The RIMcomb Research Project: Towards the Application of Building Information Modeling in Railway Equipment Engineering

S. Vilgertshofer, D. Stoitchkov, S. Esser & A. Borrmann
Chair of Computational Modeling and Simulation, Leonhard Obermeyer Center
Technical University of Munich, Germany

S. Muhič
AEC3 Deutschland GmbH, Germany

T. Winkelbauer
SIGNON Deutschland GmbH, Germany

ABSTRACT: This paper presents our research towards the utilization of Building Information Modeling (BIM) methods in Railway Equipment Engineering. While BIM is already applied in several railway infrastructure construction projects, so far, it is mainly used for construction engineering tasks, i.e. for modeling the alignment, bridges or tunnels. The RIMcomb research project targets the further application in scope of rail engineering subsections such as control and safety systems, train control systems, telecommunication systems, electric power systems, rail power supply or cable management. In this paper, we are going to give a general overview of the project and will discuss first findings of our approach to digitizing conventional technical drawings in order to translate their content into a machine-readable form.

1 INTRODUCTION

The planning and restoration of railway tracks including technical equipment is of high importance for maintaining this important backbone of the Europe's infrastructure. While Building Information Modeling (BIM) is widely established in the building domain, it is less often used for the planning and construction of infrastructure facilities. So far, it has not been applied as a general concept for railway infrastructure equipment engineering yet.

At the moment, the development of various proposals for extending BIM concepts and standards for the infrastructure domain is underway. buildingSMART International (bSI) is working on extending the Industry Foundation Classes (IFC) for representing infrastructure facilities. Currently, IFC is primarily focused on the building domain and comprises a wide range of classes, which enable the exchange of detailed BIM models. Although first infrastructure concepts (such as alignment and linear referencing), have been published as part of the latest version IFC4.1, more specific elements are still missing. To fill this gap, the Rail Room was established in 2017 as a subsection of bSI dedicated to the development of rail-specific extensions. In the past, several proposals had been published that provided the foundation for the respective standardization effort, Examples include IFC Tunnel (Vilgertshofer et al., 2016), IFC Rail (bSI SPEC) and IFC Road (bSI SPEC) (buildingSMART, 2016). Contributing to these efforts, is one of the main objectives of the RIMcomb project.

On a general level, the RIMcomb research project aims at introducing BIM methods into the sector of railway equipment engineering by analyzing current conventional workflows in order to identify beneficial BIM use cases. The main goal is the avoidance of data loss or inconsistent data during different planning stages as well as to identify labor-intensive tasks, which could benefit from automation. Especially, inconsistencies across the different disciplines involved in railway design is a severe problem that can be overcome by using BIM methods instead of conventional 2D drawings.

In this regard, we are looking into possibilities of using the existing version of the IFC data format to represent railway equipment.

Additionally, we analyzed preexisting data schemas that are currently used in this domain. As these are not necessarily compatible with one another we are developing a tool that can import various data formats and uses this information for creating an integrated BIM model, preferably in the IFC format. Furthermore, we use conventional 2D drawings as an additional data source for BIM model generation.

This paper is structured as follows: In Section 2 we give a general overview of the RIMcomb research project's scope in order to put the subsequently described approaches into context. Section 3 describes our approach of digitizing technical drawings. We give an overview of the methods applied and the conclusions of our testing process. Section 4 describes how we intend to further use the digitized plan data and other data sources for the creation of BIM models

representing railway equipment. The paper ends with a summary and an outlook.

2 THE RIMCOMB PROJECT

The research project "RIMcomb: Railway Information Modeling for the Equipment of Railway Infrastructure" was initiated by SIGNON Deutschland GmbH in 2016 in cooperation with Technical University of Munich and AEC3 Deutschland GmbH. The project is funded by the Bavarian Research Foundation and started in early 2017.

The main focus of the research project is to develop and adapt new computer-supported methods for model-based collaboration between the different subsections of technical railway equipment in order to increase the efficiency of the planning process and the quality of the outcome.

During the design, planning and construction of railway infrastructure and technical equipment, a multitude of domains experts are involved. Therefore, data exchange between these participants is an issue that needs to be addressed, as there a various specialized software tools available for different tasks that do not implement any common data standard.

Also, most of railway construction projects involve the modernization or alteration of existing infrastructure and thus the industry has to rely on technical drawings from past decades that are not necessarily consistent with the real-world circumstances.

As described in Section 2 we developed a method that allows the automatic recognition of plan symbols in technical drawings of railway infrastructure. Besides the use of this data described in Section 3, we also aim at comparing the generated data with real-world data in order to identify discrepancies. This use case may create a significant benefit for the railway companies as the manual comparison of the as-built drawings with real-world stock data requires a considerable amount of effort but is nonetheless necessary.

Here, machine-learning and convolutional neural networks (as outlined in Section 2) will also be employed to process video files of railway tracks in order to identify objects in single frames for the mapping of objects such as signals, balises, switches or poles of overhead lines.

Another aspect in the scope of the research project is the development of a method that allows the automated checking of technical rules and regulations. As such approaches have already been applied outside of the infrastructure domain (Preidel and Borrmann, 2016), we see a huge benefit in introducing them into the domain of railway equipment, as the amount of rules and regulations in this domain requires a large amount of manual work.

3 DITIALIZATION OF PLAN DATA

One major topic in the research project is the digitalization of conventional drawings depicting railway equipment infrastructure. While most drawings are available digitally, the interpretation of these plans has to be undertaken manually. This is necessary when the accuracy of plans has to be compared to real-world circumstances or in case of stocktaking.

Our approach aims at supporting this process in order to reduce the manual effort by automating at least parts of this image interpretation process. The first step towards this goal is the automatic recognition and highlighting of plan symbols on a given drawing and the subsequent storing of their count and location.

3.1 *Theoretical background*

In a first step, three preexisting methods of image recognition are described. We evaluated those techniques in respect to the given problem. As none of those methods matched our requirements completely we also tested Convolutional Neural Networks, which are already widely used for image recognition, in respect of their ability to detect plan symbols.

3.1.1 *Template Matching*

Template Matching is a well-known method for the searching of a template image in a larger image. This is made by sliding the template image over the input (larger) image and comparing them at every position. The result of this method is a grayscale image with a size of *(W-w+1, H-h+1)*, where *W* and *H* are the width and the height of the input image, *w* and *h* are the width and the height of the template image. We investigated different comparison methods for each one of which there is a normalized version (Kaehler and Bradski, 2016).

In this work two of these methods are used:

- Normalized Square Difference Matching Method:

$$R(x,y) = \frac{\sum_{x',y'}\left(T(x',y') - I(x+x',y+x')\right)^2}{\sqrt{\sum_{x',y'}T(x',y')^2 \cdot \sum_{x',y'}I(x+x',y+y')^2}}$$

- Normalized Correlation Coefficient Matching Method:

$$R(x,y) = \frac{\sum_{x',y'}\left(T'(x',y') \cdot I'(x+x',y+x')\right)}{\sqrt{\sum_{x',y'}T'(x',y')^2 \cdot \sum_{x',y'}I'(x+x',y+y')^2}}$$

Here *T* is the template image, *I* the input image, *R* the result image and

$$T'(x',y') = T(x',y') - 1/(w \cdot h) \cdot \sum_{x'',y''} T(x'',y'')$$

$$I'(x+x',y+y') =$$

$$I(x + x', y + y') - 1/(w \cdot h) \cdot \sum_{x'',y''} T(x + x'', y + y'')$$

With the first comparison method a perfect match is 0 and a perfect mismatch is 1. For the second method 1 stands for a perfect match and -1 for a complete mismatch.

3.1.2 Contours search

This technique compares objects with their contours. A contour lies on the border between the black and white spaces. A contour tree contains the hierarchy between the contours or how they relate to one another. Contours can be compared with the help of image moments. An image moment is a characteristic of a given contour calculated by summing over the pixels of that contour. The Hu invariant moments (Hu, 1962) were used in our work to compare contours. The Hu moments are combinations of different normalized central moments and are scale, rotation and translation invariant. However, this method works only if the symbol is not connected to other lines or objects in the image, because then the contour around the symbol can't be defined properly.

3.1.3 Cascade Classifiers

First proposed by Viola & Jones, 2001, it was originally used for face detection, but can be used for many types of objects. This method learns by searching for Haar-like features in an image. These features differentiate between dark and bright parts of an image by subtracting the sum of the pixels in the white parts from the sum of the pixels in the dark parts. The features are placed over the image in different locations and sizes to extract certain patterns.
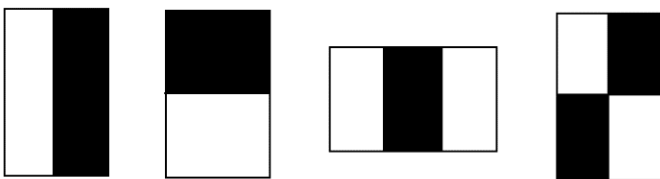
Figure 1. The Haar-like features

A cascade classifier is a machine learning method in which the information computed in a given classifier is used for the next classifier and becomes more complex at each stage. "The Viola-Jones detector uses AdaBoost, but inside of a larger context called a "rejection cascade". This "cascade" is a series of nodes, where each node is itself a distinct multi-tree Ada-Boosted classifier. The basic operation of the cascade is that sub-windows from an image are sequentially tested against all of the nodes, in a particular order, and those windows that "pass" every classifier are deemed to be members of the class being sought."(Kaehler and Bradski, 2016).

3.1.4 Convolutional Neural Networks

A neural network is a form of machine learning, in which the computer "learns" from given data. For the sake of this work, a convolutional neural network (CNN or ConvNets) is used, which speeds up the training process especially with images. One of the first CNN is called LeNet5 (LeCun et al., 1998), which has started an new era of state-of-the-art artificial intelligence. CNNs are so effective for image recognition because they use filters to detect patterns (or features) in an image. Different locations of the image are searched for these features and a value is saved, representing how well every pattern matches the image in a given location (Rohrer, 2016). This results in a map which represents where each feature occurs in the image. By matching the feature for every possible location in the image, a convolution is made. The features are then passed to the actual neural network (In Figure 2 from the left):
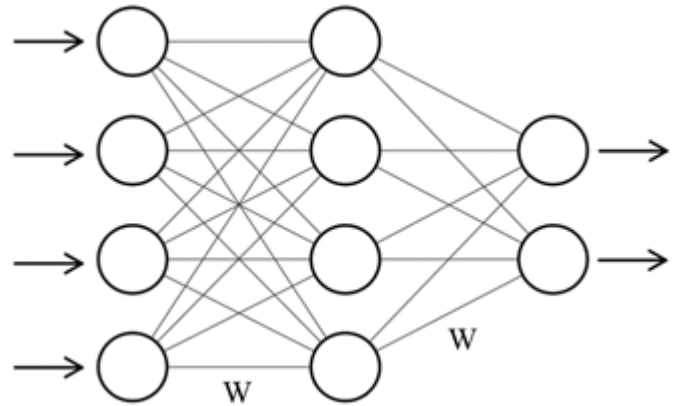
Figure 2. Simplified neural network with an input layer on the left, an output layer on the right and one hidden layer in the middle.

Every layer has multiple neurons which are connected to one another between the layers (Figure 2). Each of the connections between the nodes is weighted (marked with *W*), which means that they are multiplied by a number, that is between -1 and 1. The neural network "learns" by adjusting the weight until the required output is obtained. An activation function is required to generate an output from given input in a processing unit (or neuron). The ReLU (Rectified Linear Unit) function was used in this work to normalize the values in the feature map that was calculated. This simplifies the calculation by setting all negative values to zero. The ReLU has the mathematical form: *f(x) = max(0, x)*. To make the calculations less complex the size of the feature map can be reduced with a max pooling layer. This is done by taking only the maximum value of a given window and saving it at the correct location in a new smaller feature map.

3.2 Concept

Even in today's modern world, in which almost everything is digitized, many technical drawings are still
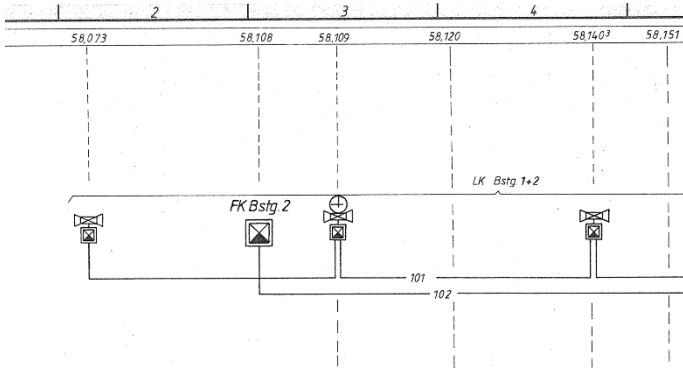
Figure 4. Test image.

Figure 3. Test symbol.

only available in paper form. The main reason for this is that most of the infrastructure that exists today was built before the widespread availability of computers. Many of these drawings are now being digitized to be used by modern computer programs. Much of the digitization process of a technical drawing consists of recognizing and locating different symbols which is often a difficult and time-consuming task. Therefore, this work aims to find a method of automating this process at least in parts and drastically reducing the processing time.

The different methods for symbol recognition were tested in different ways. The first two methods do not require training and can be implemented directly to find symbols. The Cascade Classifiers and the Convolutional Neural Networks require many images for training. These images are generated by cutting technical drawings into thousands of smaller images and then placing the searched symbol over half of the images in different sizes and locations. This generates two kinds of images – positive images with the symbol and negative images without it. After these methods have been trained, they must be tested to measure

the accuracy of finding the symbol. This course of action differs from other approaches that use machine learning techniques for image detection. Normally, our way of generating the training data would result in a machine learning algorithm, that would only detect the exact image that is was trained with – which is not useful for e.g. detecting faces in a picture. However, since plan symbols are always of a similar shape or match one another exactly, the chosen training data work fine in our scope.

3.3 *Prototypical implementation*

The methods were tested on many different technical drawings to measure the accuracy. An example can be seen in Figure 4, which is a part of a telecommunication technical drawing.

Figure 3 shows the symbol for which the image was searched.

As a result, the corresponding symbols in the image are marked by red rectangles (Figure 5). The different methods show similarly results with this image, but the template matching method is not scale and rotation invariant and works only if the symbol always has the same size and rotation. The cascade classifiers method is scale and rotation invariant, but still fails if the shape of the symbol is simple and therefore does not have many distinct features. However, the convolutional neural networks were found to be very accurate, even when tested with many different types of symbols.
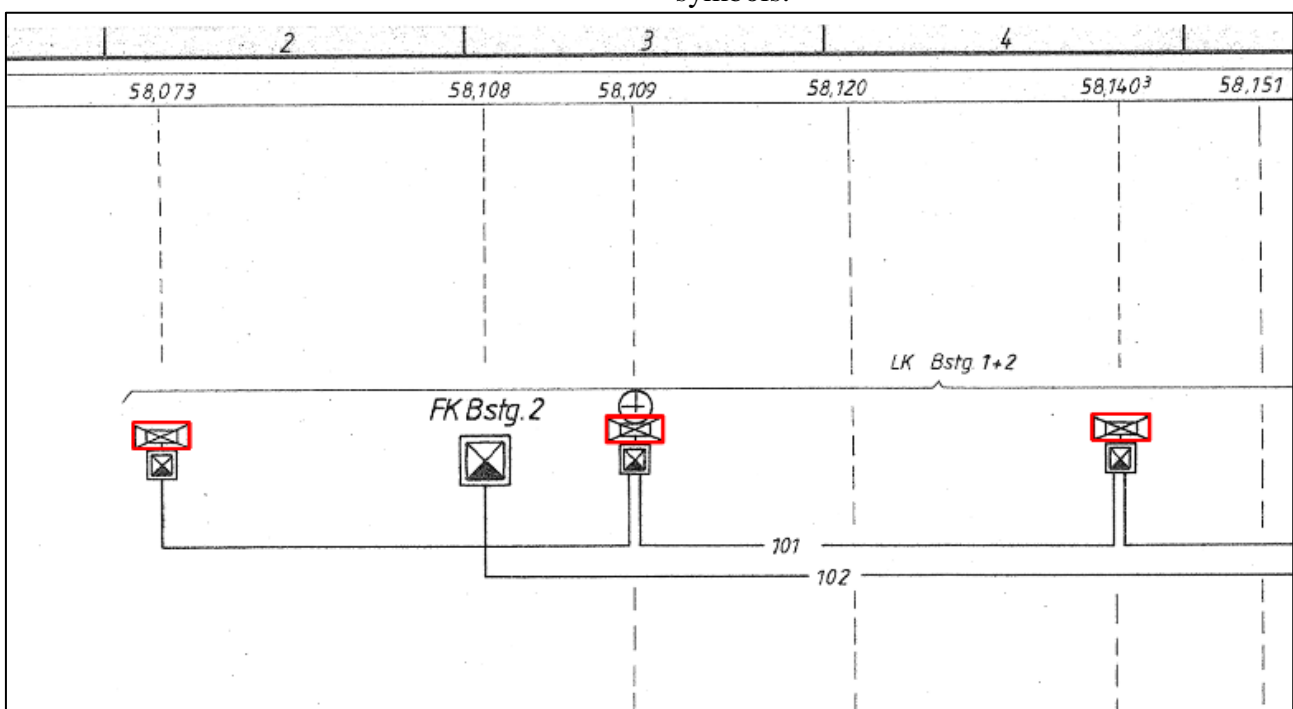

Figure 5: Exemplary results.

	Advantage	Disadvantage	Limitation
Template matching	Very easy and fast to implement	Not rotation and scale invariant, thereshold value	Works only if the symbol always occur with the same size and rotation
Contours	Easy and fast to implement, rotation and scale invariant	Connected or overlapped symbols, open shapes, threshold value	Not practical for big technical drawings in which the symbols are connected or crossed by lines
Cascade classifiers	Versatile, works under many conditions	Requires training, threshold values	Works only if the symbol has many features to differentiate it, requires big image data
Convolutional neural networks	Very versatile, works under almost any conditions	Requires training	Powerful GPU, requires big image data

Figure 6: Overview of advantages and disadvantages of the different image recognition methods.

The CNN was not only tested with technical drawings, but also with artificially generated images to better test the accuracy. It showed above 95% of accuracy for detecting if a symbol is on the image or not and 80-85% of accuracy for finding the exact location of the symbol.

3.4 *Conclusion*

Both the template matching and the contour methods are very easy and fast to implement but work only under certain conditions. The cascade classifier is a more complex method, but still fails if the symbol has too few features. The last technique, the artificial neural networks, is the most "sophisticated" method and can be used in many situations with very few disadvantages and almost no limitations with today's powerful computers. The neural network, which was designed for this work, can be further modified for even better results. An overview of our findings is given in Figure 6 (Stoitchkov, 2018).

4 BIM MODELS FOR RAILWAY INFRASTRUCTURE

The previously described method of automatically detecting plan symbols is only one aspect of the RIMcomb research project's goals. To use the collected data for creating BIM models we need to investigate how such models can be created. Furthermore, the image recognition approach is only one of various sources of preexisting data that can be used for model creation. In the following section, an overview of data formats is given as well as an approach of how a model can be created from different data sources.

4.1 *Data Formats*

One of the main aspects of Building Information Modeling is data consistency and the collaboration of different project partners using the same set of data. These two aspects require data formats, which can be imported and exported by different software applications.

An analysis of the software market offers many different software tools, which meet these requirements and can be used for creating building models or performing simulations on them. The vendor-neutral format "Industry Foundation Classes" (IFC) provides the possibility to exchange model data amongst an enormous range of different applications. Besides open formats, many software providers implement proprietary interfaces between their tools, which often lead to a higher data quality in the receiving application but is also limited to the use of a few tools.

In contrast to these developments for building constructions, no state-of-the-art application or exchange format does exist for modelling infrastructure projects, although projects such as IFC-Rail or the proposal of Allah Bukhsh et al. (2016) are under development. One reason is based in the geometrical project dimensions: Buildings normally have a ground area less than 100 meters per site and an elevation of a couple of meters. Therefore, a high information density occurs on a comparatively small volume (different layers of a wall, structural analysis, architectural properties, etc.). In contrast to this, infrastructural projects usually reach over many kilometers and therefore the data density can extremely vary within the project's alignment. Thinking of modelling a railway path between two stations makes this aspect a bit clearer: The nearer the station is, the more signals, switches or security systems are needed whereas the lines in the outer field only need rails, swells, railway signals and electric components.

This leads to the need of new storing approaches in (existing) data formats. There are already some data formats in use today, however, they are mostly limited to one discipline of railway engineering or can only be used by one specific software tool.

Modelling of infrastructural components is quite a challenge today, as most of the software products for infrastructure planning are based on the paradigm of drawing generation and thus are not capable to represent semantically rich 3D models.

At the same time, these tools are tailored and well-suited for the respective engineering task. On the other hand, the creation of BIM models for railway engineering is only partially supported. Creating model components in well-established BIM authoring tools, which are known from building modeling, can only be realized by either using building elements and append additional properties or creating new components based on generic templates. Both approaches can only offer the high data quality that is known from today's building models, when a lot of effort is put into the model creation.

Also, the data exchange of such models is a big challenge. Once they have been created in a BIM authoring tool, the export into a vendor-neutral format leads to another significant issue. Up to now, IFC has no preexisting classes for storing infrastructural components (they will be introduced in IFC 5). Besides IFC, there are some data formats such as *PlanPro* or *railML* providing schemata for storing and transferring information about infrastructural components (especially for railway infrastructure) but cannot represent the elements' geometry in a high quality.

4.2 *Model creation*

Due to the lack of a standardized data format and the need of the digitalization of existing drawings, a software-application is being developed in scope of the RIMcomb research project that combines data from different sources and automatically creates an integrated BIM model from this diverse information (Figure 6). To integrate as much import data formats as possible, a basic task is to define a set of the minimal necessary information to create a model, which is suitable for advanced modelling and simulation tasks (e.g. supplying the existing state of a railway path for redevelopment planning).

To start with a manageable set of data, the data schema *PlanPro* was used to set up the preprocessing tool. *PlanPro* is developed by DB Netz AG and it is used for gathering information needed in an interlocking system. Therefore, this format is powerful for storing information about different components that are necessary for security aspects of railway traffic. However, this schema holds only coarse information about the railway alignment, which causes to integrate additional GIS data. We plan to integrate this information in a later implementation step.

The link between the preprocessing tool and the BIM Editor is realized by an SQL database, which is controlled via the Entity Framework. This offers a lot of flexibility for the use of the preprocessed data and retains the possibility to use other programs for the modelling process as well.
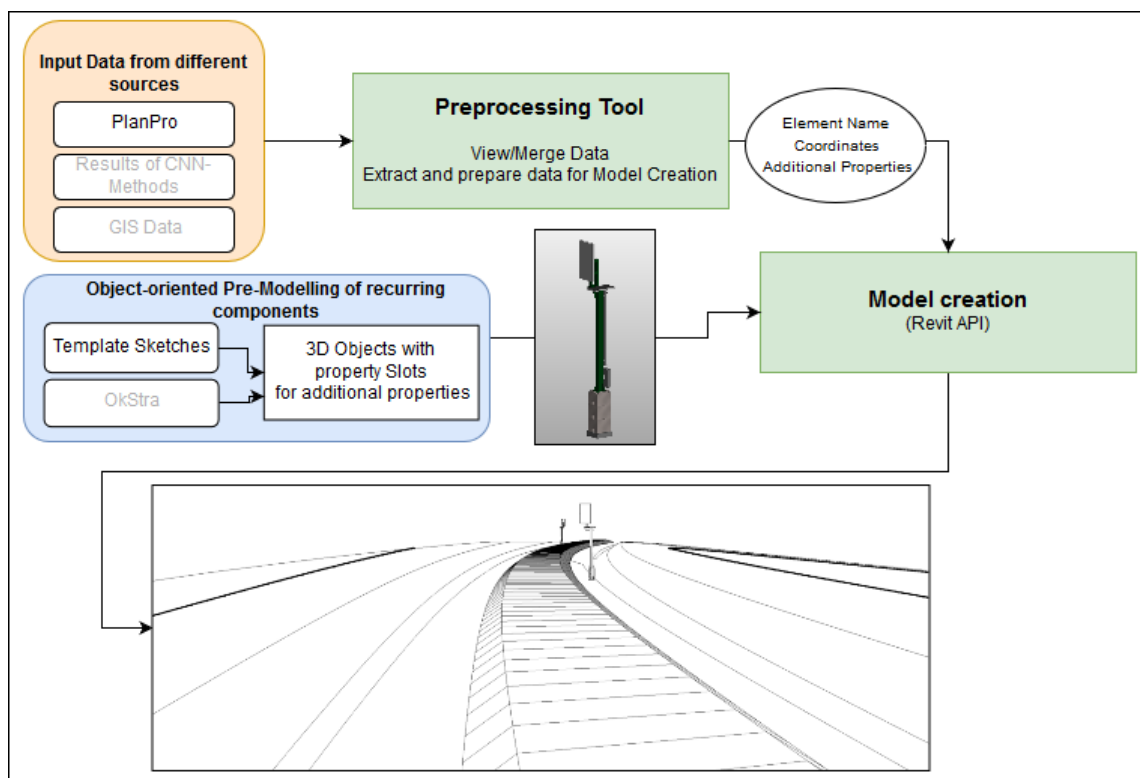

Figure 6: Workflow of creating BIM models of railway equipment infrastructure from various data sources by using predefined object-oriented model part templates.

The tool will serve as a testing platform in the research project to investigate how we can use and combine existing data in order to create a functioning BIM model.

5 SUMMARY AND OUTLOOK

This paper gives a general overview of the RIMcomb research project that aims at introducing BIM methods and technologies into the domain of railway equipment engineering. In addition to the main project goals, we discussed the ongoing research and first findings. In this scope, the automatic recognition of technical symbols in technical drawings by using different methods of image recognition is described in detail. Our results show that this method has large potential in automating a labor-intensive task and can also be employed to create semantic models of railway infrastructure. As a machine-learning based approach has the highest rates of correct recognitions we aim at improving this method in further research. This is especially important for the comparison of elements depicted in technical drawings with real-world video data gathered from railway track inspections.

REFERENCES

Allah Bukhsh, Z., Hartmann, T. & Stipanovic, I., 2016. From Analysis of Information Needs towards an Information Model of Railway Infrastructure. 23rd International Workshop of the European Group for Intelligent Computing in Engineering, Krakow, Poland.

buildingSMART, 2016. Standards Library, Tools and Services - buildingSMART. https://www.buildingsmart.org/standards/standards-tools-services/ (accessed 5.15.18).

Hu, M.-K., 1962. Visual pattern recognition by moment invariants. IRE transactions on information theory, 8(2):179–187. https://doi.org/10.1109/TIT.1962.1057692

Kaehler, A. & Bradski, G., 2016. Learning OpenCV - computer vision in c++ with the opencv Library. O'Reilly Media.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P., 1998. Gradient-based learning applied to document recognition, in: Proceedings of the IEEE. pp. 2278–2323. https://doi.org/10.1109/5.726791

Preidel, C. & Borrmann, A., 2016. Towards code compliance checking on the basis of a visual programming language. ITcon 21, 402–421.

Rohrer, B., 2016. How do Convolutional Neural Networks work? https://brohrer.github.io/how_convolutional_neural_networks_work.html (accessed 5.15.18).

Stoitchkov, D., 2018. Analysis of Methods for Automated Symbol Recognition in Technical Drawings. Bachelor's thesis, Technical University of Munich.

Vilgertshofer, S., Jubierre, J.R. & Borrmann, A., 2016. IfcTunnel - A proposal for a multi-scale extension of the IFC data model for shield tunnels under consideration of downward compatibility aspects, in: Proc. of the European Conference on Product and Process Modeling (ECPPM). Limassol, Cyprus.

Viola, P. & Jones, M., 2001. Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Comput. Soc. https://doi.org/10.1109/CVPR.2001.990517