $mAP@0.50:0.95$ score of 0.865. Overall, the models indicate a good performance across the range of IoU thresholds. Considering the detection speed and $mAP@0.50$ performance YOLOv5l_b is selected as the best model for this application. Some detection results of gradient symbols are presented in Fig. 6.

4.3 Overall Pipeline Results

The models successfully detect the symbols despite variations such as rotation and partial view. To assess the performance of the overall pipeline, it is tested on four real longitudinal views, each containing multiple gradient symbols. To evaluate the OCR accuracy, a character-level analysis is performed. The recognized characters extracted by EasyOCR are compared to the expected gradient parameters associated with each drawing. The accuracy is then calculated by determining the percentage of correctly recognized parameters relative to the total number of parameters present.

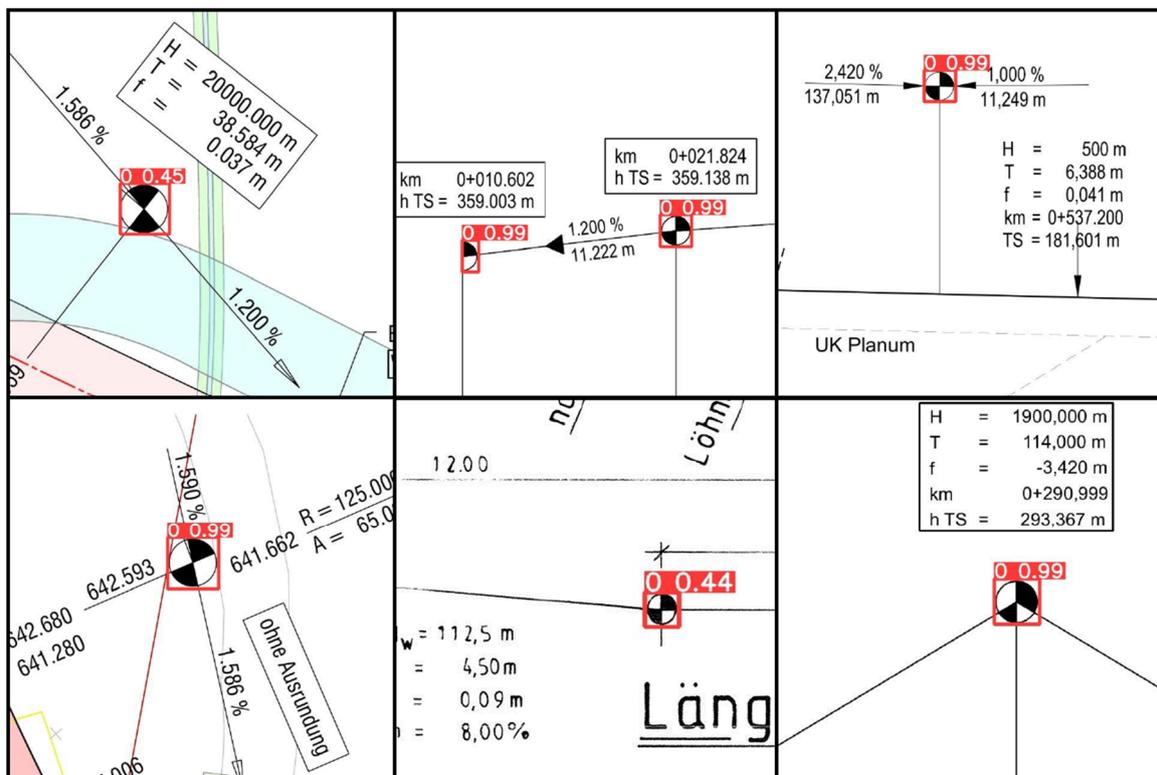


Fig. 6: Example results of gradient symbol detection.

The evaluation results reveal an overall accuracy of 70%. This shows that the pipeline can successfully extract gradient parameters from most of the drawings. However, there is room for improvement, especially in cases where parameter recognition becomes difficult due to variations in fonts and image quality.

5. DISCUSSION & CONCLUSION

The results of this study demonstrate the effectiveness of the proposed multistage pipeline for automatically extracting bridge gradient information from pixel-based construction drawings using deep learning techniques. The YOLOv5 models trained on different datasets showed high precision scores on the test dataset, indicating their capability to detect gradient symbols accurately. The results indicate that training the models on the synthetic dataset is beneficial to overcome a lack of data. The addition of background images has improved the overall performance of the models.

The evaluation metrics, including mAP@0.50 and mAP@0.50:0.95, revealed strong performance across all models. Notably, YOLOv5l_b achieved the highest mAP@0.50 score with 98%, making it the most suitable model for this application considering detection speed and accuracy. The detection results of gradient symbols further illustrate the pipeline's ability to successfully identify symbols despite variations in rotation and partial views. The pipeline's overall performance in extracting gradient parameters from real bridge drawings is promising, with an OCR accuracy of approximately 70%. However, there is still room for improvement, especially in cases with challenging text recognition due to varying fonts, and image quality.

In conclusion, the proposed pipeline presents an effective approach for the retro-digitization of existing bridges, significantly reducing the time and effort required for this crucial task. The successful extraction of gradient information from construction drawings holds great potential for improving bridge asset management and maintenance planning. For future research, fine-tuning, and optimization of the OCR component could further enhance accuracy and pave the way for broader applications.

REFERENCES

- Akanbi, T., & Zhang, J. (2022). Semi-Automated Generation of 3D Bridge Models from 2D PDF Bridge Drawings. In F. Jazizadeh, T. Shealy, & M. J. Garvin (Eds.), *Construction Research Congress 2022* (pp. 1347–1354). <https://doi.org/10.1061/9780784483961.141>
- Baek, Y., Lee, B., Han, D., Yun, S. and Lee, H. (2019). Character Region Awareness for Text Detection. *In the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19365-19374. <https://doi.org/10.1109/CVPR.2019.00959>
- Borrmann, A., König, M., Koch, C., & Beetz, J. (2018). Building Information Modeling: Why? What? How? In A. Borrmann, M. König, C. Koch & J. Beetz (Eds.), *Building Information Modeling*. Springer. https://doi.org/10.1007/978-3-319-92862-3_1
- Durdyev, S., Ashour, M., Connelly, S., & Mahdiyar, A. (2022). Barriers to the implementation of Building Information Modelling (BIM) for facility management. *Journal of Building Engineering*, 46, Article 103736. <https://doi.org/10.1016/j.jobbe.2021.103736>
- Faltin, B., Schönfelder, P., & König, M. (2023). Inferring Interconnections of Construction Drawings for Bridges Using Deep Learning-based Methods. In E. Hjelseth, S. F. Suján & R. J. Scherer (Eds.), *ECPPM 2022-eWork and eBusiness in Architecture, Engineering and Construction 2022* (pp. 343-350). CRC Press. <https://doi.org/10.1201/9781003354222>
- Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., TaoXie, Fang, J. imyhxy, Lorna, Zan Yifu, Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, ... Jain, M. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. Zenodo. <https://doi.org/10.5281/zenodo.3908559>
- Kim, S., Park, S., Kim, H., & Yu, K. (2021). Deep Floor Plan Analysis for Complicated Drawings Based on Style Transfer. *Journal of Computing in Civil Engineering*, 35(2), Article 04020066. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000942](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000942)
- Mafipour, M. S., Ahmed, D., Vilgertshofer, S., & Borrmann, A. (2023). Digitalization of 2D Bridge Drawings Using Deep Learning Models. *The 30th EG-ICE: International Conference on Intelligent Computing in Engineering*. https://www.ucl.ac.uk/bartlett/construction/sites/bartlett_construction/files/digitalization_of_2d_bridge_drawings_using_deep_learning_models.pdf
- Moreno-García, C. F., Elyan, E., & Jayne, C. (2019). New trends on digitisation of complex engineering drawings.

Neural computing and Applications, 31, 1695-1712. <https://doi.org/10.1007/s00521-018-3583-1>

Poku-Agyemang, K. N., & Reiterer, A. (2023). 3D Reconstruction from 2D Plans Exemplified by Bridge Structures. *Remote Sensing*, 15(3), 677. <https://doi.org/10.3390/rs15030677>

Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3), 400–407. <http://www.jstor.org/stable/2236626>

Schönfelder, P., Aziz, A., Faltin, B., & König, M. (2023). Automating the retrospective generation of As-is BIM models using machine learning. *Automation in Construction*, 152, Article 104937. <https://doi.org/10.1016/j.autcon.2023.104937>

Shi, B., Bai, X., & Yao, C. (2017). An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11), 2298–2304. <https://doi.org/10.1109/TPAMI.2016.2646371>

Wei, C., Gupta, M., & Czerniawski, T. (2022). Automated Wall Detection in 2D CAD Drawings to Create Digital 3D Models. *Proceedings of the 39th International Symposium on Automation and Robotics in Construction* (pp. 152-158). IAARC Publications. <https://doi.org/10.22260/ISARC2022/0023>

Zhao, Y., Deng, X., & Lai, H. (2020). A Deep Learning-Based Method to Detect Components from Scanned Structural Drawings for Reconstructing 3D Models. *Applied Sciences*, 10(6), 2066. <https://doi.org/10.3390/app10062066>

Zhao, Y., Deng, X., & Lai, H. (2021). Reconstructing BIM from 2D structural drawings for existing buildings. *Automation in Construction*, 128, Article 103750. <https://doi.org/10.1016/j.autcon.2021.103750>