Digitalization of 2D Bridge Drawings Using Deep Learning Models

M. Saeed Mafipour[1*], Daniyal Ahmed[1], Simon Vilgertshofer[1], Andre Borrmann[1]
[1] Chair of Computational Modeling and Simulation, Technical University of Munich, Germany
m.saeed.mafipour@tum.de

Abstract. Technical drawings are a resource to create the geometric digital twin (DT) of existing bridges. A bridge DT demonstrates the current geometric-semantic information of the structure and supports the operation and maintenance process of bridges. Despite the significant advantages of a bridge DT, creating its 3D model from drawings is costly and labor-intensive. This paper presents a method to digitalize the technical drawing of bridges by deep learning models such that the required data for geometric modeling can be extracted more straightforwardly. The parametric model of bridge elements is created and used to generate a synthetic dataset. This dataset is combined with the actual drawings and a deep learning model is trained to detect bridge elements. Dimensions are also extracted using a pre-trained model and digitalized through optical character recognition (OCR). The results of the paper show that the model can detect different elements in drawings with a mean average precision (mAP) of 89.15%.

1. Introduction

The recent ASCE report card (ASCE, 2021) shows that the deterioration rate of bridges is exceeding the rate of rehabilitation, repair, and replacement. Following the national bridge standards, the structural conditions of bridges need to be evaluated at regular intervals during the service life of the structure. In current practice, most of the processes involved with inspection, condition assessment, and quality assurance of bridges is only partially supported by digital methods that in turn increase the required time and cost for the evaluation of bridges.

Building Information Modeling (BIM) is playing a prominent role in the architecture, engineering, and construction (AEC) industry by providing the geometric-semantic representation of assets (Chipman, 2016). In the infrastructure domain, Bridge Information Modeling (BrIM) has been proposed in various phases of the structure (Marzouk and Hisham, 2012). In comparison with traditional 2D drawings, BrIM models are capable of capturing the design complexities and assembly. A detailed comparison by (Kumar et al., 2017) showed the significant advantage of using BrIM over the conventional 2D drawings in the implementation time of three different bridge projects by spending five times lower time. BrIM models can be enriched with detailed information regarding the current status of the bridge and used as support for condition assessment, risk identification, and health monitoring of the structure. Most recently, the concept of as-is BrIM has been extended to digital twin (DT) (Pan et al., 2019, Lu et al., 2020). A DT can inherit all the features of a BrIM model and is linked with the existing asset in reality.

Despite the considerable advantages of BrIM and DT models throughout the entire life cycle of bridges, traditional 2D drawings are still the most commonly used standard. This is mainly because most of the process involved with the design, construction, and maintenance of bridges is loosely and only partially supported by digital methods. Also, most of the existing bridges have been constructed decades ago only using 2D drawings. To benefit from the advantages of BrIM models, the traditional 2D drawings need to be transformed into a 3D model representing detailed information on bridges. In the current practice, the geometric 3D model of bridges is created manually from 2D drawings which in turn, increases the required time and cost

(Girardet and Boton, 2021). Hence, the corresponding transportation authorities are currently faced with heightened pressure to speed up the modeling process of bridges from 2D drawings.

This paper presents a method to interpret the 2D drawings of single-span reinforced concrete (RC) bridges to facilitate the required steps to generate a BrIM model. The 2D PDF drawings of bridges are analyzed and based on the element type and its 2D views, the existing classes are derived. Next, the PDF files are preprocessed and converted to a raster graphics format. An object detection task is defined and a deep learning model is trained for extracting the elements and their corresponding views. To augment the dataset, the parametric model of each class is created in a CAD authoring system, and by changing the value of parameters, new synthetic drawings are generated and added to the actual dataset. The trained model is tested on the unseen scans of the actual bridges. To extract the value of parameters, a scene text detection task is defined and used to detect the bounding boxes around dimensions. These numbers are finally converted to a digital format using optical character recognition (OCR). Finally, the related views are represented by a graph and the parametric model of elements is generated semi-automatically.

2. Background

Various methods have been proposed for the digitalization of technical drawings. The proposed methods can be categorized into image processing techniques and deep learning models.

Image processing techniques start with cleaning drawings and continue with transformation as well as the extraction of semantic information through primitive/symbol detection methods. These algorithms employ graphic recognition and document analysis to recognize characters in a document. A text extraction algorithm based on image processing can be found in (Fletcher and Kasturi, 1988) which employs a connected component generation algorithm and Hough transformer to generate text strings and characters. Image processing also contains practical techniques such as binarization through which the number of channels can be reduced and existing noises are partially removed (Ghorbel, 2012). Transformation of the raster images to vector images is another technique that is capable of describing sets of pixels by primitives such as lines, polylines, or curves (Lewis and Séquin, 1998). The existing vectorization methods can be categorized into Hough transform-based, thinning-based, contour-based, run-graph-based, mesh-pattern-based, and sparse-pixel-based (Wenyin and Dori, 1999). Symbol recognition has been also another task addressed by image processing. The existing symbol recognition methods are mostly based on rule databases (Or et al., 2005) that can recognize new symbols adaptively. These techniques have been principally or partially used in processing traditional 2D drawings. Lewis and Séquin (1998) generated the 3D model of the building from drawings through a line detection algorithm and connectivity rule between elements. Akanbi and Zhang (2022) proposed a framework to generate the IFC model of bridges from technical drawings by converting the tagged data from different views of bridges to a 3D shell model. Lu et al. (2020) recognized special symbols in CAD drawings of buildings using a clustering algorithm to generate the BIM model of buildings.

Contrary to the image processing techniques that use hand-crafted features to process inputs, deep learning models benefit from a mechanism to automatically generate the required features for processing images. These models are generally less dependent on pre-defined thresholds and show higher flexibility in prediction. Semantic segmentation, object detection, scene text detection, and optical character recognition are covered tasks by deep learning models to enrich technical drawings. Vilgertshofer et al. (2020) employed a convolutional neural network (CNN) to detect symbols and infrastructure elements in railway technical drawings. Elyan et al. (2018)

also classified symbols in technical drawings using a random forest (RF), support vector machine (SVM), and CNN model. Nguyen et al. (2021) detected objects and text in large-scale technical drawings through a fast R-CNN model and optical character recognition (OCR). Van Daele et al. (2021) used DBSCAN, as a clustering algorithm to segment elements and classified them through a CNN.

3. Bridge element detection

Contrary to building floor plans in which a similar pattern with unique symbols can be mostly observed, technical drawings of bridges contain various elements with different shapes. These shapes generally show different views of cross-sections or faces of the elements that need to be constructed in practice. Processing the entire technical sheet of a bridge can be cumbersome due to the existence of various elements and lack of similarity. Hence, this research focuses on single-span reinforced concrete (RC) bridges and proposes a pipeline to digitalize the drawings of this specific type of bridges that are almost similar in shape.

3.1 Preprocessing

2D bridge drawings are generally in PDF following a vector file format. To enable the processing of the input drawings, they need to be converted to a raster graphic file format. Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF), and Portable Network Graphics (PNG) are the most well-known raster formats. In this research, the input PDF files are converted to JPEG format. This step can be skipped if the images are already in raster-based format. To reduce the processing load and save memory during the training process, all the images are resized to the width of 720px while preserving the same aspect ratio in the calculation of height. Also, the original dimensions are saved for any further inverse transformation. The images are also binarized/normalized by setting a threshold to zero and one.

3.2 Identification of relevant classes

Empirical analysis of 15 single-span bridges showed that similar patterns can be found between the same elements of bridges. Depending on the existing visual similarity among the faces/cross-sections, 14 classes have been recognized in bridges whose class name and number of instances have been shown in Figure 1(a). As can be seen, the collected dataset is imbalanced as it contains a different number of instances in each class. Also, the number of instances in some of the classes might not be adequate for a deep learning model to learn the existing patterns correctly. To address these issues, synthetic instances are created through parametric modeling and added to the current dataset. This process which is covered in the next section leads to 1213 instances as shown in Figure 1(b).

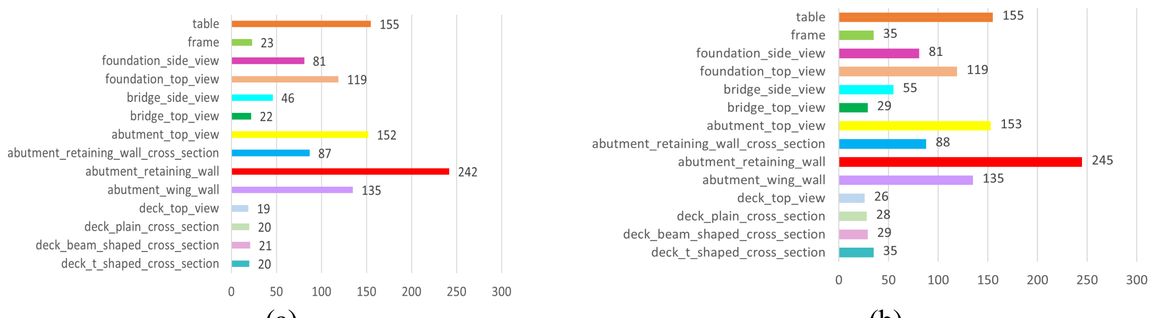


Figure 1: Number of instances per class: (a) before data augmentation; (b) after data augmentation.

3.3 Data augmentation through parametric modeling

Parametric modeling is a solid modeling approach that eliminates the need for the creation of the model from scratch every time dimensions change (Shah and Mäntylä, 1995). Parametric models save all the dependencies and relations between components and provide an access point through which the model can be steered (Ji et al., 2013). To increase the number of instances per class and improve the stability of the model in prediction, the parametric model of all the classes is created as shown in Figure 2. Due to the parametric design of elements, the parameter values (dimensions) can be simply changed and various instances from the classes are generated. These instances are placed in technical sheets and PDF files of the drawings are generated. This process which is very close to the actual process of generating drawings in practice can provide more samples for training the deep learning model.

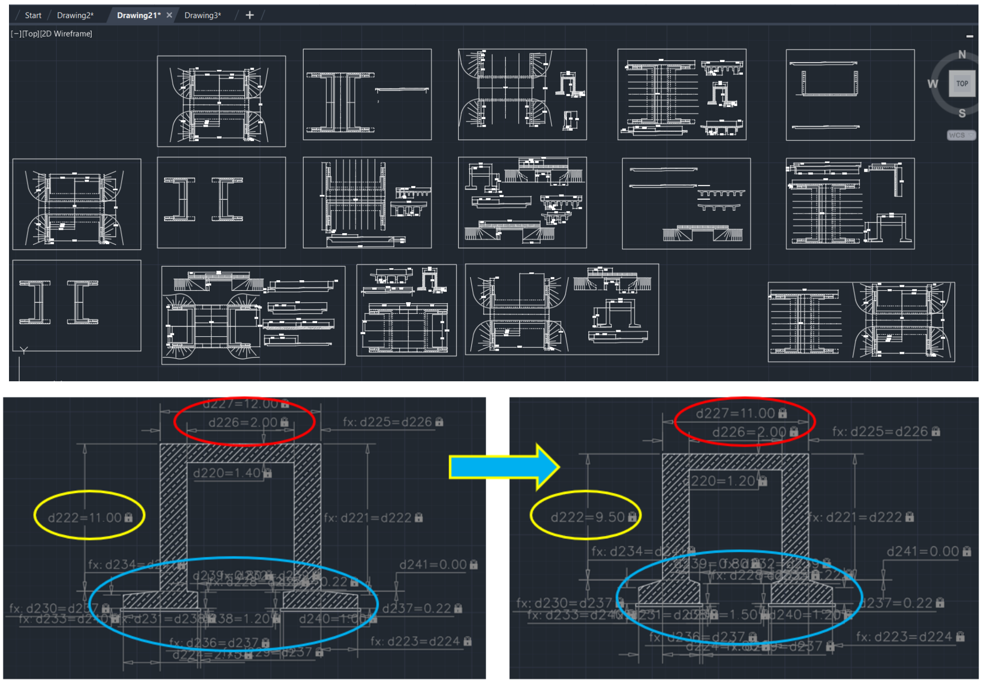


Figure 2: Parametric design of element to generate synthetic technical drawings.

3.4 Deep learning model for object detection

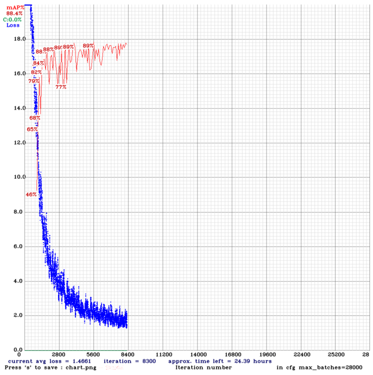
YOLO (Bochkovskiy et al., 2020) is utilized to detect elements in the technical drawing of bridges. YOLO is a fast and efficient deep learning model which is capable of automating the object detection task. This model divides the input images into an $S \times S$ grid and predicts the number of B bounding boxes inside each cell. The bounding boxes are predicted based on the extracted features from the entire image. Each bounding box takes a confidence score showing how confident the model is about a bounding box containing an object. The confidence score is equal to the intersection over union (IoU) of prediction. In addition to the bounding boxes that are predicted inside each cell, the conditional class probabilities C are predicted by the network. These probability values show how likely the cell contains the object of interest. The model is finally trained such that the more confident bounding boxes inside each cell are predicted.

3.5 Hyperparameters

YOLO can be configured through a set of hyperparameters. The hyperparameters of the model have been obtained through a trial and error process and the values leading to the best results are reported. A batch size of 32 with subdivision 16 has been considered for training. 80 percent of images have been used for training the model and 20 percent for testing. The input size of images is fixed to 608 × 608 and the model is trained with a learning rate of 0.001 for 8300 iterations. As recommended in the documentation of the model, the maximum number of batches is set to the number of classes times 2000, i.e. 14 × 2000 = 28000. Two steps are also considered for the model at which scales are applied. These values are 80% and 90% of the maximum number of batches (steps = 22400, 25200). The number of 57 filters is also selected before each of the three layers of the YOLO architecture.

3.6 Results of element detection

The model is trained for more than 8000 iterations and its performance is tested on the actual technical drawings of bridges. To evaluate the performance of the model, four evaluation metrics of precision, recall, f1 score, mean intersection over union (mIoU) and mean average precision (mAP) is calculated. Figure 3 shows the validation/testing results of the model after 8300 iterations. As can be seen, the model has been capable of achieving an mAP of 89.15%. This value in the prediction of 14 different classes shows that the model is reliable in the detection of bridge elements.



Precision	0.95
Recall	0.91
F1 score	0.92
mIoU	79.13%
mAP	89.15%

Figure 3: The validation/testing results of the model on unseen drawings.

To calculate the confidence level of the detected bounding boxes, a threshold value equal to 0.4 is set for IoU such that the greater values are considered true positive predictions. Also, a Non-Maximal Suppression (NMS) threshold is set (NMS = 0.6) to only retain the bounding boxes with the highest probability of object detection. Figure 4 shows the predicted bounding boxes by the models with the confidence level in the prediction of each bounding box. As can be seen, the trained model has shown a successful performance in the prediction as the confidence level of most of the bounding boxes is equal to 1. Note that many instances might exist in the technical drawing of bridges while most of them might be related to semantics such as the reinforcement details of elements. Although these instances can also provide useful information about elements, they cannot provide the required geometric information for modeling. As the geometric information of elements is of interest, no class has been assigned to these instances and the model has also ignored them in prediction.

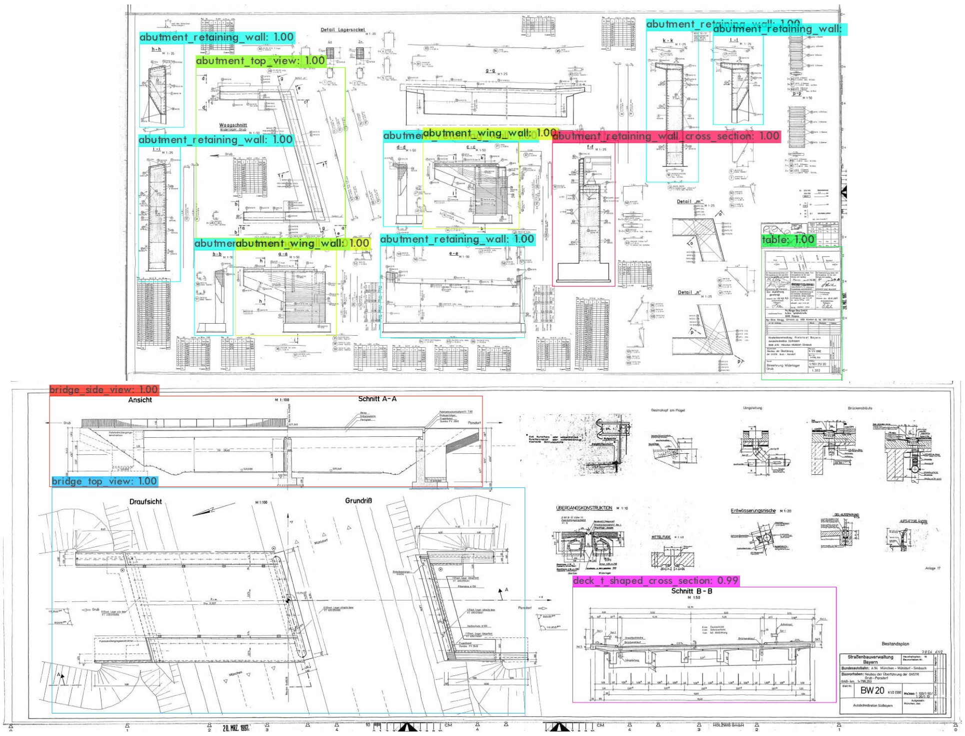


Figure 4: Predicted bounding boxes by the trained model with the confidence level.

4. Text detection

Element detection by a deep learning model can simplify the digitalization problem to a large extent as it narrows down the problem from the entire image to the image of each element. Also, a label is assigned to each element representing the type of the element. To further digitalize the extracted instances from the previous step, the dimensions of the elements as well as the existing text can be detected. To do so, Craft (Baek et al., 2019), as a deep learning model is employed. This model benefits from a VGG 16 (Simonyan and Zisserman, 2014) architecture in the encoding process of images and decodes them based on a U-net architecture (Ronneberger et al., 2015) to combine high-level features with the low-level features generated in the initial layers. In technical drawings, dimensions/text are written either horizontally or vertically. As the convolution layers are not invariant to rotation, most of the vertical dimensions might not be detected by the model. To address this problem, in addition to the original image, its rotated version (90 degrees) is also processed by the model. The detected bounding boxes in the second version are transformed back and merged with the original version. This process results in bounding boxes showing the existing text/dimensions in each instance (Figure 5). To digitalize the detected numbers, they can be read through optical character recognition (OCR), and the machine-readable text format of the numbers is generated.

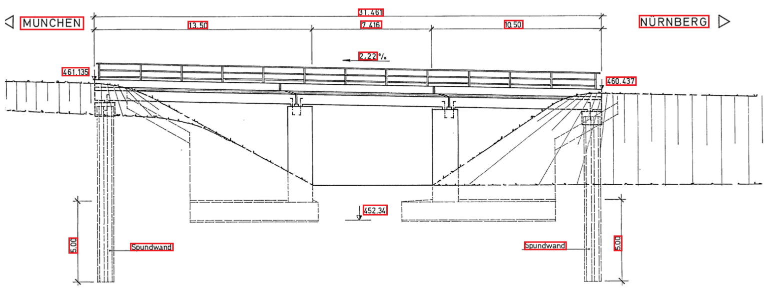


Figure 5: Detected bounding boxes by the pre-trained model after postprocessing.

5. Geometric model generation

Element detection results in bounding boxes showing the instances required for the geometric modeling of bridges. In general, there are certain types of dependencies between the instances. For example, abutments consist of a retaining wall and two wing walls, each of which is drawn from different views. To acquire the required information for 3D modeling, these views need to be connected. To this end, a graph data model is created as shown in Figure 6 that receives the detected instances by the deep learning model and categorizes them based on their labels. Each node of this graph contains various views that can result in the 3D model of the element. To facilitate the modeling process, a library containing families representing the parametric model of each element is created. This reverse engineering approach results in a parametric prototype model that only requires the value of parameters to generate the model from its various views. Considering the parametric model of each bridge element and the attached reports to each node, the required views can be shown to users, and the proposed values are manually imported into the model. As a result, a parametric model is obtained that represents the element shown in the technical drawings.

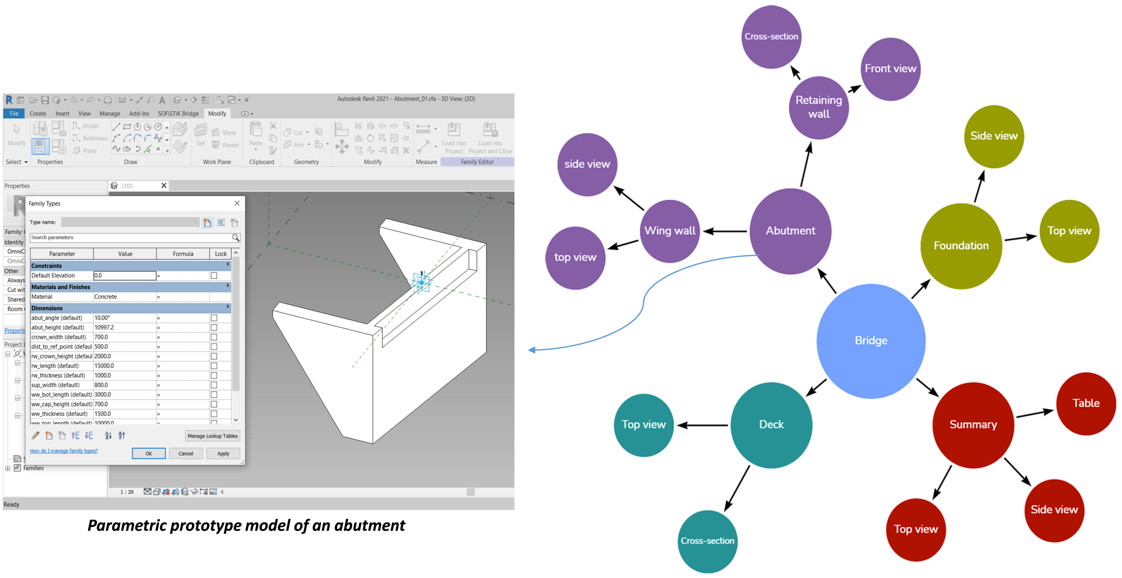


Figure 6: Geometric Modelling through reverse engineering and a graph data model.

6. Conclusion

This paper has presented a method to facilitate the parametric modeling process of bridges from their technical drawings. 14 different classes that might be required for the geometric modeling of single-span bridges are derived. Depending on the existing classes, the parametric model of each class is created and synthetic drawings are generated. A deep learning model is trained on the actual and synthetic drawings to detect the existing elements in the drawings. The results of testing the model on unseen actual drawings show that the model is capable of recognizing instances with a mean average precision (mAP) of 89.15%. The existing text/dimensions in the bridge instances have been also detected by a pre-trained deep learning model. To facilitate the 3D model generation, a graph model has been proposed to represent the dependencies between different views. Following reverse engineering, the parametric prototype model of the bridge elements is created. This parametric model can be connected to the various views belonging to the objects/nodes and the machine-readable values are imported into the model through a semi-automatic process. In future works, the proposed approach is used for creating the entire model of bridges and it is further automated by tagging dimensions to the edges of elements.

Acknowledgment

We thank the German Federal Ministry for Digital and Transport (BMDV) for funding this research in the scope of the TwinGen project.

References

ASCE (2021). ASCE's 2021 Infrastructure Report Card.

Baek, Y., B. Lee, et al. (2019). Character region awareness for text detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Bochkovskiy, A., C.-Y. Wang, et al. (2020). "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934*.

Chipman, T. (2016). Bridge information modeling standardization, US Department of Transportation, Federal Highway Administration.

Elyan, E., C. M. Garcia, et al. (2018). Symbols classification in engineering drawings. *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE.

Fletcher, L. A. and R. Kasturi (1988). "A robust algorithm for text string separation from mixed text/graphics images." *IEEE transactions on pattern analysis and machine intelligence* **10**(6): 910-918.

Ghorbel, A. (2012). Interprétation interactive de documents structurés: application à la rétroconversion de plans d'architecture manuscrits, INSA de Rennes.

Girardet, A. and C. Boton (2021). "A parametric BIM approach to foster bridge project design and analysis." *Automation in Construction* **126**: 103679.

Ji, Y., A. Borrmann, et al. (2013). "Exchange of parametric bridge models using a neutral data format." *Journal of Computing in Civil Engineering* **27**(6): 593-606.

Kumar, B., H. Cai, et al. (2017). An assessment of benefits of using BIM on an infrastructure project. *International Conference on Sustainable Infrastructure 2017*.

Lewis, R. and C. Séquin (1998). "Generation of 3D building models from 2D architectural plans." *Computer-Aided Design* **30**(10): 765-779.

Lu, Q., L. Chen, et al. (2020). "Semi-automatic geometric digital twinning for existing buildings based on images and CAD drawings." *Automation in Construction* **115**: 103183.

Marzouk, M. and M. Hisham (2012). Bridge information modeling in sustainable bridge management. ICSDC 2011: Integrating Sustainability Practices in the Construction Industry**:** 457-466.

Nguyen, M. T., V. L. Pham, et al. (2021). Object detection and text recognition in large-scale technical drawings. ICPRAM 2021.

Or, S.-h., K.-H. Wong, et al. (2005). "Highly automatic approach to architectural floorplan image understanding & model generation." *Pattern Recognition*: 25-32.

Pan, Y., A. Borrmann, et al. (2019). Built Environment Digital Twinning. Report of the International Workshop on Built Environment Digital Twinning presented by TUM Institute for Advanced Study and Siemens AG. Technical University of Munich, Germany.

Ronneberger, O., P. Fischer, et al. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer.

Shah, J. J. and M. Mäntylä (1995). Parametric and feature-based CAD/CAM: concepts, techniques, and applications, John Wiley & Sons.

Simonyan, K. and A. Zisserman (2014). "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*.

Van Daele, D., N. Decleyre, et al. (2021). An automated engineering assistant: Learning parsers for technical drawings. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Vilgertshofer, S., D. Stoitchkov, et al. (2020). "Recognising railway infrastructure elements in videos and drawings using neural networks." *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction* **172**(1): 19-33.

Wenyin, L. and D. Dori (1999). "From raster to vectors: extracting visual information from line drawings." *Pattern Analysis & Applications* **2**: 10-21.