Informed Machine Learning Methods for Instance Segmentation of Architectural Floor Plans

Alexander Hakert[1] and Phillip Schönfelder[1]
[1]Department of Civil and Environmental Engineering, Ruhr University Bochum,
Universitätsstraße 150, 44801 Bochum, Germany
E-mail(s): alexander.hakert@rub.de, phillip.schoenfelder@rub.de

Abstract: Architectural floor plans can be a valuable source of information for reconstructing digital models for existing buildings. For instance, room geometry and topology are essential to create building models and therefore need to be extracted from drawings. In case the drawings only exist as raster-image files (e.g., from scanning), engineers have to be assigned the tedious task of converting the pure image data into geometrically rich models. Driven by advances in deep learning-based image processing methods and the need to automate the extraction of floor plan geometry, earlier studies subjected floor plan images to semantic segmentation and instance segmentation. This research focuses on potential improvements to these approaches, primarily motivated by informed machine learning (IML) concepts. In this regard, expert knowledge is integrated to accelerate the training process and improve inference performance. A baseline Mask R-CNN model is trained on an open-source dataset and is compared to models enhanced with IML techniques. Among others, the effects of (1) a neighboring pixel loss, (2) a weighted cross-entropy loss, and (3) human preprocessing are investigated. Finally, the most promising techniques are combined to train a model which outperforms the baseline by 10.1% in the average precision score. In general, all of the proposed IML techniques improve segmentation results and can be considered in the development of future methods for segmenting floor plan images.

Keywords: BIM, Deep Learning, Informed Machine Learning, Instance Segmentation, Floor Plan

1 Introduction

The idea of building information modeling (BIM) [1] includes the development and maintenance of a digital building model which contains geometric and semantic information about a given structure. While BIM models for new buildings may be created during planning, models are rarely available for existing buildings. Therefore, in order to ensure efficient planning of maintenance, extension or deconstruction tasks, the necessary building documentation files must be collected, and the contained information must be manually assembled to form a digital model. An important type of raw data is

2D drawings, e.g., the floor plans of buildings, which are usually available as raster graphics from scanning paper drawings. Individual elements, such as walls, doors or rooms, must be segmented in the raster graphics and converted to a vector graphic in order to be transformed into BIM model elements with correct geometry.

Various methods have already been developed to automatically extract information from drawings: For the task of segmenting floor plan images, more and more data-driven approaches using convolutional neural networks (CNN) are applied in the automated creation of BIM models. Dodge et al. [2] test different architectures of fully convolutional network for the segmentation task and obtain state-of-the-art accuracies in the segmentation of walls. Liu et al. [3] use a CNN as part of a pipeline for vectorizing raster graphics. The CNN segments and detects connection points of walls, which are combined into vectorized walls in further steps. Also, Surikov et al. [4] use multiple machine learning approaches to vectorize floor plan images: A UNet model is used to efficiently segment walls and a Faster R-CNN model is used to detect windows and doors. Jang et al. [5] use a version of the DeepLab architecture to segment walls, which are then vectorized in further steps along with doors. Kim et al. [6] present a solution to the problem of analyzing floor plans of different size and complexity. They divide the floor plans into segments of fixed size and use a CNN for element detection. The authors of [7] also deal with complex floor plans by presenting a method that converts floor plans of different drawing styles to a unified format using an adversarial network and then vectorizes them using a junction extraction method. While many of the methods use segmentation and vectorization of walls to identify rooms, in other models rooms are directly segmented by CNNs: For instance, Kalervo et al. [8] present a method that segments and vectorizes floor plans using a multi-task CNN. Ramasamy et al. [9] use a Cascade Mask R-CNN in combination with a keypoint CNN to improve detection results.Chen et al. [10] also use a Mask R-CNN to segment rooms. They then perform a coordinate descent on each room, which eventually results in a vector graphics version of the floor plan, i.e. geometries are described by fundamental primitives. Recent approaches show promising results in the automatic conversion of raster-image floor plans to vector graphics. However, the precise segmentation of floor plan drawings remains a challenge due to the diversity in complexity and style of actual floor plans.

The concept of IML describes the integration of external knowledge into the machine learning pipeline. The knowledge can originate from different sources and may be integrated into the pipeline in various ways. Potentially, these techniques may lead to shorter training time, i.e. faster convergence, and better inference performance, without altering or expanding the training dataset. This makes it possible to train better models, given the same data, with only some tweaks to the training process, the model architecture, or the preprocessing. For more information, the reader is referred to [11], which defines a taxonomy on the topic and provides many examples for the use of IML.

This study focuses on the aspect of instance segmentation and the influence of IML methods on the performance of the chosen model. It proposes possible enhancements to the automated segmentation of floor plan drawings by inducing expert knowledge into the training process via four informed machine learning (IML) techniques: (1) a pixel neighborhood loss, (2), a weighted loss function to account for an uneven distribution of elements, (3) manual preprocessing of floor plan drawings, and (4) prior knowledge about topological relationships of elements. To test the effect of the IML approaches, a baseline Mask R-CNN model [12] is trained and tested on images taken from a publicly available

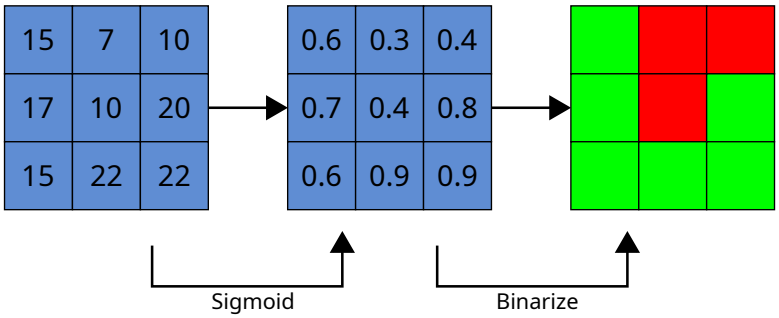

Figure 1: Procedure for determining a pixel weight for the neighbor loss.

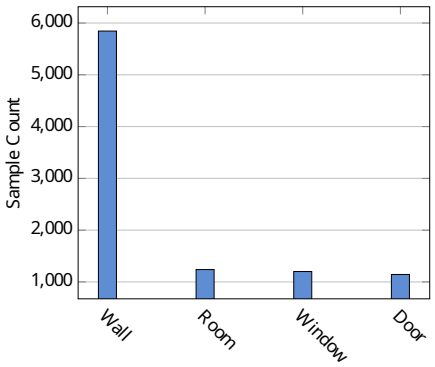

Figure 2: Number of samples per class in the CVC-FP dataset [13].

dataset. The performance scores of the baseline model and the IML models are compared to quantify the effects of the IML techniques. The presented methods can be used to develop future floor plan segmentation methods, contributing to the automated creation of BIM models. The paper is organized as follows: The used IML techniques are described in Section 2 and the performance of the accordingly modified models is evaluated in Section 3. Section 4 concludes the study and addresses possible future studies.

2 Methodology

2.1 Pixel Neighborhood

The concept of pixel neighborhood is introduced to take the connections of neighboring pixels of the floor plan image into account. In most cases, rooms have a large, simple polygon shape and are bounded by walls and other elements in straight lines. Therefore, irregularities (i.e., deviations from rectangular or similar geometries) can be considered unlikely and, thus, neighboring pixels are usually expected to belong to the same class. In order to incorporate this piece of external knowledge in the training process, Yuan and Xu [14] introduce a *neighbor loss*, which is used as a replacement for the Mask loss in this work. The idea of neighbor loss is to weight the individual pixels depending on their neighborhood when determining the binary cross entropy loss. In the first step, a sigmoid function is applied to all logits, normalizing the values to a range between 0 and 1. Then, the individual pixel values are binarized with a threshold of 0.5. The weight w_i of each pixel i is finally determined by the number of differently classified neighboring pixels and multiplied by a factor K, i.e.,

$$w_i = \max\left(1, K\left|\sum_{j\in 3\times 3} D_j M_j\right|\right), \quad \text{with } D = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}, \tag{1}$$

where M is the binarized neighborhood mask and D is a constant matrix to count the neighbors different to the pixel. The process is repeated for each pixel of the Mask R-CNN's output mask of size $m \times m$. The resulting weight matrix W of the pixels is then incorporated into the mask loss

$$L_{mask}(t, y, W) = -\frac{1}{m^2} \sum_{i \in m \times m} W_i * (t_i * \log(\varphi(y_i)) + (1 - t_i) * \log(1 - \varphi(y_i))), \tag{2}$$

where t is the ground truth mask, y is the output from the mask branch, and φ is the sigmoid function. The factor K is a hyperparameter and, thus, may be optimized with respect to the resulting model performance. Within this study, it is set to K = 10, inspired by [14].

2.2 Class Weights

This approach addresses the distribution of class occurrences within the dataset. In this study, the CVC-FP dataset [13] is used for all experiments. The wall class strongly dominates the dataset, amounting to five times as many samples as the other classes. This imbalance can cause the model to develop a tendency towards neglecting rooms, windows, and doors, while showing satisfactory performance for wall objects. As a remedy, the loss of the class branch L_{cls} is adjusted so that it accounts for the different number of samples per class. In the training process, the number of samples per class in the ground truth of each image is counted. Then, as described by Shrivastava [15], the weights for each class are determined by

$$w_c = \frac{1}{\sqrt{n_c}}, \tag{3}$$

where w_c is the weight for class c and n_c is the number of samples of that class in a given image. The *weighted loss* L_{cls} for each region of interest (RoI) is then determined by

$$L_{cls}(p, w, k) = -w_k * \log p_k, \tag{4}$$

where p is the probability distribution of classes at the output of the class branch, w is the weight vector determined by equation 3, and k is the ground truth class of the RoI.

2.3 Manual Preprocessing

As a further approach, it is investigated how the pre-processing of floor plan images by a user affects the model's performance. Floor plan images often measure up to several thousand pixels in height and width. If these images are then scaled to a size of 512×512 pixels to be fed into the model for segmentation, floor plan elements such as walls, doors, or windows may become too small to be detected and segmented. To avoid this information loss during scaling, the floor plan images are preprocessed by the expert user. The scanned images usually contain not only the actual floor plan, but also additional information, such as title blocks, descriptions, tables, or simply whitespace. Since this information is not needed for the task at hand, it can be removed by the user by simply cropping

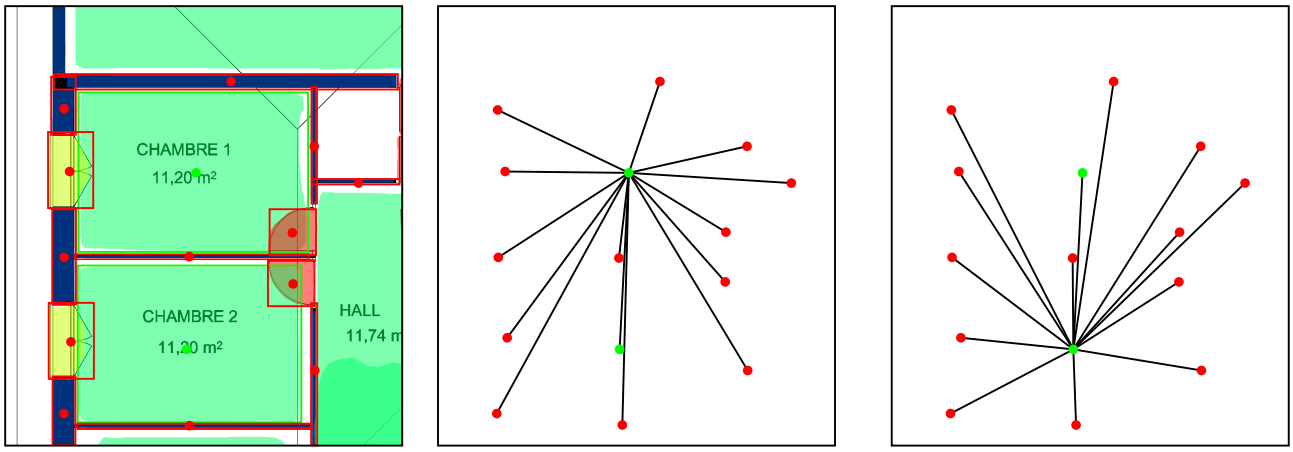

Figure 3: Visualization of $L_{neighborhood}$ computation. a) display of segmented objects with bounding boxes, b) and c) distances from room 1 and 2, respectively, to other objects.

the floor plan out of the scanned image. In this work, for the training process of the Mask R-CNN, the user's pre-processing is simulated by automatically cropping the ground plan of the images of the dataset used. To do this, the ground truth segmentation map of the images is used to cut out the floor plans with a random distance of 50 to 150 pixels from the outermost elements.

2.4 Class Neighborhood

Floor plans exhibit various topological relationships between the depicted elements. For example, doors and windows are connected to walls, and rooms are enclosed by several walls. It follows that two rooms are typically separated by elements that do not belong to the room class. To incorporate this structural relationship into the Mask R-CNN training process, an additional *class neighborhood loss* function is proposed for the model. During training, the model computes bounding boxes and class predictions for all the RoIs of an image. The predicted positions and classes are then used for the loss computation. Figure 3 shows an example of the steps to determine the loss. First, the class for each RoI is predicted. Second, the respective bounding box centers are determined. Finally, the Euclidean distances between the center points of all detected room objects are determined and sorted in ascending order according to the distance to the room object under consideration. The C nearest neighbors of each room object determine the loss. The hyperparameter C is set to 10. For the loss, the number of rooms in the immediate vicinity of the room under consideration is counted. The process is repeated for all detected rooms of the image. The counted room neighbors are summed up and then averaged over the number of considered rooms. The loss $L_{neighborhood}$ is thus determined by

$$L_{neighborhood}(E, C) = \frac{1}{|E|} \sum_{e \in E} C_k, \tag{5}$$

where E is the set of detected room elements of the image and C_k is the number of rooms among the 10 nearest neighbor objects of room e.

3 Results

For the experiments in this work, the CVC-FP dataset [13] is used to train and to test the Mask R-CNN models. The dataset consists of 122 scanned floor plans with different sizes, images qualities and drawing styles, which are annotated with object segmentation masks. In particular, it contains 5845 walls, 1239 rooms, 1201 windows, and 1144 doors. The image sizes in the dataset range from 1098 $\times$ 905 pixels to 7,383 $\times$ 5,671 pixels. For all experiments, 102 images of the dataset are used for training, 10 images are used for validation, and 10 images are used for testing. Three consecutive experiments are performed using a publicly available Mask R-CNN implementation [16]. First, it is investigated how the described IML approaches affect the instance segmentation performance of the Mask R-CNN. For this purpose, they are compared to a baseline version of the model. In a second experiment, multiple approaches are applied simultaneously to investigate which combination of approaches can further improve the model. Finally, the performance of the Mask R-CNN is increased through a longer training process.

To test the effect of the IML approaches on the performance of Mask R-CNN, the baseline model is modified according to each technique. For the first two experiments, the respective models are trained for 50 epochs with a learning rate of 0.001, using 100 RoIs for each image. The model is initialized by taking the weights of a model pre-trained on the COCO dataset. For the last experiment, the models are trained for 150 epochs with a lower learning rate of 0.0001. Table 1 shows the results of the evaluation of each experiment. The authors choose the metrics AP_{50} with the IoU threshold 0.5 as a measure for detection accuracy, AP with IoU thresholds 0.5–0.95 in steps of 0.05 for potential comparison with other methods, and AP_{tight} with IoU threshold 0.75–0.95 as a metric for accurate segmentations. It is noted that all methods improve the basic model in certain aspects. The most notable improvement is observed when using user-based preprocessing of the floor plan images. The best result is achieved by combining pixel neighborhood and preprocessing, resulting in an AP_{50} score of 76.2%. Figure 5 shows the validation performance of each model during the longer training process. The use of the pixel neighborhood has the highest impact on the AP_{tight}, presumably, due to the generation of more regularly shaped masks. Figure 4 shows exemplary segmentations of a floor plan with the baseline model and a model with all presented IML techniques combined. The latter achieves a segmentation performance of 46.7% AP and 75.9% AP50, exceeding the baseline model performance by 9.9% and 10.4%, respectively.

4 Conclusion

In this work, expert knowledge in the form of four different IML approaches is integrated into the training process of a Mask R-CNN, for instance segmentation of floor plan images. The used methods include three different loss functions and the preprocessing of floor plan images. It is evident that the integration of expert knowledge has a strong positive effect on the model's performance. Combining methods improves the AP_{50} and AP_{tight} scores up to 16.3% and 52.9% AP_{tight}, respectively, relative to the baseline model. For better performance, the methods can be optimized and extended in future

Table 1: Test results of the baseline model and the modified versions in percent. Bold values indicate the best performance scores among the tested models, given a fixed number of training epochs. The most promising model configurations are trained for 150 epochs.

	E = 50				E = 150			
IML approach	AP	AP_{tight}	AP_{50}	mIoU	AP	AP_{tight}	AP_{50}	mIoU
Original	28.6	11.6	51.1	36.7	36.8	15.7	65.5	55.4
Neighbor	29.3	13.3	50.6	38.9	-	-	-	-
Weights	29.9	12.1	54.5	41.9	-	-	-	-
Crop	32.0	12.7	58.1	44.3	42.9	19.2	74.1	58.0
ClsNeighbor	29.0	11.5	53.8	39.7	-	-	-	-
Neighbor+Weights	29.3	12.6	51.5	37.5	-	-	-	-
Neighbor+Crop	35.3	16.4	61.2	47.2	**46.9**	**24.0**	**76.2**	**61.4**
Weights+Crop	36.0	**16.6**	61.4	44.1	43.3	19.6	75.0	59.7
ClsNeighbor+Crop	**36.7**	16.5	**63.7**	**48.0**	44.4	21.7	74.0	59.5
All	36.3	16.2	63.3	47.5	46.7	23.4	75.9	58.3

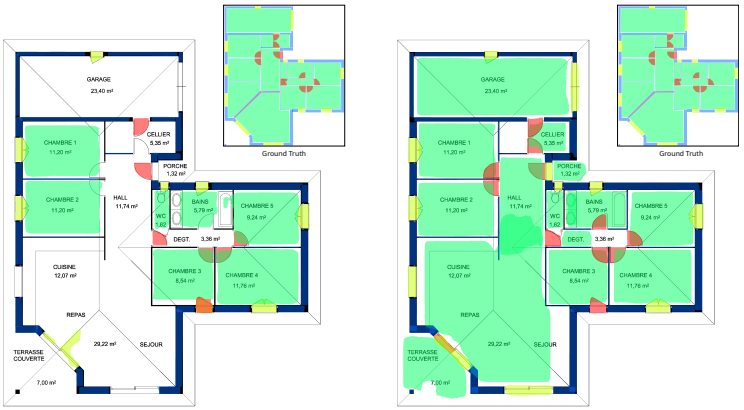

Figure 4: Example segmentations with the original model on the left and the combination of all approaches on the right. Floor plan image taken from the CVC-FP dataset [13].

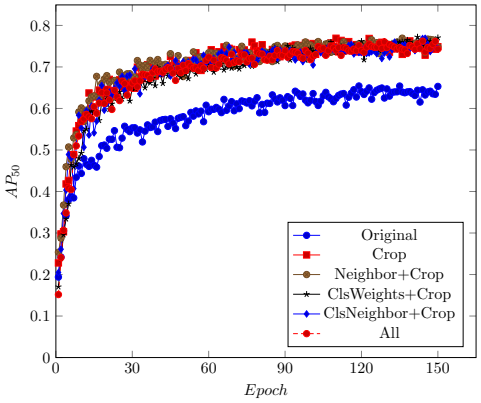

Figure 5: The models' AP_{50} scores on the validation set during the training of the different models.

works. The presented approaches to integrate expert knowledge show great potential and should be considered for the development of new floor plan segmentation methods.

Acknowledgements

This research is conducted as part of the BIMKIT project, funded by the German Federal Ministry for Economic Affairs and Climate Action.

References

[1] A. Borrmann, M. König, C. Koch, and J. Beetz, editors, *Building Information Modeling: Technology Foundations and Industry Practice*. 2018. DOI: 10.1007/978-3-319-92862-3.

[2] S. Dodge, J. Xu, and B. Stenger, “Parsing floor plan images”, in *15th Int. Conf. on Machine Vision Appl.*, 2017, pp. 358–361. DOI: 10.23919/MVA.2017.7986875.

[3] C. Liu, P. Kohli, J. Wu, and Y. Furukawa, “Raster-to-vector: Revisiting floorplan transformation”, in *IEEE Int. Conf. on Comput. Vision*, 2017, pp. 2214–2222. DOI: 10.1109/ICCV.2017.241.

[4] I. Y. Surikov, M. A. Nakhatovich, S. Y. Belyaev, and D. A. Savchuk, “Floor plan recognition and vectorization using combination UNet, Faster-RCNN, statistical component analysis and Ramer-Douglas-Peucker”, in *Int. Conf. on Comput. Sci., Commun. and Security*, vol. 1235, 2020, pp. 16–28. DOI: 10.1007/978-981-15-6648-6_2.

[5] H. Jang, K. Yu, and J. Yang, “Indoor reconstruction from floorplan images with a deep learning approach”, *Int. J. of Geo-Information*, vol. 9, p. 65, 2020. DOI: 10.3390/ijgi9020065.

[6] H. Kim, S. Kim, and K. Yu, “Automatic extraction of indoor spatial information from floor plan image”, *Int. J. of Geo-Information*, vol. 10, p. 828, 2021. DOI: 10.3390/ijgi10120828.

[7] S. Kim, S. Park, H. Kim, and K. Yu, “Deep floor plan analysis for complicated drawings based on style transfer”, *J. of Comput. in Civil Eng.*, p. 04 020 066, 2021. DOI: 10.1061/(ASCE)CP.1943-5487.0000942.

[8] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, “Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis”, pp. 28–40, 2019. DOI: 10.1007/978-3-030-20205-7_3.

[9] A. Ramasamy Arunkumar, S. Mohan, and J. Kumar, “Segmentation of spatial and geometric information from floorplans using CNN model”, *Turkish J. of Comput. and Math. Edu.*, vol. 12, pp. 1909–1920, 2021. DOI: 10.17762/turcomat.v12i9.3620.

[10] J. Chen, C. Liu, J. Wu, and Y. Furukawa, “Floor-SP: Inverse cad for floorplans by sequential room-wise shortest path”, in *IEEE/CVF Int. Conf. on Comput. Vision*, 2019, pp. 2661–2670. DOI: 10.1109/ICCV.2019.00275.

[11] L. von Rueden, S. Mayer, K. Beckh, *et al.*, “Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems”, *IEEE Trans. on Knowl. and Data Eng.*, pp. 1–1, 2021. DOI: 10.1109/TKDE.2021.3079836.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN”, in *IEEE Int. Conf. on Comput. Vision*, 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

[13] L.-P. de las Heras, O. R. Terrades, S. Robles, and G. Sánchez, “CVC-FP and SGT: A new database for structural floor plan analysis and its groundtruthing tool”, *Int. J. on Document Anal. and Recognition*, vol. 18, pp. 15–30, 2015. DOI: 10.1007/s10032-014-0236-5.

[14] W. Yuan and W. Xu, “NeighborLoss: A loss function considering spatial correlation for semantic segmentation of remote sensing image”, *IEEE Access*, vol. 9, pp. 75 641–75 649, 2021. DOI: 10.1109/ACCESS.2021.3082076.

[15] I. Shrivastava, *Handling class imbalance by introducing sample weighting in the loss function*, 2020. [Online]. Available: https://medium.com/gumgum-tech/handling-class-imbalance-by-introducing-sample-weighting-in-the-loss-function-3bdebd8203b4 (visited on 04/05/2022).

[16] A. Kelly, *Mask r-cnn for object detection and instance segmentation on keras and tensorflow*, https://github.com/akTwelve/Mask_RCNN, 2020.