Automated Extraction of Structural Elements and Section Views from Bridge Drawings Using Deep Learning

Hakan Bayer[1] **and Markus König**[1]

[1]Chair of Computing in Engineering, Ruhr University Bochum, Bochum, Germany

hakan.bayer@ruhr-uni-bochum.de, koenig@inf.bi.rub.de

Abstract -

Technical drawings are essential for managing and maintaining existing bridges, especially in creating accurate digital models and integrating them into Building Information Modeling (BIM) workflows. However, digitizing older bridges, which often lack BIM models is challenging due to the time-consuming and error-prone process of manual data extraction from 2D drawings. Therefore, this study focuses on detecting key structural components, such as anchors, tendons, and section views, which are critical for structural analysis, maintenance, and retrofitting. Using state-of-the-art object detection models, YOLOv8 and YOLOv11, the proposed pipeline achieves a mean average precision (mAP) of 97.4% for anchors, 95.7% for tendons, and 63.9% for section views. Optical Character Recognition (OCR) tool Pytesseract further enhances the process by accurately extracting IDs for anchors and tendons with 90% precision. Combining object detection with precise text extraction simplifies the creation of high-quality digital bridge models and improves BIM integration. Specifically, the extracted data provides semantic information on key structural components. This information can facilitate the process of creating a digital model and integrating it into BIM workflows.

Keywords -

Building Information Modeling; Computer Vision; Deep Learning; Symbol Detection; Optical Character Recognition; Construction Drawings

1 Introduction

Bridges are a crucial component of transportation infrastructure that enables the transportation of people and goods. Maintaining their functionality and ensuring safe operations requires routine inspections and effective maintenance management. Digital tools like Building Information Modelling (BIM) can enhance maintenance planning efficiency. BIM is a collaborative digital methodology that involves creating and sharing interdisciplinary digital models of structures [1]. These models contain semantic and geometric information, sharing the state of the project in real time.

Among all phases of a project's life cycle, the operations and maintenance (O&M) phase presents significant opportunities for BIM application [2]. A key challenge however lies in the limited availability of accurate digital building models for existing structures [3]. In many cases, this is because the old constructions were designed and constructed without the implementation of BIM. Consequently, it is necessary to develop these models in retrospect.

The digitization of existing high-rise buildings and infrastructure is particularly difficult, requiring the expertise of architects and engineers to reconstruct geometric and semantic information from various sources such as point clouds, images, and construction drawings [4]. While point clouds and images are valuable resources, they often rely on costly equipment and fail to reveal internal structural details. In contrast, construction drawings offer a more accessible and practical alternative, especially for older bridges where they are often the only available source of information. As a result, construction drawings are frequently the preferred choice for creating digital building models.

Extracting key information from bridge construction drawings, including section views, structural elements, and their identifiers, is a crucial initial step in developing accurate digital models. These drawings often feature critical internal components like anchors, tendons, and section views, which are essential for understanding bridges' structural integrity and functionality. Anchors for instance are vital elements in prestressed concrete bridges used to secure tendons and transfer tensile forces to the surrounding concrete. Tendons act as reinforcement ensuring the bridge can handle heavy loads and maintain stability. On the other hand, section views offer detailed representations of inner components enabling engineers to understand their precise placement and spatial relationships within the bridge structure.

This research focuses on automating the extraction of these critical components, including section views, anchors, tendons, and their identifiers from CAD based bridge technical drawings to support the development of accurate digital models. Using state-of-the-art deep learning frameworks, such as YOLOv8 [5] and YOLOv11 [6] in combination with OCR tools like PyTesseract [7], the proposed method achieves precise detection of these components and extracts relevant semantic information. This output data facilitates the creation of digital bridge models, enabling better integration into BIM workflows hence supporting more efficient bridge maintenance and management.

2 Related Work

In deep learning and computer vision, significant attention has been given to reconstructing high-rise building models from technical drawings, with a primary focus on detecting and reconstructing external elements such as walls, rooms, and openings [8]. For instance, [8] utilized ResNet to identify walls in floor plans, while [9] leveraged conditional GANs (cGANs) to generate heat maps and vectorized representations of building layouts. These approaches have proven effective for creating digital building models from architectural drawings.

However, the digitization of technical drawings, particularly for infrastructures such as bridges has received limited attention in research [10]. Although some studies are exploring semi-automated analysis of such drawings, the field remains underdeveloped. [11] proposed a semi-automated process employing the Douglas-Peucker algorithm to detect corner points of illustrated components. These corner points are then used to reconstruct the external geometry of the components, which ultimately generates the structural outline of the bridge. [12] introduced a different semi-automated framework that converts drawings into vector-based formats. This process aligns the extracted views and uses the gathered data to reconstruct the geometry of the bridge. The resulting digital model is then exported in the IFC (Industry Foundation Classes) format for further use.

More recently, deep learning-based approaches have been explored in this domain. [13] presented a pipeline using YOLOv5 and the CRAFT text detector to identify individual components and labels on technical drawings. The detected objects were clustered based on their labels to account for multiple views, with experts manually reviewing the clustered data to create bridge models. Similarly, [14] employed Faster R-CNN to detect section symbols and applied OCR to extract view titles, linking symbols to their corresponding views and establishing interconnections across drawing sets. [15] utilized YOLOv5 for gradient symbol detection and integrated EasyOCR for text extraction, automating the process of extracting bridge gradients from construction drawings. Additionally, the study by [16] introduced a computer vision-based process for generating bridge superstructure models from pixel-based construction drawings. Their approach leverages Keypoint R-CNN for symbol pose estimation, enabling the spatial organization of drawing views and facilitating the development of accurate digital bridge models.

Despite these advancements, no study to date has addressed the automated extraction of inner structural components of bridges, such as anchors, tendons, and section views, from pixel-based technical drawings. This study fills this gap by employing YOLOv8 [5] and YOLOv11 [6] to detect these critical components and PyTesseract [7] to extract their associated IDs. These advancements enable the digitization of bridge inner structures and facilitate the creation of digital models for integration into BIM workflows.

3 Methodology

This study presents a semi-automated pipeline for detecting anchors, tendons, and section views in CAD based 2D bridge technical drawings, as well as extracting associated IDs for anchors and tendons. The methodology integrates classical computer vision techniques, deep learning-based object detection, and OCR for text recognition. The workflow, as illustrated in Figure 1, consists of three main stages: (1) Object Detection, (2) Shape Detection, and (3) Text Extraction.

As shown in Figure 1, the pipeline begins with a pixel-based bridge drawing input, which contains cross-sectional representations of internal structural components such as anchors, tendons, and section views. The first stage involves object detection, where elements such as anchors, tendons, and section views are detected using an object detection model. In the second stage, shape detection was performed using OpenCV to locate ID boxes by detecting rectangular regions related to anchor and tendon elements. Finally, the text extraction stage employs OCR to extract the ID numbers from the identified rectangle regions. The pipeline outputs the detected section views, anchors, tendons, and associated IDs, enabling efficient process and analysis of bridge drawings.

3.1 Dataset Creation and Annotation

Due to the limited availability of real bridge technical drawings, a combination of real and synthetic data was used to create the dataset. The original dataset comprised 62 internal structural drawings of bridges from Germany. These drawings were divided into sections, resulting in 71 images. Of these, 42 images were designated for training and validation, while 29 were set aside for testing the pipeline. The 42 training-validation images were particularly large, with an average width of 11,290 pixels and an average height of 7,744 pixels.

Since the 42 training-validation images were insufficient for effective model training, synthetic data was generated by overlaying cropped symbols (anchors, tendons, and section views) onto real construction drawing backgrounds that did not initially contain these symbols using copy-paste method. This approach ensured a diverse and robust dataset for training and validation.

The train validation dataset consists of three primary components: anchor element dataset, tendon element dataset, and section view dataset. The anchor element dataset includes two classes: longitudinal anchor sections

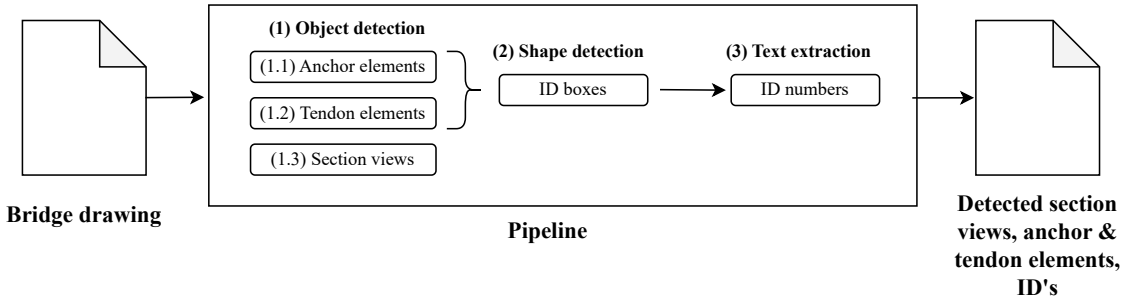

Figure 1. Workflow diagram illustrating the detection of anchors, tendons, and section views in bridge drawings, along with the extraction of associated IDs.

and front anchor sections. Similarly, the tendon dataset contains only one class which is front tendon sections. The section views dataset contains four types of classes: longitudinal section view, front sectional view of the final anchor point, cross-section view, and longitudinal section view of the final anchor point. Some example annotated objects of the dataset classes are shown in Fig. 2. These annotations capture the variability and complexity of the internal structural drawings.

3.1.1 Anchor Dataset

From the real training-validation dataset, 204 longitudinal anchor symbols and 74 front anchor symbols were selected and cropped out. The longitudinal anchor symbols had an average width of 292 pixels and an average height of 163 pixels, while the front anchor symbols had an average width of 93 pixels and an average height of 95 pixels. These symbols were randomly rotated (0°, 90°, -90°) and overlaid onto real construction drawing backgrounds to create synthetic images. This process generated 500 synthetic labeled images, which were automatically annotated in YOLO bounding box format. This dataset was then combined with the 42 annotated real bridge drawings, which created final anchor dataset of 542 images. The dataset was then split into 80% training and 20% validation sets.

3.1.2 Tendon Dataset

The tendon dataset was created using 53 tendon front sectional views extracted from the real training-validation dataset. These tendon elements had an average width of 41 and height of 41 pixels. These cropped symbols were randomly positioned on real construction drawing backgrounds, and their locations and dimensions were automatically labeled in YOLO bounding box format. Similar to the anchor dataset, this process generated 500 synthetic labeled images, resulting in a final tendon dataset of 542 images, which was also split into 80% training and 20% validation sets.

3.1.3 Section View Dataset

The section view dataset includes four types of classes: longitudinal section view, front section view of the final anchor point, cross-section view , and longitudinal section view of the final anchor point. The longitudinal section view depicts the internal arrangement of the entire bridge along its length, while the front section view of the final anchor point focuses on the final anchor point and its components from a frontal perspective. Cross-section view shows vertical cuts along the bridge's width, highlighting structural details, and longitudinal section view of the final anchor point provides a longitudinal perspective of the anchor components. The section views are pixel wise relatively large compared to tendon and anchor elements. Due to their complexity, size, and variability, huge number of synthetic data was generated compared to the other datasets. CAD files of the training and validation set were used to generate additional data by modifying attributes such as size, perspective, and surrounding text. Also a copy-paste method similar to that used for the anchor and tendon datasets was employed. In total, 1,842 images were generated including 42 annotated real bridge drawings. All images were automatically labeled using the YOLO bounding box format. Then the final dataset was divided into 80% training and 20% validation sets.

3.2 Object Detection

The detection pipeline utilizes multiple YOLO architectures, YOLOv8 [5] and YOLOv11 [6], trained on separately annotated datasets for anchor elements, tendon elements, and section views. Each dataset was trained on models of varying scales, such as medium and large configurations. The inference image size was chosen to match the respective training image size to maximize accuracy.

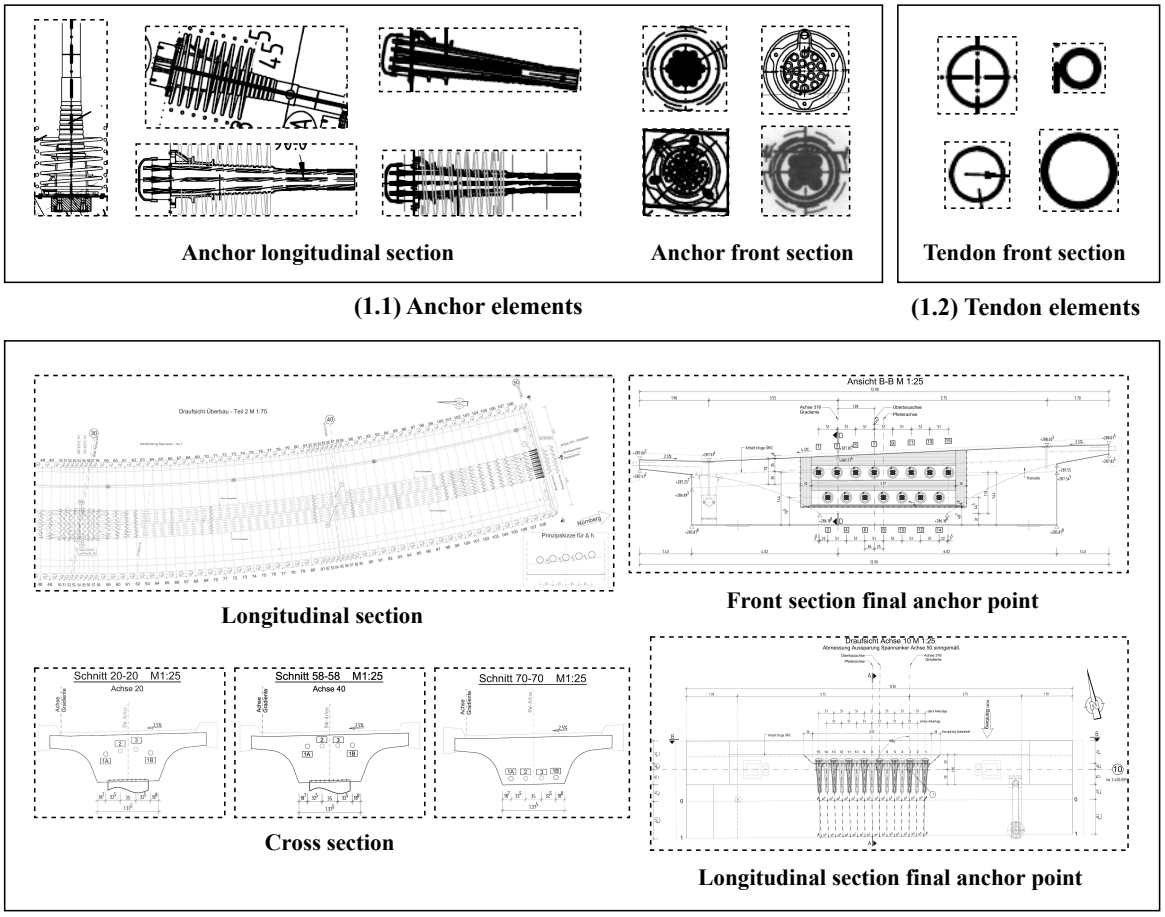

Figure 2. Examples of annotations from the dataset, highlighting anchor elements (1.1) and tendon elements (1.2), along with 4 different section views (1.3). These include longitudinal section views, cross-sections, front section views of the final anchor point, and longitudinal section views of the final anchor point. These annotations are crucial for training object detection models and generating synthetic data.

After training, the best models were integrated into the detection pipeline to accurately detect objects.

3.3 Shape Detection

The shape detection stage focuses on identifying rectangular regions in the drawings that likely contain IDs associated with anchors and tendons. This is achieved using classical computer vision techniques implemented with OpenCV. The process starts by converting the input images to grayscale, followed by applying Gaussian blur to reduce noise and enhance edge details. Adaptive thresholding is then applied to create a binary image, which simplifies edge detection.

Contours are extracted from the binary image to detect potential rectangular shapes. These shapes are filtered based on specific criteria, such as their area and aspect ratio, to isolate valid ID boxes. During this stage, we assume that IDs are enclosed within rectangular shapes based on the standard drawing conventions of a specific drawing format. It is important to note that some firms may use different drawing conventions, such as enclosing IDs within circular or elliptical shapes. Our pipeline, however, is made to function with rectangular ID boxes following the conventional norms utilized in these particular drawings for accuracy and consistency.

3.4 Text Extraction

The text extraction stage employs PyTesseract to extract textual identifiers from the ID boxes identified during the shape detection process. To enhance the precision of text recognition, the images are preprocessed prior to applying OCR. This preprocessing involves converting images to grayscale and using adaptive thresholding to improve contrast and text clarity within the ID boxes. To prevent errors caused by extracting text from unrelated elements in the drawings, the OCR process is restricted to a predefined character set that includes numbers, letters, hyphens, slashes, and periods. This restriction ensures that only valid identifier formats are captured, minimizing the inclusion of irrelevant or incorrect text. Once the text is extracted, it is mapped to the corresponding tendon or anchor elements based on the spatial relationship between the detected text and ID boxes. As section views do not include associated identifiers, text extraction is performed exclusively for tendon and anchor elements.

4 Training and Results

4.1 Training Process

The training process utilized eight NVIDIA A100 GPUs. The datasets for anchors, tendons, and section views were trained independently due to their unique characteristics, varying object sizes, and complexities. To evaluate detection performance, input image sizes were adjusted based on the relative scale and complexity of the objects in each dataset.

Both YOLOv8 [5] and YOLOv11 [6] models were trained for 100 epochs on each dataset, using two different image resolutions to compare the impact of varying sizes on model performance. Additionally, two model configurations, medium and large, were employed. Higher resolutions were used for the tendon and anchor datasets, as these objects are relatively small compared to the overall size of the drawings. In contrast, moderate resolutions were applied to the section view dataset, which contains larger objects. Specifically, the tendon dataset was trained with image sizes of 3200 and 3600 pixels, the anchor dataset with 2560 and 3200 pixels, and the section view dataset with 640 and 1280 pixels.

This setup allowed the models to effectively capture the distinct features of each dataset, enabling robust detection performance and ensuring adaptability to the diverse complexities of technical bridge drawings.

4.2 Object Detection Results

The performance of the YOLOv8 and YOLOv11 models was evaluated across three datasets: section view, anchor, and tendon, using a testing set of 29 unseen images. Evaluation metrics included Precision (P), Recall (R), and mean Average Precision at 50% IoU threshold (mAP50). Table 1 summarizes the results for each dataset and model configuration.

For the anchor dataset, the YOLOv11 large model achieved an impressive overall mAP50 of 0.974, with a Precision of 0.917 and a Recall of 0.970. Among the specific classes, the Anchor front section class achieved the highest mAP50 of 0.994, demonstrating near-perfect detection accuracy. The Anchor longitudinal section class also showed strong performance, with a mAP50 of 0.954.

In the tendon dataset, the YOLOv11 large model delivered a mAP50 of 0.957, with a Precision of 0.914 and a Recall of 0.952. The use of high-resolution images during training and testing allowed the model to effectively capture and detect the fine-grained details of small and elongated objects like tendons.

For the section view dataset, the YOLOv11 large model achieved an overall mAP50 of 0.639, with a Precision of 0.712 and a Recall of 0.623. However, performance varied significantly across individual classes. The Cross-section class achieved the highest mAP50 of 0.858, demonstrating robust detection capabilities. In contrast, the Longitudinal section and Longitudinal section final anchor point classes showed lower mAP50 values of 0.436 and 0.565, respectively. The Front section final anchor point class achieved a moderate mAP50 of 0.697. The lower performance in section view detection can be explained due to the high variability in section drawings and limited real dataset samples. Unlike anchors and tendons, section views differ significantly across drawings making object detection more challenging.

Overall, the YOLOv11 large model consistently outperformed YOLOv8 across all datasets, demonstrating superior accuracy and robustness in handling diverse object detection tasks. The results also highlight the importance of using appropriately high-resolution images during training and testing particularly for datasets with smaller objects as it significantly enhances detection performance.

4.3 Overall Pipeline Results

The proposed pipeline demonstrates strong performance in detecting section views, anchor and tendon symbols, and extracting IDs even when faced with challenges such as rotation, scale differences, and partial visibility. Testing on 29 real images showed that the pipeline reliably identified anchors, tendons, and section views, achieving high object detection accuracy. However, text extraction faced some issues, particularly when shapes were rotated, making it harder to identify IDs. As outlined in the *Shape Detection* section, the pipeline's shape detection relies on drawings adhering to standard conventions such as rectangular ID boxes commonly used by a specific firm. Based

Table 1. Object Detection Results of YOLOv8 and YOLOv11 Models on Test Dataset (29 Images) for Section View, Anchor, and Tendon Datasets with Different Image Sizes. Metrics include Precision (P), Recall (R), and mAP50. The table provides the average performance across all object classes within each dataset, offering an overall assessment of detection accuracy. Bold fonts indicate the best results for each metric.

Dataset	Image Size	Model	P	R	mAP50
Anchor Dataset	2560	YOLOv8m	0.952	0.927	0.964
	3008	YOLOv8m	0.958	0.942	0.970
	2560	YOLOv8l	0.958	0.893	0.949
	3008	YOLOv8l	0.934	0.954	0.971
	2560	YOLOv11m	**0.960**	0.927	0.955
	3008	YOLOv11m	0.955	0.947	**0.974**
	2560	YOLOv11l	0.952	0.933	0.957
	3008	YOLOv11l	0.917	**0.970**	**0.974**
Tendon Dataset	3200	YOLOv8m	0.899	0.837	0.895
	3600	YOLOv8m	0.925	0.902	0.936
	3200	YOLOv8l	0.914	0.865	0.934
	3600	YOLOv8l	0.928	0.895	0.930
	3200	YOLOv11m	0.900	0.866	0.915
	3600	YOLOv11m	0.922	**0.944**	0.959
	3200	YOLOv11l	**0.932**	0.887	0.943
	3600	YOLOv11l	0.914	0.952	**0.957**
Section View Dataset	640	YOLOv8m	**0.813**	0.546	0.590
	1280	YOLOv8m	0.566	0.544	0.484
	640	YOLOv8l	0.578	0.5 60	0.535
	1280	YOLOv8l	0.519	0.618	0.529
	640	YOLOv11m	0.702	0.615	0.585
	1280	YOLOv11m	0.634	0.595	0.535
	640	YOLOv11l	0.528	0.590	0.555
	1280	YOLOv11l	0.712	**0.623**	**0.639**

on this assumption the text detection accuracy was calculated by determining the percentage of correctly recognized parameters relative to the total number of parameters present. The pipeline achieved 90% accuracy in text extraction, demonstrating its ability to extract IDs in most cases. However, performance was hindered by rotated text boxes and slight variations in font styles and sizes. Future refinements could include pre-processing techniques like rotation correction and fine-tuning the model for the dataset to enhance recognition accuracy.

An example of the pipeline's performance is illustrated in Fig. 3, where it successfully detects objects and extracts text from real test drawings. These results highlight the pipeline's ability to support the automatization process of digitization of technical drawings and extract critical information, making it a valuable tool for further applications in bridge modeling.

5 Discussion and Conclusion

This study presents a semi-automated pipeline for extracting structural components and their identifiers from bridge technical drawings, addressing the challenges of digitizing older structures. By integrating deep learning-based object detection models, YOLOv8 and YOLOv11, with the optical character recognition tool PyTesseract, the pipeline achieves high accuracy in identifying and extracting critical elements and their associated information.

The YOLOv11 large models demonstrated the best performance, achieving mAP50 scores of 0.974 for anchor elements and 0.957 for tendon elements. Detection of section views showed moderate success, with an mAP50 of 0.639. Text extraction using PyTesseract achieved 90% accuracy. However, the pipeline was tested on a limited number of drawings following specific rectangular ID standards, which restricts its generalizability. This limitation highlights the need to generalize the approach for broader applicability. Additionally, the assumption of rectangular ID boxes restricts generalizability to other formats such as circular or elliptical IDs. Furthermore, OCR struggles with rotated or noisy text which could be improved with advanced preprocessing techniques.

Despite these achievements, challenges remain particularly in text recognition and adapting the pipeline to diverse drawing standards. Future research could focus on fine-tuning the OCR model for bridge drawings, expanding shape detection to accommodate circular or elliptical ID boxes, and increasing the range of synthetic training datasets. Exploring alternative detection models may also enhance performance, especially for complex components like section views.

Overall, this study presents a solution for automating the digitization of bridge technical drawings. The extracted section views and elements, along with their IDs, can serve as structured inputs for BIM software enabling efficient reconstruction. The proposed method streamlines the creation of accurate digital models and facilitates their integration into BIM workflows.

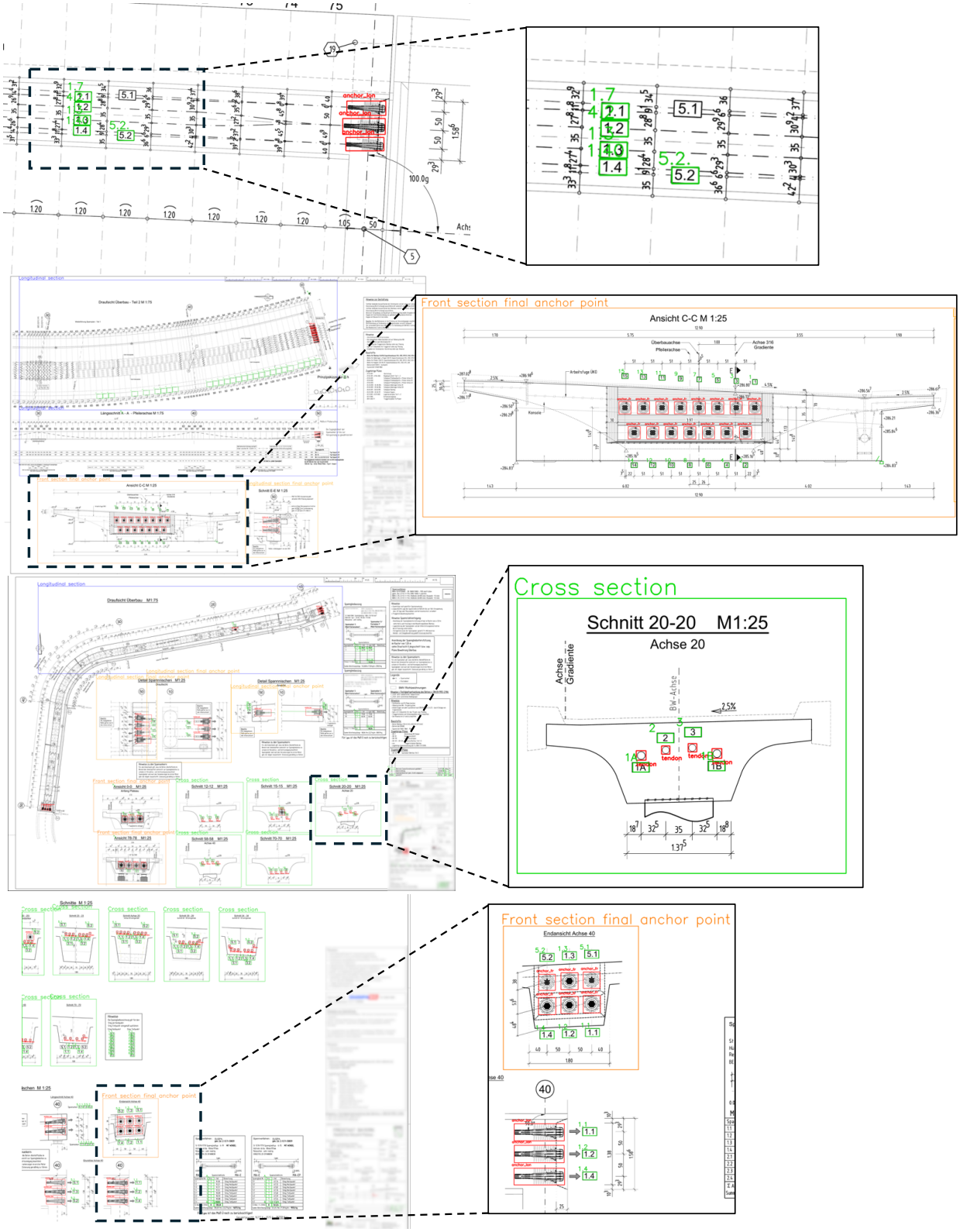

Figure 3. Demonstration of the pipeline's performance on a real test dataset using the best-performing models with the highest mAP50 values. The figure showcases the accurate detection of section views (e.g., cross sections, front section final anchor points), identification of symbols, and precise extraction of their associated IDs. Enlarged views highlight specific areas of successful object detection and ID extraction, underscoring the pipeline's effectiveness in processing complex engineering drawings.

References

[1] André Borrmann, Markus König, Christian Koch, and Jakob Beetz. Building Information Modeling – Why? What? How? In *Building Information Modeling - Technology Foundations and Industry Practice*, pages 1–24. Springer, 09 2018. doi:10.1007/978-3-319-92862-3.

[2] Dania Abideen, Akilu Yunusa-Kaltungo, Patrick Manu, and Clara Cheung. Digitalization of construction life cycle: A systematic review of building and reliability information modelling (BRIM). In *AIP Conference Proceedings*, volume 2428, page 020001, 11 2021. doi:10.1063/5.0070961.

[3] Xu Yiye and Yelda Turkan. BrIM and UAS for bridge inspections and management. *Engineering, Construction and Architectural Management*, 27: 785–807, 2019. doi:10.1108/ecam-12-2018-0556.

[4] Phillip Schönfelder, Angelina Aziz, Benedikt Faltin, and Markus König. Automating the retrospective generation of as-is BIM models using machine learning. *Automation in Construction*, 152:104937, 2023. doi:10.1016/j.autcon.2023.104937.

[5] Glenn Jocher, Ayush Chaurasia, Jiankai Qiu, and Anthony Stoken. Yolov8: State-of-the-art object detection models. Online: `https://docs.ultralytics.com/models/yolov8/`, 2023. Accessed: 26/03/2025.

[6] Glenn Jocher, Ayush Chaurasia, Jiankai Qiu, and Anthony Stoken. Yolov11: State-of-the-art object detection model. Online: `https://docs.ultralytics.com/models/yolo11/`, 2024. Accessed: 26/03/2025.

[7] Samuel Hoffstaetter. Pytesseract: Python wrapper for google tesseract-ocr. Online: `https://github.com/madmaze/pytesseract`, 2023. Accessed: 26/03/2025.

[8] Chialing Wei, Mohit Gupta, and Thomas Czerniawski. Automated wall detection in 2D cad drawings to create digital 3D models. In *Proceedings of the 39th International Symposium on Automation and Robotics in Construction*, pages 152–158, 2022. doi:10.22260/ISARC2022/0023.

[9] Seongyong Kim, Seula Park, Hyunjung Kim, and Kiyun Yu. Deep floor plan analysis for complicated drawings based on style transfer. *Journal of Computing in Civil Engineering*, 35(2):04020066, 2021. doi:10.1061/(ASCE)CP.1943-5487.0000942.

[10] C. F. Moreno-García, Eyad Elyan, and Chrisina Jayne. New trends on digitisation of complex engineering drawings. *Neural Computing and Applications*, 31:1695–1712, 6 2019. doi:10.1007/s00521-018-3583-1.

[11] Kwasi Nyarko Poku-Agyemang and Alexander Reiterer. 3D reconstruction from 2D plans exemplified by bridge structures. *Remote Sensing*, 15, 2023. doi:10.3390/rs15030677.

[12] Temitope Akanbi and Jiansong Zhang. Semi-automated generation of 3D bridge models from 2D PDF bridge drawings. In *Construction Research Congress 2022*, pages 1347–1354, 2022. doi:10.1061/9780784483961.141.

[13] M. Saeed Mafipour, Daniyal Ahmed, Simon Vilgertshofer, and Andre Borrmann. Digitalization of 2D bridge drawings using deep learning models. In *The 30th EG-ICE: International Conference on Intelligent Computing in Engineering*, 2023. URL `https://www.ucl.ac.uk/bartlett/construction/sites/bartlett_construction/files/digitalization_of_2d_bridge_drawings_using_deep_learning_models.pdf`.

[14] Benedikt Faltin, Phillip Schönfelder, and Markus König. Inferring interconnections of construction drawings for bridges using deep learning-based methods. In *ECPPM 2022-eWork and eBusiness in Architecture, Engineering and Construction 2022*, volume 62, pages 343–350, 2023. doi:10.1201/9781003354222-44.

[15] Hakan Bayer, Benedikt Faltin, and Markus König. Automated extraction of bridge gradient from drawings using deep learning. In *CONVR 2023 - Proceedings of the 23rd International Conference on Construction Applications of Virtual Reality*, 2023. doi:10.36253/979-12-215-0289-3.68.

[16] Benedikt Faltin, Phillip Schönfelder, Damaris Gann, and Markus König. Reconstructing as-built beam bridge geometry from construction drawings using deep learning-based symbol pose estimation. *Advanced Engineering Informatics*, 62:102808, 2024. doi:10.1016/j.aei.2024.102808.