

Multi-Unit Floor Plan Recognition and Reconstruction Using Improved Semantic Segmentation of Raster-Wise Floor Plans

Lukas Kratochvila^{1*}†, Gijs de Jong^{2†}, Monique Arkesteyn³, Šimon Bilík^{1,4}, Tomáš Zemčík¹, Karel Horák¹, Jan S. Rellermeyer^{2,5}

¹*Department of Control and Instrumentation, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic.

²Department of Software Technology, Faculty of Electrical Engineering Mathematics, and Computer Science, TU Delft, Delft, The Netherlands.

³Department of Management in the Built Environment, Faculty of Architecture and the Built Environment, TU Delft, Delft, The Netherlands.

⁴Computer Vision and Pattern Recognition Laboratory, Department of Computational Engineering, Lappeenranta-Lahti University of Technology LUT, Lappeenranta, Finland.

⁵Dependable and Scalable Software Systems, Institute of Systems Engineering, Faculty of Electrical Engineering and Computer Science, Leibniz University Hannover, Hannover, Germany.

*Corresponding author(s). E-mail(s): kratochvila@vut.cz; Contributing authors: gijs-dejong@live.nl; m.h.arkesteyn@tudelft.nl; bilik@vut.cz; zemcikt@vutbr.cz; horak@vut.cz; rellermeyer@vss.uni-hannover.de;

†These authors contributed equally to this work.

Abstract

Digital twins have a major potential to form a significant part of urban management in emergency planning, as they allow more efficient designing of the escape

routes, better orientation in exceptional situations, and faster rescue intervention. Nevertheless, creating the twins still remains a largely manual effort, due to a lack of 3D-representations, which are available only in limited amounts for some new buildings. Thus, in this paper we aim to synthesize 3D information from commonly available 2D architectural floor plans. We propose two novel pixel-wise segmentation methods based on the MDA-Unet and MACU-Net architectures with improved skip connections, an attention mechanism, and a training objective together with a reconstruction part of the pipeline, which vectorizes the segmented plans to create a 3D model. The proposed methods are compared with two other state-of-the-art techniques and several benchmark datasets. On the commonly used CubiCasa benchmark dataset, our methods have achieved the mean F1 score of 0.86 over five examined classes, outperforming the other pixel-wise approaches tested. We have also made our code publicly available to support research in the field.

Keywords: Floor plan digitalization, Digital twins, Semantic segmentation, 3D reconstruction, Vectorization

1 Introduction

Urban areas experience a steady growth, and over two-thirds of the population worldwide will be considered urbanized by the year 2050 [1]. Heavy and complex traffic flows and multiple buildings of various ages and structural arrangements make emergency planning against events such as a fire or flood very challenging. Even though the emergency operators usually have access to the building schemes and floor plan drawings, the individual embodiments of these documents may vary markedly, meaning that, for instance, the paper sheet rendering differs textually and graphically from that of the electronic version (such as a PDF file). Quick understanding of the 2D drawings during a rescue task might then be challenging due to not only these differences but also a high number of the records and their lower clarity compared to the 3D space representation.

A promising way to change this situation lies in using the digital twin of the relevant city, as demonstrated on the examples of Cambridge campus, [2] and an office building facade element, [3]; the approach is also discussed in an extensive overview centered on the individual aspects and applications, [4]. The digital twin models could become an important part of urban management, improving the response to emergencies and making safety planning more direct and efficient. The majority of existing digital twin systems, however, were developed only through simulation or for recently built structures with an existing 3D model.

The research presented in this paper has been conceived to close the above-outlined gap by formulating a novel procedure to generate automatically a 3D representation of buildings from conventional 2D-floor plans, which are usually available from the competent municipal authorities. As the option proposed herein only estimates the height of the floors, it could be termed a 2.5D model by some authors. The 3D models are seamlessly integrable into the digital twins to model entire cities in a scalable and

economical manner, unlike using a point cloud as shown, for example in [5], which is computationally very demanding and with noisier results. In generating such a 3D model from the 2D plans, an essential issue rests in reliable recognition of critical structural elements in floor plans, including walls, windows, stairs, and railings. While numerous approaches are in use presently, a substantial number of these assume the 2D representation to already come in the form of vectors [6], a case not typical of older buildings. Other approaches do not rely on this assumption but aim merely at relatively small-sized plans of apartments and smaller units, not covering entire multi-unit floors or buildings.

The central goals of the research can be summarized as follows:

- developing a novel recognition and reconstruction pipeline that involves semantic-wise pixel segmentation, recognition of structural elements, and 3D reconstruction;
- evaluating the proposed segmentation method with other state-of-the-art (SOTA) techniques;
- testing the method on various datasets consisting of multi-unit floor plans and various drawing standards;
- publishing the framework to support further investigation in the domain.¹.

2 Related Work

Automated recognition of architectural plans and drawings embodies a challenging problem which may benefit multiple fields and subdomains, such as creating virtual twins of buildings [2], optimal evacuation path planning [7], and firefighting. As applicable architectural plans are often unavailable electronically or do not exist at all, digitizing older drawings or the common floor and evacuation schemes might be a necessary precondition for this task.

The first attempts to facilitate automated recognition of architectural plans and drawings were made more than 30 years ago, as presented in paper [8]. Other procedures, utilizing classic computer vision techniques such as segmentation, recognition, and classification, appeared at the beginning of the new millennium; these options included, for instance, innovative systems [9], binary image operations [10, 11], the Hough transformation [12], and a combined approach comprising structural and semantic analysis complemented with OCR-retrieved information [13]. These methods might not focus only on the reconstruction from the 2D plans, but also from measured data in form of 3D point cloud as shown for example in survey of such methods [14]. This approach might be advantageous in the case of non-existing plans of older buildings. Although other segmentation methods have been designed [15], we focus especially on advances in convolutional neural networks (CNNs) after the introduction of AlexNet in 2014 [16]. A compressive survey of deep segmentation networks, their applications and future directions is shown in [17]. As deep learning-based techniques have found use in executing the tasks characterized herein, proving to be very efficient in semantic segmentation, we decided to opt for deep learning.

A CubiCasa5K-related paper [18] introduces a new, large dataset of 5,000 floor plans with more than 80 annotated classes; by extension, an original multi-task model

¹Our code: <https://anonymous.4open.science/r/ensembles-7946/>

is outlined. Exploiting the previously published segmentation network from [19], the author adds a trainable module to adjust the weights in the multi-task loss calculation. Paper [20] exposes a complex framework combining wall segmentation based on DeepLabV3+ [21] and opening detection with a YOLOv4 [22] object detector. The results of the research comprise a novel scale calculation relying on floor plan annotations to determine the size of a 3D representation. The whole method was tested on an in-house compiled dataset. Compared to the solution in [23], the approach increases the accuracy of both the wall and the room segmentation types.

In [24], the aim lies in converting a 2D raster drawing into 3D data through a deep neural network technique on a limited dataset of 30 floor plans, utilizing the 3DPlanNet. To achieve this, they authors use a complex pipeline consisting of pattern recognition with a moving kernel to enable wall detection; object detection in corners; wall junction; and door, window, and room type recognition using the TensorFlow detection API followed by the generation of nodes, edges, and detected objects such as doors and windows. The approach was tested on a KOVI dataset, which, regrettably, is not described in greater detail. The research set out in [25] concentrates on an automatic method for segmenting raster-wise floor plans by means of learning-based hierarchical segmentation and assembling the obtained geometrical primitives into a planar structure. A 3D model generation from the floor plan is included in the proposed framework. A VGG [26] classifier to handle the objects and boundaries is followed by a custom semantic segmentation network with a loss function, respecting the boundaries and type of the room. The obtained geometrical primitives are identified and fused with the relation information into a planar graph using mixed integer programming. The proposed approach was tested on R2V [19] and R3D [27] datasets.

The authors of [28] propose a method for segmenting roughcast floorplans into individual rooms/regions. The proposed method represents the input vector or vectorized image as a graph, and uses a two-stream GNN (Graph Neural Network). The paper makes a good argument in favour of directly processing floorplan data in the vector domain. The method is trained and evaluated on the CubiCasa5k and R2V datasets. The method is not directly comparable to this paper, as the task is substantially different.

In the article [29], the authors suggest a reverse use case while using a mobile robot with 2D LIDAR and SLAM techniques to reconstruct 2D plans from the existing buildings in order to check their compliance with the plans. Such obtained 2D plan is matched with the original one and validated using a custom Euclidean distance-based error metric. This approach might be promising to combine with the reconstruction from 2D to 3D in the case of non-existing or incomplete floor plans to create a virtual twins.

In comparison with a majority of the current research works using the existing vector floor plans, we focus on the reconstructions from the raster-wise data while introducing a novel recognition and reconstruction pipeline. Although it means to add additional steps prior the vectorization as the scanning, semantic segmentation and filtering of the segmented plan, we believe that these steps are balanced with a possibility of processing the existing non-digitalized paper plans after scanning. Our

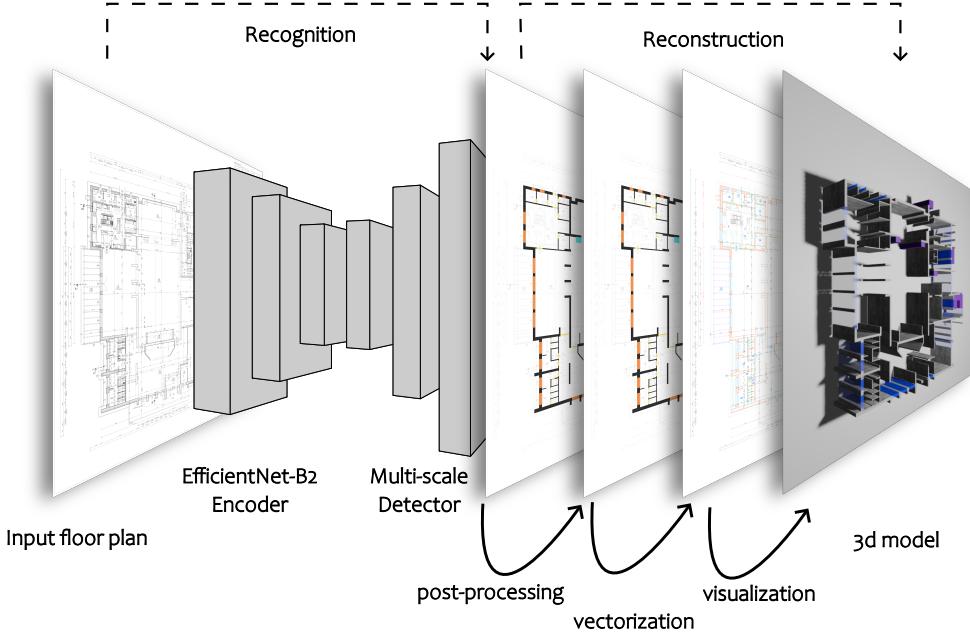


Fig. 1: The recognition part, exposed on the left-hand side, produces a segmentation mask of the floor plan by using a custom convolutional neural network (CNN). The segmentation mask is subsequently refined through the reconstruction step, shown on the right-hand side, which applies post-processing, vectorization, and Blender-based visualization.

framework is also publicly available and evaluated over well known dataset together with an ablation study of all parts of our suggested framework.

3 Proposed Methods

The structure of the designed floor plan processing pipeline is shown in Figure 1. The unit can be separated into a recognition part (on the left), which performs a pixel-wise semantic segmentation of the given floor plan, and a reconstruction sector (on the right), which further processes the segmented elements into a final 3D model.

3.1 Recognition

The recognition part exploits a custom CNN based on an MDA-UNet [30], MACU-Net [31], and U-Net3+ [32]; by extension, it presents an asymmetric convolution block, an attention mechanism, a multi-scale feature skip connections, and a multi-task training objective. The training objective is designed to capture a pixel, a patch, and multi-scale losses. Figure 2 depicts the two model architectures and their structural properties. In addition, we included different input image sizes to the training process to create a multi-scale detector that can process variously sized input images.

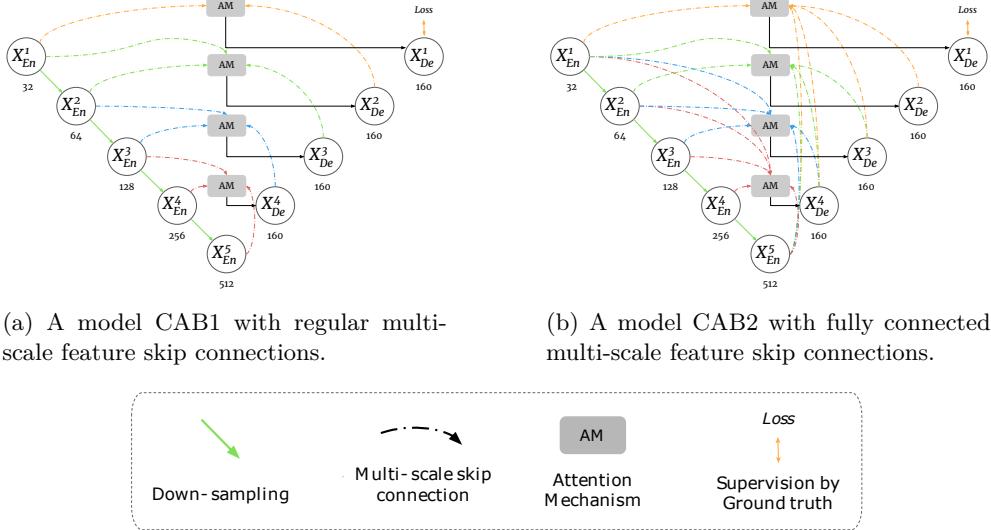


Fig. 2: Two model architectures: (a) regular multi-scale feature skip connections, and (b) fully connected multi-scale feature skip connections. Here, X_{En}^i means an encoder and X_{De}^i a decoder block.

3.1.1 Asymmetric Convolution Block

To better capture the learned feature maps, different types of convolution blocks are applicable. In addition to the normalization and activation functions paired with the convolution layers, the design and use of kernels play an important role in improving the performance of a CNN [33]. We propose an Asymmetric Convolution (AC) block, to replace the standard square convolutions. Asymmetric convolution blocks have been shown to enhance the skeleton features by replacing the standard $k \times k$ (k denotes the kernel size) kernels with an aggregation of the $k \times k$ kernel and two additional horizontal $1 \times k$ and vertical $k \times 1$ kernels [33]. Directional kernels have also been evaluated on floor plans; however, they do not deliver clear performance improvements compared to horizontal and vertical kernels [23, 34].

3.1.2 Attention Mechanism

Each of the proposed models utilizes an attention mechanism that combines a channel attention module with a spatial attention module to highlight the feature maps separately, based on their spatial or channel importance, similarly to the design proposed by [35]. The attention module infers a channel attention map M_c and a spatial attention map M_s from an intermediate feature F , which is subsequently weighed

through multiplication formalized by Equation (1). The final refined feature is computed by performing an asymmetric convolution on the concatenated channel and spatial refined feature.

$$AM(F) = AC \left(\left[F_c \otimes M_c(F), F_s \otimes M_s(F) \right] \right), \quad (1)$$

$$F_c = C^{1 \times 1}(F),$$

$$F_s = C^{1 \times 1}(F),$$

where F_c and F_s are the channel compressed representations of the intermediate feature map F using two 1×1 convolutions. Note that the spatial map M_s is derived from the full-sized feature F . The concatenation is denoted by square brackets, and \otimes denotes the element-wise multiplication.

Concatenating average and max-pooled feature maps has been shown as an effective step in reinforcing spatial features [30]. Different dilation rates have been used to capture the features from different-sized receptive fields, reinforcing small and large spatial features. The spatially enhanced features are aggregated through element-wise addition and normalized by the sigmoid function to produce the final spatial map.

The proposed attention mechanism incorporates spatial and channel attentions maps similar to those introduced in [35]. The spatial and channel attention modules (SAM, CAM) are exposed in Figure 3. Our asymmetric convolution resembles that outlined in [33], but the group normalization is used as shown in Figure 4.

Channel Attention Module

The disadvantage of directly aggregating full-scale features, as proposed by the U-Net3+, is that the weights of the channels of these features are assumed to be equal. To better utilize the features, a channel attention module is proposed to weigh the feature map $F \in \mathbb{R}^{H \times W \times C}$ according to the channel feature map $M_c \in \mathbb{R}^{1 \times 1 \times C/2}$, where C indicates the number of channels of the feature map. Multiplying the intermediate feature map with the channel feature map allows the model to reinforce the informative channels and to restrain the indiscriminative ones [36]. Our paper proposes an improved channel attention module of the MACU-Net [31] capable of refining feature blocks of an arbitrary size.

The channel feature map $M_c \in \mathbb{R}^{1 \times 1 \times C/2}$ is inferred along the spatial dimensions of an intermediate feature, as formalized by Equation (2). Due to the large number of channels in the original input feature map, a compressed version F_c having half the number of channels is first computed via 1×1 convolution. The spatial dimensions of F_c are then squeezed through separate adaptive average pooling and max pooling, producing F_c^{avg} and F_c^{max} of $1 \times 1 \times C/2$. Separate and simultaneous average and max pooling have been demonstrated to improve the representation power of a model, [35]. Important channels are extracted by compressing the pooled feature maps into one-sixteenth of their original size through a convolution and ReLU operation. To produce the channel feature map, the channel features are expanded to $C/2$ channels and

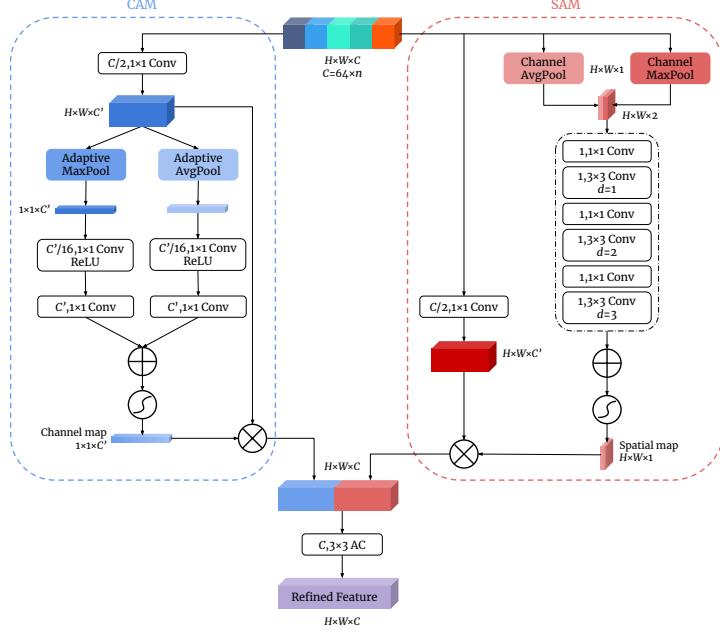


Fig. 3: An overview of the attention module, which consists of a channel attention module (CAM) and a spatial attention module (SAM). The symbols \oplus , \otimes and \odot denote element-wise addition, multiplication, and the sigmoid operation.

aggregated by element-wise addition. The merged channel map is normalized by the sigmoid function

$$M_c(F) = \sigma \left(C^{1 \times 1} \left(\text{ReLU} \left(C^{1 \times 1} \left(F_c^{\text{avg}} \right) \right) \right) + C^{1 \times 1} \left(\text{ReLU} \left(C^{1 \times 1} \left(F_c^{\text{max}} \right) \right) \right) \right) \quad (2)$$

where σ denotes the sigmoid function, ReLU stands for the ReLU activation, and C represents a convolution operation.

Spatial Attention Module

The purpose of a spatial attention module is to highlight major features by weighing the feature map $F \in \mathbb{R}^{H \times W \times C}$ according to the spatial feature map $M_s \in \mathbb{R}^{H \times W \times 1}$, where H , W , and C are the height, width, and number of channels of the feature map. The MDA-UNet has applied a variant of a spatial attention module to improve the fuse multi-scale features in the U-Net3+ [30]. A limitation of the spatial attention module in the MDA-UNet rests in that the element was designed to refine a single encoder feature. Thus, we propose an innovated spatial attention module (Figure 3) which considers both the regular and the fully connected multi-scale features.

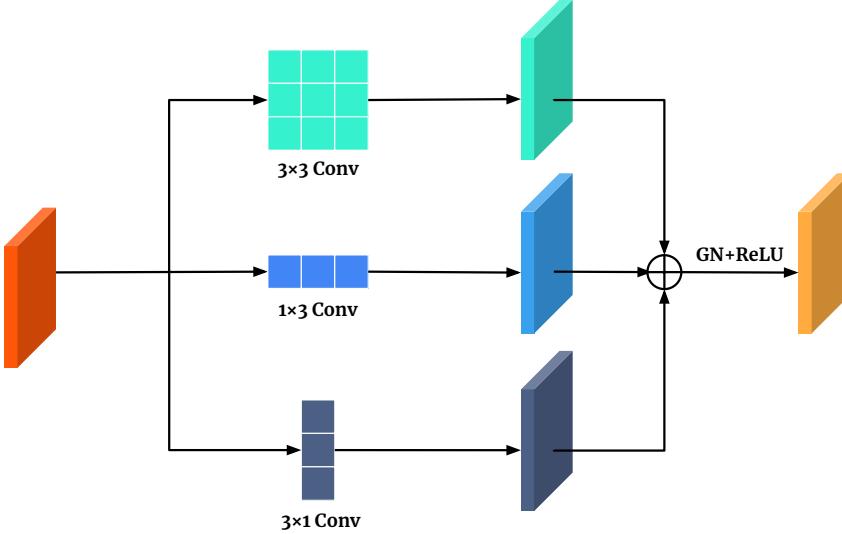


Fig. 4: An asymmetric convolution (AC) block using group normalization (GN) enhancing the skeleton features. The symbol \oplus represents element-wise addition.

A spatial attention module (SAM) aims to learn a spatial feature map $M_s \in \mathbb{R}^{H \times W \times 1}$ by inferring along the input feature's channel dimension as described by Equation (3). To compute the spatial attention map, average and max channel pooling [37] are first applied to produce $F_s^{\text{avg}} \in \mathbb{R}^{H \times W \times 1}$ and $F_s^{\text{max}} \in \mathbb{R}^{H \times W \times 1}$, respectively. Channel pooling is used thanks to its effectiveness in enhancing the spatial features [35]. The concatenated pooled feature maps are then passed to the spatial convolutional layer C_s , which enhances the spatial features by using different dilation rates.

Concatenating the average and max-pooled feature maps has been shown to be effective in reinforcing the spatial features [30]. Various dilation rates have been used to capture the features from differently sized receptive fields, reinforcing both the small and the large spatial features. The spatially enhanced features are aggregated through element-wise addition and normalized via the sigmoid function to deliver the final spatial map.

$$M_s(F) = \sigma \left(C_s \left(\begin{bmatrix} F_s^{\text{avg}}, F_s^{\text{max}} \end{bmatrix} \right) \right), \quad (3)$$

$$C_s(F) = \sum_{i=1}^3 \left(C^{1 \times 1}(F) + C^{3 \times 3, d=i}(F) \right)$$

where σ denotes the sigmoid function and C_s stands for the spatial convolutional block, respectively.

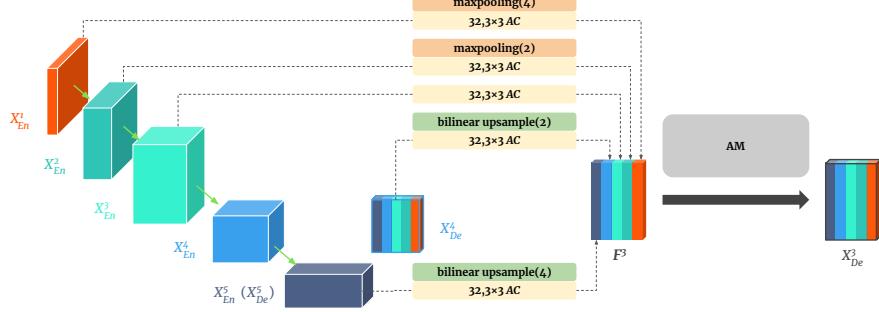


Fig. 5: An example of an intermediate feature map F^3 of the third decoder layer X_{De}^3 . The abbreviations AC and AM indicate an asymmetric convolution block and an attention mechanism; X_{En}^i denotes blocks of convolutions.

3.1.3 Multi-Scale Feature Skip Connections

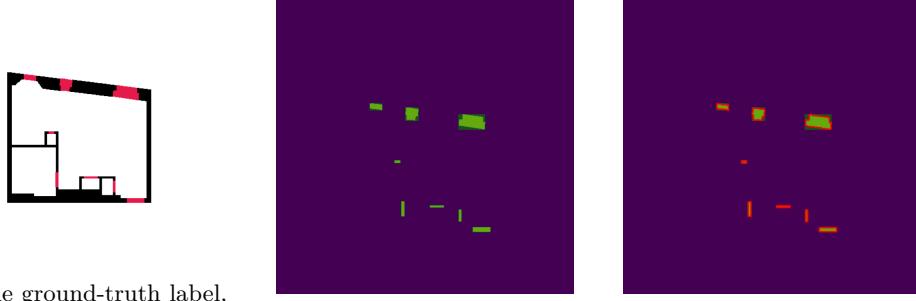
As the multi-scale feature skip connections have recently provided promising results, such as the UNet# network [38] which combines dense and full-scale skip connections and integrates different scale semantics of feature maps, our architectures incorporate them. Below, Figure 5 shows the construction of X_{De}^3 . The intermediate feature map F of the model architectures is structured similarly to that of the U-Net3+. For each up-sampling level $i < N$ and respective decoder X_{De}^i , the feature map F^i to be refined comprises a mix of differently scaled feature maps from the encoder and the decoder layers.

As shown in Figure 2, the first architecture CAB1 contains regularly connected multi-scale feature skip connections, where the individual encoder layers are combined with the previous (shallow) layer of the encoder. In the case of a decoder, the layers are combined with encoder layers of the same or previous depth and the previous (deeper) layer of the decoder. The second architecture, CAB2, uses the same encoder as the model CAB1; however, the decoder layers are combined with all previous (shallow) encoder layers and all previous (deeper) decoder ones.

3.1.4 Training Objective

Further, we propose a multi-task training objective, with each task designed for a specific purpose, namely, handling class imbalance, utilizing semantic relations in labels, and exploiting multi-scale predictions. We employ an asymmetric unified focal loss to handle class imbalance by suppressing the background loss and enhancing the foreground loss as described in [39].

Relying on a soft loss function can facilitate stabilizing the training by guiding the network during the training process [40, 41]. Heatmap regression can be an effective measure to compute such a loss, showing promising results for human pose estimation. Opening a regression loss and utilizing the predicted heatmaps to regress the boundaries of the opening are discussed in [20].



(a) The ground-truth label, boundaries, and openings highlighted in black and red, respectively.

(b) Opening the connected components highlighted in green.

(c) The contours of the connected components highlighted in red.

Fig. 6: The ground-truth opening label and the first two steps to derive the opening endpoints.

The difference between the proposed approach and the method presented by the authors of [20] lies in how the ground-truth heatmaps are created. In [20], the endpoints of the openings are employed to form the heatmaps, which are manually defined for their new dataset. Instead of manually adding these endpoints to the existing datasets, we derive them from the opening labels by further processing. A similar approach has been shown to reduce the memory footprint of a model significantly [42]. The average heatmap H_c for each pixel i is then defined as follows:

$$H_c(i) = \frac{1}{|B|} \sum_{\beta}^B H_{c\beta}(i) \quad (4)$$

where B is set of controlling value β , which controls the spread of the peak of the values in the heatmap. Generating the heatmap $H_{c\beta}$ for each β requires the following three steps, also exposed in Figure 6:

1. Find the connected components of category c using [43] as shown in Figure 6b.
2. Find the contour of each component of step 1 using [44] as shown in Figure 6c.
3. Given β , compute the value $H_{c\beta}$ of each pixel i of category c using Equation (5) as shown in Figure 7.

$$H_{c\beta}(i) = \max_{j \in \mathcal{O}_c} \exp\left(-\frac{\|y_i(c) - y_j(c)\|_2^2}{\beta^2}\right) \quad (5)$$

where \mathcal{O}_c is the set of endpoints of opening c , β controls the spread of the peak of the values in the heatmap, and y_i is ground truth pixel i , resp. y_j is ground truth pixel j , for endpoints of openings c .

The reason for using the average heatmap $\frac{1}{|B|} \sum_{\beta}^B (H_{c\beta}(i))$ is that the step motivates the model to keep refining its openings prediction. An example of such a combined heatmap is shown in Figure 8.

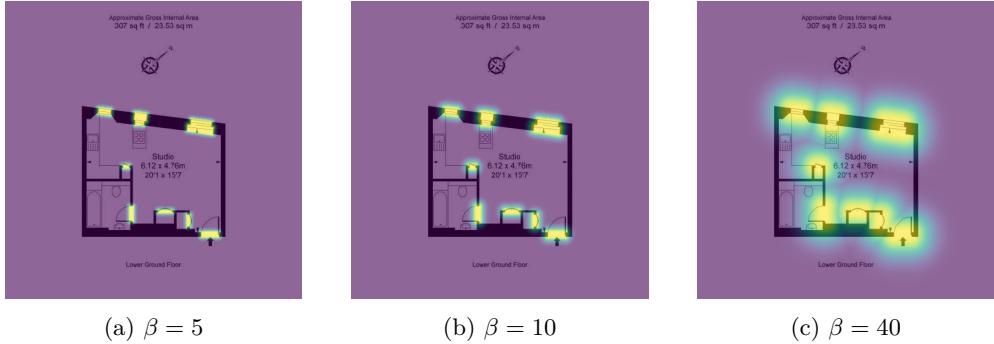


Fig. 7: Three examples of generated heatmaps of the opening class to increase the values for β , superimposed on the original floor plan.

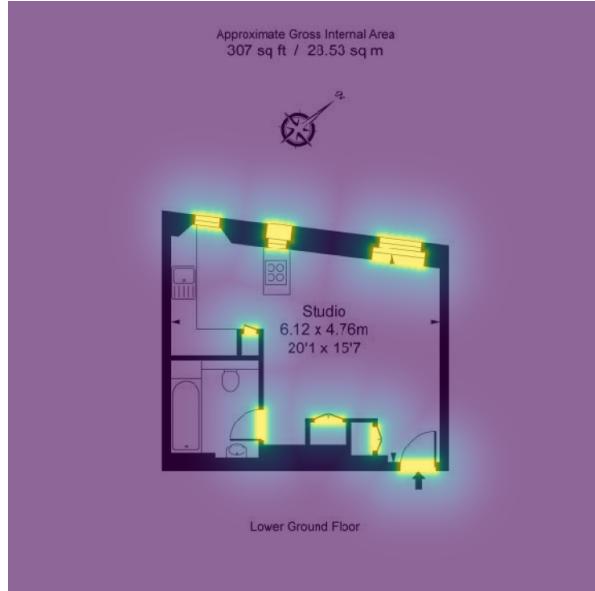


Fig. 8: An example of an average heatmap computed by using $B = \{5, 10, 40\}$ superimposed on the original floor plan.

The values controlling the spread in the heatmap in B should be chosen carefully: If they are set too high, the resulting heatmap values ≥ 0.5 outside of an opening could confuse the model trying to accurately detect it. Thus, the combined mean heatmap of values B should have a sharp decline towards 0.5, followed by a slower decrease in the value in order to facilitate learning. The mean heatmap with $B = \{2, 10\}$, as illustrated in Figure 9, can model this effectively.

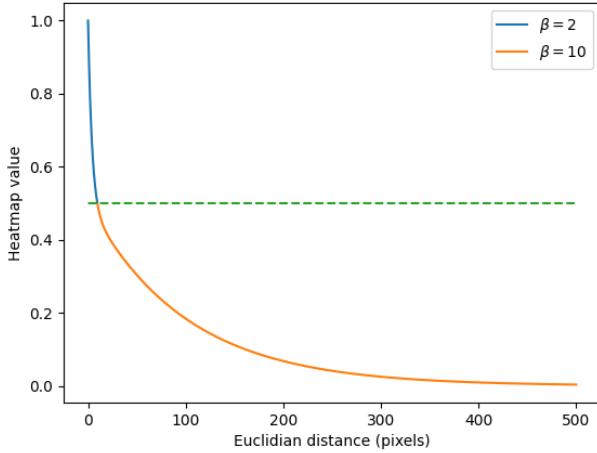


Fig. 9: The heatmap values to increase the distances with $B = \{2, 10\}$. The term $\beta = 2$, marked in blue, is responsible for the sharp decline towards 0.5, necessary to prevent contradictory learning. The term $\beta = 10$ ensures a slower decline after 0.5 to allow further learning. The heatmap value 0.5 has been marked in green for reference.

$$\mathcal{L}_{\text{MHR}} = \frac{1}{|O|} \sum_c^O \frac{1}{|\hat{y}|} \sum_i \|\hat{y}_c(i) - H_c(i)\|_2^2 \quad (6)$$

The final multi-task loss is the weighted sum of all previously defined loss functions, for which the weights are learned through training by using the approach proposed in [45].

The unified focal loss applies a form of unary supervision, which lacks the spatial discrimination to exploit the semantic structure in the labels. To capture this structure, we use the affinity field loss defined in [46].

The third loss, formalized in Equation (6), represents a heatmap regression task similar to that presented in [20]. This loss is of a distinct character, as the heatmaps are indirectly derived from the segmentation masks instead of being predicted by the model. Regarding the heatmaps, the heatmap regression loss \mathcal{L}_{MHR} is a simple average of the squared euclidean distance between the prediction map \hat{y}_c and the heatmap H_c over all the openings O and pixels i in the floor plan.

3.2 Reconstruction

The reconstruction part of the pipeline refines the segmentation masks from the recognition sector by applying three steps: post-processing, vectorization, and visualization. The post-processing method continuously refines the connected components in the segmentation mask until an fitting threshold is reached. Such refined components of the segmentation mask are transformed into polygons by deriving the endpoints of the

minimum area rotated rectangle of each component. The polygons are then refined by merging the vertices that are sufficiently close or collinear. The resulting set of polygons is used to visualize the segmentation mask with the Blender.

Algorithm 1 Polygon Approximation

Require: Joint mask \hat{y} , uncertainty threshold ϵ_u
 Find the initial connected components C in \hat{y} , using [43]
while Any C left **do**
 Find a rotated rectangle bb with the minimum area enclosing the component [47]
 Calculate the fitting score u_{bb} of bb according to Equation (7)
if u_{bb} is smaller than ϵ_u **then**
 Divide the bb into two smaller rectangles
 Add a new C of each new rectangle
else
 Add the approximate polygon P_i to the list of polygons P
end if
end while

3.2.1 Approximate Polygons

The first step to derive the approximate polygons from the segmentation mask rests in creating a joint mask consisting of the union of all the semantic classes. Such a joint mask is proceeded by algorithm 1:

The ϵ_u is a constant fitting threshold set to 0.5 determined empirically. The value was found to be high enough to produce accurate approximations and low enough to generalize excessively noisy predictions. The fitting score u_{bb} is defined in eq. (7).

$$u_{bb} = \left(\frac{1}{|bb|} \sum_{i \in bb} \hat{y}(i) \right) \quad (7)$$

3.2.2 Refined Polygons

The refined polygons are obtained by reducing the approximate ones in two steps according to [20]. The first step is performed to merge all vertices whose Euclidean distance is smaller than or equal to the pre-defined distance threshold ϵ_d . The second step then refines the polygons by removing vertices approximately collinear to two other vertices. A vertex is considered collinear to two other vertices if the angle between them is greater than or equal to the threshold ϵ_a . The distance and angle thresholds ϵ_d and ϵ_a have been found empirically and set to $\epsilon_d = 4$ and $\epsilon_a = \cos(14^\circ)$. Refining the polygons is described in algorithm 2.

Algorithm 2 Polygon Refining

Require: Set of polygons P , thresholds ϵ_d, ϵ_a
Construct the initial KD-tree of vertices V from all P and create an empty set V_c of vertices to be merged
for All $v_i \in V$ **do**
 To a set V_c add v_j if $v_i \in P_i, v_j \in P_j$ and $P_i \neq P_j$ and $\|v_i - v_j\|_2^2 \leq \epsilon_d$
end for
while Any $v_j \in V_c$ **do**
 Replace $v_i \in V_j$ and $v_j \in V_c$ with $\frac{v_i + v_j}{2}$ in the respective polygons P_i and P_j
 Construct a new KD-tree of vertices V from all P
 for All $v_i \in V$ **do**
 Compute the set $V_c, v_i \in P_i, v_j \in P_j$ and $P_i \neq P_j$ and $\|v_i - v_j\|_2^2 \leq \epsilon_d$
 end for
end while
while Any $|\cos(\overrightarrow{v_i v_j}, \overrightarrow{v_j v_k})| \geq \epsilon_a$ for $v_i, v_j, v_k \in P_i$ **do**
 Remove the middle vertex v_j
end while

4 Experimental Description

In our experiments, we first compare the proposed segmentation models CAB1 and CAB2 of our pipeline with the DFPR [23] and Cubicasa5k [18] SOTA methods on the Cubicasa dataset. Further, the CAB1 and CAB2 models are evaluated on the Cubicasa, R3D, CVC-FP, and MLSTRUCT-FP datasets. We decided to employ the DFPR and Cubicasa5k methods, as they are well described and, with the implementations, publicly available. The Cubicasa, R3D, and CVC-FP datasets are ones with publicly available data, have different classes, and are evaluated along a multiple methods. The MLSTRUCT-FP dataset was evaluated because of the high number of images, even though it has annotated only the wall class.

4.1 Model Modifications

To ensure a fair comparison between the evaluated models, we modified the DFPR and Cubicasa5k by removing the fixed input size requirements and room predictions. The room predictions were removed because they would otherwise only reduce the accuracy of the boundary and opening classes. Concerning the Cubicasa5k model, the balanced entropy loss function from the DFPR model was adopted, as the original categorical cross-entropy loss failed to converge. The Cubicasa5k model is an improved version of [19]; this model is therefore indirectly evaluated as well. The edited models are noted as the DFPR_β and the CubiCasa5k_β .

A Github repository [48] was employed to implement the DFPR_β models. We modified the model to be able to handle images with different input sizes. The CubiCasa5k_β was adopted from repository [49]. Again, we modified the model to be able to handle different image sizes. Unlike the DFPR_β and the CubiCasa5k_β with the VGG16 [26] backbone, our new models CAB1, and CAB2 use the EfficientNetB2 [50].

4.2 Dataset Description

As mentioned above, several datasets associated with floor plan analysis are available. Below, Table 1 compiles some of the existing datasets that were deemed suitable for the discussed task. The suitability is based on the nature of the floor plan data plus the format and the extent of the data annotation.

Dataset name	Samples	W	G	R	D	L	O	S	Data origin	Note
†CVC-FP [51]	122	✓	X	X	✓	X	✓	X	Europe region	
†Rent3D [27]	215	✓	X	X	X	X	X	X	Central London	Include unit photos 2-30 photos / unit
R-FP [52]	500	✓	X	X	X	X	X	X		Highly variable data
*R2V [19]	100k+	✓	X	X	✓	X	X	X	Japan region	Original data [53]
†CubiCasa5K [18]	5000	✓	X	✓	✓	X	✓	✓	Finland region	
Versailles FP [54]	500	✓	X	X	X	X	X	X	Versailles Palace	
Floor Plan CAD [6]	15663	✓	X	X	✓	✓	✓	✓		
†MLSTRUCT-FP [55]	954	✓	X	X	X	X	X	X	Chilean region	Multi-unit floor plan dataset

Table 1: The datasets potentially suitable for the discussed framework; (†) evaluated in this research, * NOT publically accessible. The abbreviations stand for: W (Wall), G (Glass wall), R (Railing), D (Door), L (sLiding door), O (windOw) and S (Stairs).

The proposed annotations listed in Table 1 are based on the annotated classes outlined in Figure 14. If a class is not annotated as a separate one but is included in another class (for instance, a *sliding door* is annotated as a *door*), it is considered not annotated. The seven selected classes (*wall*, *glass wall*, *railing*, *door*, *sliding door*, *window* and *stairs*) follow from the general requirements of the rescue services on the model representation. Most of the datasets do not annotate all of these classes, as they were designed for substantially different applications.

The datasets employed in evaluating this study (the Rent3D or R3D and the CubiCasa) were not used in the originally published form. The R3D [27] was utilized in its extended variant as proposed by Zeng et al. [23]. Instead of the CubiCasa5K [18], we applied a refined version, named simply the CubiCasa; this contained fewer floor plans compared to the original, more emphasis being placed on consistency and label accuracy. Additional information may be found in Table 2.

Both of the datasets were augmented by combining flips, rotations, crops, and scaling. The data splits were set to 60, 20, and 20% (train, validation, test). The segmentation model was also evaluated on the MURF, our multi-unit floor plan dataset, based on images of large buildings. Due to their sizes and noise levels, the floor plans in the MURF are of a high complexity compared to the CubiCasa. The MURF considers all seven semantic classes: walls, glass walls, railings, doors, sliding doors, windows, and stairs. Regrettably, this dataset could not be published herein due to regulatory compliance. Still, we decided to present the results, as the MURF dataset is highly complex and valuable. The results are exposed in Figure 14.

Dataset name	Samples	Resolution [px]	Augmented samples
CVC-FP	122	w: 905-7383; h: 1027-5671	6966
Rent3D (extended) [23]	225	w: 337-1104; h: 175-1024	957
CubiCasa	560	w: 254-5052; h: 206-6768	7265
MLSTRUCT-FP	954	w: 6360-9650; h: 6300-9500	42653

Table 2: Evaluated dataset details, numbers of samples before and after augmentation and range of sample resolutions

Description	Color	Value	Boundary	Opening
Wall		1	✓	✗
Glass wall		2	✓	✗
Railing		3	✓	✗
Door		4	✗	✓
Sliding door		5	✗	✓
Window		6	✗	✓
Stairs		7	✗	✗

Table 3: Structural element classes of MURF with colors, boundary and opening descriptions.

4.2.1 Investigated semantic classes

The investigated semantic elements were selected upon a consultation with rescue services and are shown in Table 3. In a full scale, they are available only in the MURF dataset. The boundary and opening labels are used for the heatmap regression loss. The glass walls refer to walls that are breakable and transparent but can not be opened like windows; the railings refer to traversable half-height; and the sliding doors then refer to doors that open in a distinct manner without a swing path, including both doors that open automatically and those that do not. The reason for including glass walls and railings in addition to regular walls is related to their semantic difference and distinct use case.

The same reason applies to sliding doors, which are added in addition to regular doors. The features of glass walls, railings and sliding doors differ sufficiently for a model to be able to learn to detect them separately. Including glass walls, railings and sliding doors in generated 3D models also improves the overall accuracy, since the 3D representations are now ‘closer’ to the real-world state of buildings. This in turn improves the applicability of generated 3D models. Railings could for instance help the fire brigade identify the best route for a building by taking the traversable walls into account.

4.2.2 R3D

In our paper, we used the R3D dataset as proposed in [27], with seven removed floorplans with incorrect padding values. Three examples of floor plans from R3D dataset are shown in Figure 10.

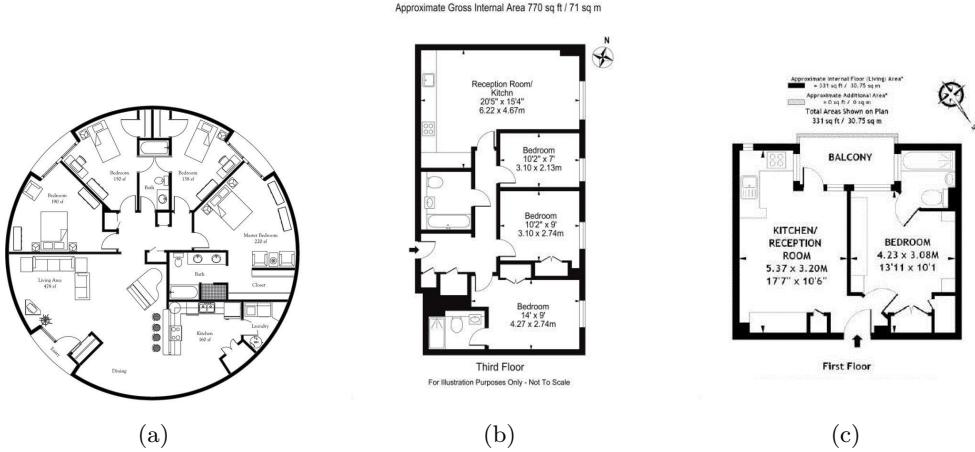


Fig. 10: Three examples of floor plans in the R3D dataset [27].

4.2.3 CubiCasa

The overall annotation quality of the CubiCasa5K dataset, proposed in [18], is significantly lower than that of R3D. The class labels are often incorrect, incomplete, or contain multiple unrelated artifacts. In several of the floor plans, the annotations do not match, due to differences in the current real-world condition of the apartments. These divergences, however, would only confuse a model that is learning to identify its elements. Figure 11 presents examples of common annotation problems. The quality of the railing annotations in the floor plans is especially poor. Although not mentioned in the original work [18], stairs have been scarcely labeled in the floor plans.

To improve the annotation quality, we performed manual selection of 560 floor plans of varying sizes and complexities to remove duplicates and find annotations that are relatively cheap to correct. In each floor plan, all labels were manually corrected. Stairs annotations were added to those floor plans which did not include them. This paper proposes CubiCasa as a refined version of CubiCasa5K, the tool to be used in the remaining portion of this work. A labeled subset of CubiCasa5K can provide sufficient information for effective learning and evaluation, as shown in [56].

To generate PNG mask representations of the original annotations in SVG format, a SVG parser was written. The annotations in CubiCasa5K contain many classes and nested relations, of which only a few are evaluated. In order to extract the semantic classes shown in Table 3, the following objects were identified in the SVGs: Stairs; Railing; Wall; Window; Door and Column. The majority of the objects was accompanied by a set of points, which was used to determine the location of the object in the floor plan. When a column was represented by a circular SVG object, the corresponding center point and radius was used. Column objects were merged with wall objects, as they are semantically identical to each other. The parsed SVG files were further processed to generate the PNG masks.

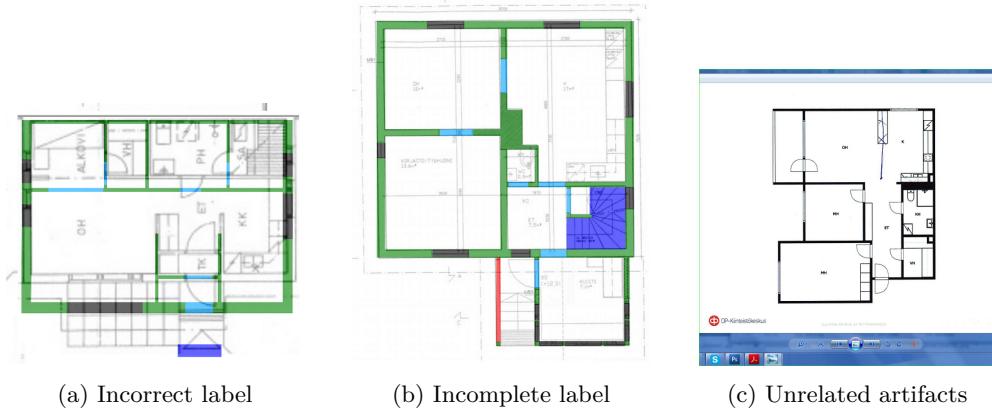


Fig. 11: Three examples of annotation inconsistencies as incorrect or incomplete (stairs) labels and unrelated artifacts caused by cropping (e.g. window related) in the CubiCasa5K dataset.

Transforming the parsed SVG files into PNG representations was not trivial due to overlapping class labels. For instance, windows are contained in walls in the annotations, which could result in small wall pixels along the boundaries of windows in generated PNG masks. Another example are railings, which are sometimes contained within the Stairs class annotations. These overlapping labels had to be corrected because they do not represent the ground-truth accurately and can inhibit the learning process. Two measures were taken to achieve this: (1) a small dilation operation with a 3×3 kernel was applied to all windows; and (2) the annotations were then combined into a single mask, relying on a predefined ‘drawing’ order to fix part of the overlap. The drawing order used to convert the SVG files into PNG representations was set as follows: Wall: 1; Stairs: 2; Railing: 3; Door: 4; Window: 5. Three examples of CubiCasa floor plans are shown in Figure 12.

4.2.4 CVC-FP

The collection of 122 scanned floor plan documents divided in 4 different subsets regarding their origin and style. For our purposes, the classes examined are Wall, Door, Window and Parking. The Parking class has been merged with the Door class. Three examples of CVC-FP floor plans are shown in Figure A1.

4.2.5 MLSTRUCT-FP

The Machine Learning Structural Floor Plan (MLSTRUCT-FP) dataset comprises of 954 high resolution multi-unit rasterized plan images [55]. Authors published their library for annotation visualization and work with the dataset. Three examples of MLSTRUCT-FP floor plans are shown in Figure A2.

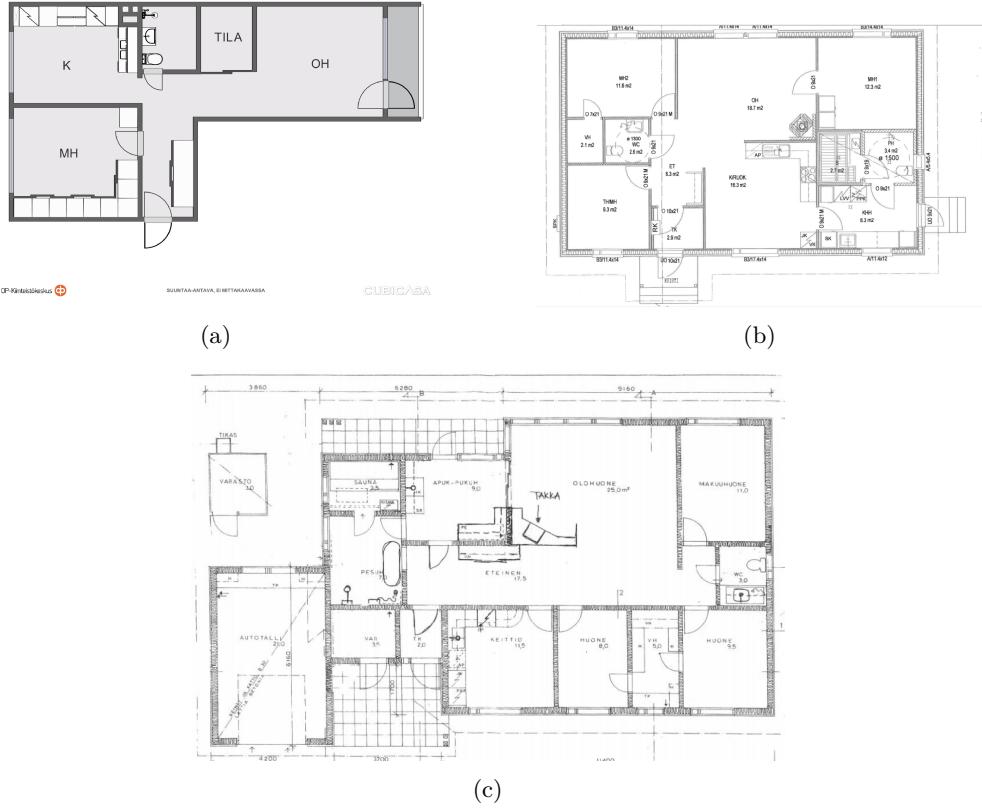


Fig. 12: Three examples of floor plans in the CubiCasa dataset [18].

4.2.6 MURF

The MURF dataset is a novel multi-unit floor plan dataset comprising large, high-quality floor plans of varying complexity. Multi-unit floor plans differ from single-unit ones, as they contain the architectural information of the entire floor of a building, compared to a single apartment. As a result, multi-unit floor plans are larger and more complex than their single-unit counterparts. MURF consists of eight floor plans of a large municipal and apartment buildings. Regrettably, this dataset could not be published herein due to regulatory compliance, but it is still described for a better understanding of our paper and its high complexity.

MURF consists of rasterized versions of the source floor plans created using Adobe Photoshop. Since the source floor plans were always provided in a PDF file format, the rasterized images could be simply be generated by converting the PDFs into a PNG image. The resolution of a rasterized PNG floor plan was set large enough so that all details of the PDF were captured with sufficient quality. Figure A3 contains 3 examples of multi-unit floor plans in MURF and Figure A4 shows theirs ground truth labels.

4.2.7 Derived dataset for the ablation study

For ablation study we create our own dataset as a combination of CubiCasa and MURF. Although the list of semantic classes of CubiCasa is smaller, it was not merged with the class list of MURF, which allows for a full evaluation on all semantic elements. The purpose of this combined dataset is to evaluate to what degree a model can generalize across different drawing notations and floor plan sizes.

4.2.8 Comparison of Datasets

R3D, CubiCasa, CVC-FP, MLStruct-FP and MURF have all been used because of their varying complexity, floor plan size, data source. Although the number of floor plans in MURF is small, the significantly larger floor plans result in a comparable amount of data. Figure 13 shows the number of pixels per semantic class of each dataset. For comparison purposes, distinct wall and door classes of MURF have been merged. The important observation to make here is that the number of floor plans does not solely dictate the amount of data in a dataset. The size and content of the floor plans are equally important. MURF contains eight high resolution floor plans, resulting in a comparable number of pixels compared to CubiCasa which is 70 times larger. The MLStruct-FP dataset contains significantly more data, however only the wall class makes it comparable to the other sets.

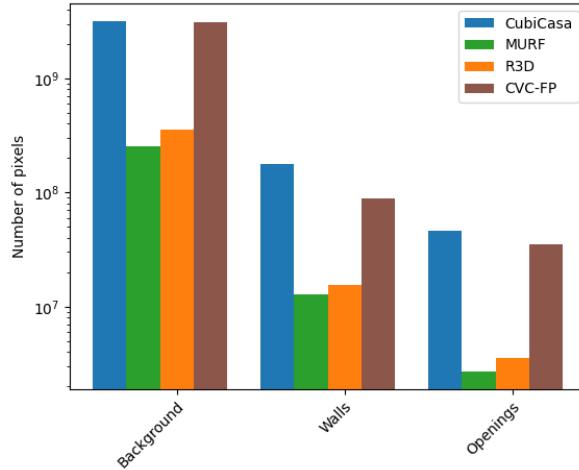


Fig. 13: Number of pixels per semantic class in MURF compared to R3D, CubiCasa and CVC-FP.

Some of the larger datasets exposed in Table 1 were not considered in the evaluation for specific technical reasons. This fact applies to the R2V dataset, which is difficult to obtain, and the Floor Plan CAD dataset, which provides the floor plans in a tiled form, with no apparent way of assembling the tiles into more complex structures that are desired for the evaluation.

4.3 Evaluation Metrics

To evaluate the experiments, we decided to use the recall, F1 score [57], and intersection over union (IoU) [58] metrics. The F1 score is defined using the basic confusion matrix indicators, the correct classification TP (true positive) and TN (true negative), and the incorrect classification FP (false positive) and FN (false negative). Whereas precision penalizes the FP results and recall does so in the FN cases, the F1 score elegantly combines both, balancing them. The F1 score is high only if both the precision and the recall are also high. The IoU is a simple metric, often used where the object shape and localization are sought (segmentation, object detection, and other aspects). Thus, we are interested in the ratio between the overlapping area of the two objects (the intersection) and the area of their union. In this study, the objects that are being analyzed are the patches of a bitmap; the intersections are then the shared pixels, and the union would be the totality of all the pixels in either object.

4.4 Experimental Setup

All of the investigated models were trained for a maximum of 200 epochs with a batch size of one, two or ten due to the GPU memory limitations and the large scale of the training images. Further, a bigger batch size would not offer any benefits in terms of the model accuracy, as the group normalization performs well on small batches [59]. For the proposed architectures CAB1 and CAB2, the ADAM optimizer was used, the learning rate being 1×10^{-4} .

Our framework implements configuration file generation presented in [60]. The configuration file consists of all training parameters including for example: model initialization weights, number of filters used to create the model, or model's backbone.

The learning rate was reduced by a half after 10 epochs until we reached a minimum of 1×10^{-5} if the validation loss had been not reduced. An early stopping criterion with a patience of 30 epochs was employed as well. All of the models were pre-trained on the ImageNet dataset [61]. The investigated implementations are based on the TensorFlow and were trained on a machine with $2 \times$ NVIDIA RTX A5000 GPU with a 24 GB VRAM, $2 \times$ dual Intel Xeon Silver 4208 CPU @ 2.10GHz 8-core processors, and 1 TB SSD storage.

5 Experimental Results

Quantitative result analysis was performed using the aforementioned metrics as a statistic on the pixel level. Qualitative results in terms of the individual floor plan segmentations and reconstructions were analysed to determine potential limitations. An example of the overall results of our proposed pipeline including the segmentation, post-processing, and 3D reconstruction on the MURF dataset is shown in Figure 14.

5.1 Qualitative Evaluation of the Experimental Results

This section provides a more extensive qualitative evaluation considering a comparison of segmentation masks and generated 3D models. The two baseline models DFPR_B and CubiCasa5K_B are compared against the CAB2 proposed architecture.

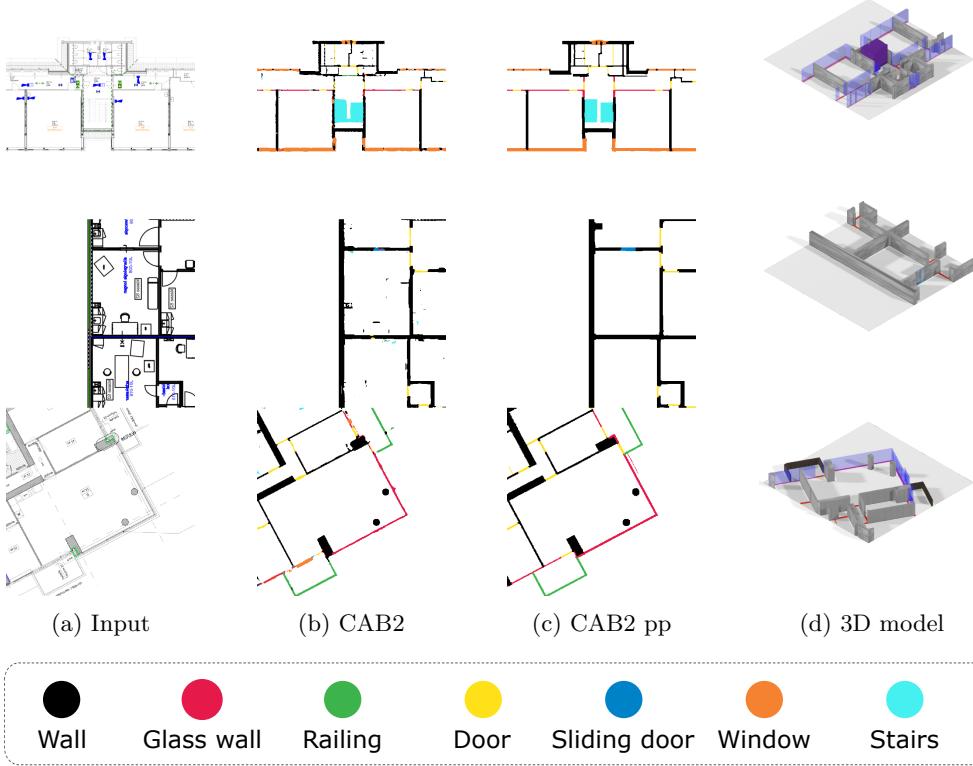


Fig. 14: Example of several floor plans from the MURF dataset proceed by the proposed framework using the CAB2 segmentation technique. (a) Input samples, (b) Outputs of the recognition module, (c) Outputs of the post-processing module, (d) 3D reconstructed models.

Figure B5, Figure B6, Figure B7 and Figure B8 contain three example floor plans and generated masks for each dataset and each model. The results show that the proposed CAB2 is more robust to the noise introduced by augmenting the floor plans in the dataset compared to the two baseline models. The difference in accuracy between the models increases for the more complex datasets such as the MURF.

Figure B9, Figure B10, Figure B11 and Figure 14 compares the generated 3D models for three floor plans for the proposed architecture for the three datasets. The post-processed segmentation mask, abbreviated by ‘pp’, is also shown. The results show that the reconstruction method is effective at reducing noise so an accurate 3D model can be generated. Figure C13 contains an example of a full-sized 3D model generated from MURF by the proposed CAB2. The results from MURF show that the features captured by the proposed architecture are sufficient to infer the semantic information in a full-sized multi-unit floor plan.

5.2 Comparing the Proposed Framework with the SOTA Methods

Importantly, Table 4 compares the performance of our pipeline carrying the CAB1 and CAB2 segmentation cores with the DFPR $_{\beta}$ and the Cubicasa5k $_{\beta}$. The proposed models performed almost the same, as the observed metrics do not differ at all. The two other models showed worse results over all of the observed classes and in all of the investigated metrics. The proposed models reached an average F1 score of 0.86 and an average IoU of 0.76 compared to the F1 score of 0.74 and IoU of 0.61 in the DFPR $_{\beta}$ and the F1 score of 0.74 and IoU of 0.60 in the Cubicasa5k $_{\beta}$.

	CAB1			CAB2			DFPR $_{\beta}$			CubiCasa5k $_{\beta}$		
	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU
Walls	0.90	0.90	0.83	0.90	0.90	0.83	0.75	0.83	0.72	0.80	0.86	0.75
Railings	0.74	0.77	0.63	0.76	0.77	0.63	0.42	0.53	0.36	0.42	0.55	0.38
Doors	0.82	0.82	0.70	0.82	0.82	0.70	0.57	0.69	0.53	0.57	0.68	0.52
Windows	0.91	0.90	0.82	0.92	0.90	0.82	0.72	0.81	0.68	0.75	0.81	0.69
Stairs	0.87	0.89	0.81	0.86	0.89	0.81	0.79	0.85	0.74	0.70	0.79	0.67
Mean	0.85	0.86	0.76	0.85	0.86	0.76	0.65	0.74	0.61	0.65	0.74	0.60

Table 4: The pixel-wise recall, F1-score, and IoU reached with various methods over the Cubicasa dataset.

Detailed results of comparison CAB1 and CAB2 segmentation cores with the DFPR $_{\beta}$ and the Cubicasa5k $_{\beta}$ different datasets: CVC-FP Table 5 and R3D Table 6.

For a more straightforward comparison with [18] and the other researches using the entire unmodified CubiCasa5k dataset, we include these results in Table 7. Similarly for a direct comparison on the R3D dataset [23] we added the Table 8.

5.3 Evaluating the Proposed Framework on Various Datasets

The data in Table 9 represent the results achieved in the proposed pipeline carrying the CAB1 and CAB2 segmentation cores over the Cubicasa, R3D, CVC-FP and MLStruct-FP datasets. These results show a comparable and consistent performance of the proposed segmentation cores over various datasets with slightly better results achieved with the CAB2 over the CVC-FP dataset.

5.4 Computational Resources

Table 10 shows training and evaluation times of the selected compared methods on the Cubicasa dataset (7265 images in the complete set, and 1453 images in the evaluation set) as measured on the hardware described in Section 4.4. It also shows the model size as saved by the Keras checkpoint and checkpoint variables. From the shown data it is apparent the the proposed CAB1 and CAB2 models are more computationally expensive to train and evaluate, but take less memory to save.

	CAB1			CAB2			DFPR _{β}			CubiCasa5k _{β}		
	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU
Walls	0.95	0.94	0.88	0.96	0.95	0.91	0.84	0.90	0.83	0.86	0.91	0.84
Doors	0.74	0.74	0.59	0.84	0.81	0.69	0.57	0.69	0.53	0.44	0.58	0.40
Windows	0.91	0.90	0.83	0.93	0.93	0.87	0.83	0.88	0.79	0.78	0.85	0.74
Mean	0.87	0.86	0.77	0.91	0.90	0.82	0.74	0.83	0.43	0.69	0.78	0.28

Table 5: The pixel-wise recall, F1-score, and IoU reached with various methods over the CVC-FP dataset.

	CAB1			CAB2			DFPR _{β}			CubiCasa5k _{β}		
	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU
Walls	0.94	0.94	0.90	0.94	0.94	0.90	0.88	0.90	0.83	0.86	0.89	0.81
Openings	0.84	0.85	0.75	0.84	0.85	0.74	0.70	0.77	0.62	0.65	0.73	0.58
Mean	0.89	0.90	0.82	0.89	0.90	0.82	0.79	0.84	0.73	0.75	0.81	0.70

Table 6: The pixel-wise recall, F1-score, and IoU reached with various methods over the R3D dataset. Note that the openings are the union of the door and window classes.

	CAB1		CAB2		CubiCasa5k [18]	
	Acc.	IoU	Acc.	IoU	Acc.	IoU
Walls	0.97	0.73	0.97	0.73	0.85	0.73
Railings	0.99	0.41	0.99	0.44	0.28	0.23
Doors	0.99	0.64	0.99	0.66	0.59	0.53
Windows	0.99	0.75	0.99	0.76	0.73	0.66
Mean	0.98	0.63	0.98	0.64	0.61	0.53

Table 7: The pixel-wise accuracy, and IoU reached with various methods over the Cubicasa5k dataset.

	CAB1		CAB2		DFPR [23]		MIP [25]	
	Acc.	IoU	Acc.	IoU	Acc.	IoU	Acc.	IoU
Walls	0.99	0.90	0.99	0.90	0.98	—	0.90	—
Openings	0.99	0.75	0.99	0.74	0.83	—	0.94	—
Mean	0.99	0.82	0.99	0.82	0.86	—	0.92	—

Table 8: The pixel-wise accuracy, and IoU reached with various methods over the R3D dataset. Note that the openings are the union of the door and window classes.

5.5 Ablation Study

Each ablation study has been performed on the combined dataset, as this has been shown to be the most difficult dataset to learn from. This allows for more room to improve a baseline model and thus better examination of the effect of an added component.

		Cubicasa			R3D			CVC-FP			MLSTRUCT-FP		
		Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU
CAB1	Walls	0.90	0.90	0.83	0.94	0.94	0.90	0.95	0.94	0.88	0.92	0.89	0.81
CAB2	Walls	0.90	0.90	0.83	0.94	0.94	0.90	0.96	0.95	0.91	0.92	0.90	0.83

Table 9: Results of the proposed pipeline with CAB1 and CAB2 segmentation models reached over Cubicasa, R3D, CVC-FP, and MLSTRUCT-FP datasets. As the MLSTRUCT-FP and R3D datasets do not contain all classes evaluated in Table 3, only walls results are compared.

	CAB1	CAB2	DFPR $_{\beta}$	CubiCasa5k $_{\beta}$
Training time [s]	181912	134686	21640	22969
Evaluation time [s]	499	403	95	183
Model size [MB]	80	101	168	355
Model variable size [MB]	59	49	150	315

Table 10: Model training and evaluation times on the Cubicasa dataset, and model size in terms of keras checkpoint, and keras checkpoint variables sizes

	AC	CAM	SAM	CAM+SAM
Pixel Acc.	0.85	0.85	0.85	0.85
Mean Class Acc.	0.66	0.66	0.66	0.70
Mean Class IoU.	0.54	0.55	0.54	0.54

Table 11: Ablation study of CAM and SAM blocks used in the attention mechanism on the combined dataset. The baseline model is a variant of model CAB2 that uses a single *AC* block.

5.5.1 Attention Mechanism

The first ablation study is on the CAM and SAM blocks used in the attention mechanism in the proposed architectures. To investigate the improved accuracy obtained by using the CAM and SAM blocks, each is added separately to a baseline architecture. The baseline architecture is a variant of model CAB2 that performs a single *AC* convolution to compute the decoder blocks, similar to the square convolution used in U-Net3+. Table 11 contains the results of the ablation study on the attention mechanism.

The results show that using both a CAM and SAM module achieves the best performance. Using only the CAM module increases the mean IoU. The SAM module is likely responsible for the improved pixel accuracy and mean class accuracy.

5.5.2 Multi-Task Loss

The final multi-task loss consists of three losses: asymmetric unified focal loss \mathcal{L}_{aUF} , adaptive affinity field loss \mathcal{L}_{AAF} , and multi-scale heatmap regression loss \mathcal{L}_{MHR} . In

	\mathcal{L}_{aUF}	$\mathcal{L}_{\text{aUF}} + \mathcal{L}_{\text{AAF}}$	$\mathcal{L}_{\text{aUF}} + \mathcal{L}_{\text{AAF}} + \mathcal{L}_{\text{MHR}}$
Pixel Acc.	0.85	0.85	0.85
Mean Class Acc.	0.66	0.68	0.70
Mean Class IoU.	0.53	0.54	0.54

Table 12: Ablation study of combining multi-task loss terms of CAB2 on the combined dataset. Terms \mathcal{L}_{aUF} , \mathcal{L}_{AAF} and \mathcal{L}_{MHR} refer to the asymmetric unified focal loss, adaptive affinity field loss, and multi-scale heatmap regression loss respectively.

order to determine their impact on accuracy, the losses are iteratively combined and evaluated on the combined dataset with CAB2. Table 12 shows the results.

The results show that the combined loss function achieves the best performance. The affinity field loss increases the mean class accuracy. The added heatmap regression loss further increases class accuracy and the mean IoU.

6 Conclusions

This paper was conceived to seek a method for generating a digital twin (in the form of a 3D model) to a real-world building from less-than-ideal floor plans. Having a reliable 3D representation of a building rather than collections of 2D plans in various forms and quality is an important step in enabling more effective work with these resources. Evacuation route planning embodies our main motivation factor, due to the cooperation with firefighters. However, the proposed method is applicable to many other urban planning problems involving traffic issues and various simulations of environmental situations. The security risks definitely form a major difficulty, but as the floor plans are already publicly available to the civil authorities, the restrictions already exist. In the future, developing an effective storing system will certainly be desirable. Development of this and similar methods is an important step towards safer and more efficient emergency services in the ever-growing urban environment.

We developed a pipeline that not only analyzes an input drawing at the pixel level but also segments and classifies selected features to generate a 3D reconstruction of the relevant building. Two new multi-scale, fully trainable pixel segmentation models designed for multi-unit floor plan analysis, the CAB1 and CAB2, are proposed as well. The pipeline outperformed the SOTA DFPR and Cubicasa5k methods, and it was evaluated on three datasets, reaching consistently high evaluation metrics over various data. By design, the recognition part lacks a region of interest (ROI) detection module, which otherwise allows cropping the floor plan; such structuring generally improves the accuracy. The main reason for not including the module rests in that object detection models, such as those in the YOLO family, are sufficient enough to crop floor plans to their appropriate size, as shown in, for example, [20], where the authors employ a variant of the YOLOv4 for their ROI module.

A comparison of the pipeline with two SOTA floor plan processing models, the DFPR and CubiCasa5K, showed that our approach surpasses these techniques on the CubiCasa dataset. We assume that the increased complexity of our pipeline allows

and will enable us to capture more effectively a larger range of features necessary to identify the elements in floor plans of different sizes.

More recent SOTA floor plan processing frameworks presented in [20], [24], and [25] unfortunately do not have their source codes publicly available, so the comparison in the same manner as with DFPR and CubiCasa5k was not possible. Nevertheless, the authors of [25] tested their framework on the R3D dataset, where they reached a recall of 0.89 in the case of walls. Our approach performed better in this case with a resulting recall of 0.94 in the case of wall recognition. Results for other methods and experiments were not comparable to those presented in this paper (different datasets, different metrics).

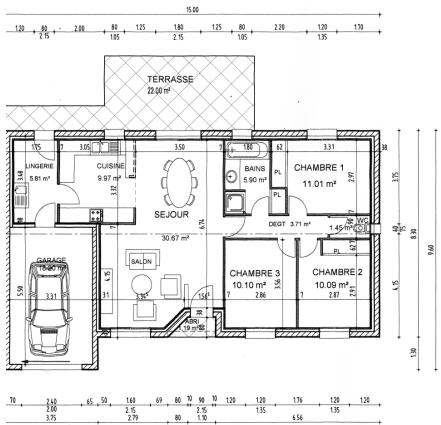
A limitation of the proposed reconstructing method lies in the visualization of stairs. Currently, stairs are visualized via polygons of an equal height, set to a higher value than the other semantic classes. Such detection, however, does not consider the orientation of the stairs, which would further improve the generated 3D models. Another constraint rests in visualizing round-shaped objects. At present, the segmentation masks are transformed into polygons, the endpoints of each connected component functioning as the vertices. Disadvantageously, this approach generates minor stair-like artifacts due to the discrete values in the segmentation mask, and the problem is especially apparent in round-shaped objects.

Besides a further analysis and tackling of these challenges, our future work will also focus on the processing of multi-floor plans, this will require implementation of methods to align the individual storey floor plans, which will allow understanding the context of stairwells and lift shafts across the building. As the presently available datasets usually contain the floor plans of smaller units such as flats, we will further focus our future research on collecting a dataset consisting of large floor plans across various types of buildings, including, public buildings, and medical facilities. We also intend to improve our segmentation network to process older plans, which are available only on paper sheets drawn according to various technical standards. This improvement is assumed to markedly refine the development of the digital twin, as the older buildings can be included in such models.

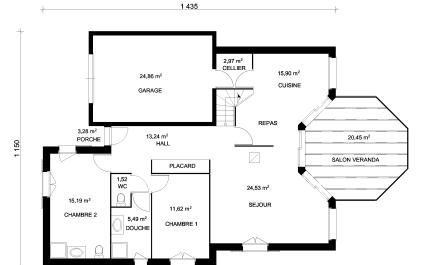
Acknowledgments

The completion of this paper was made possible by grant No. FEKT-S-23-8451 - "Research on advanced methods and technologies in cybernetics, robotics, artificial intelligence, automation and measurement" financially supported by the Internal science fund of Brno University of Technology.

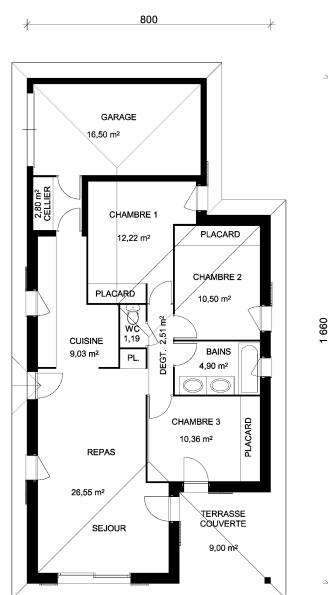
Appendix A Large scale examples of the used datasets



(a)

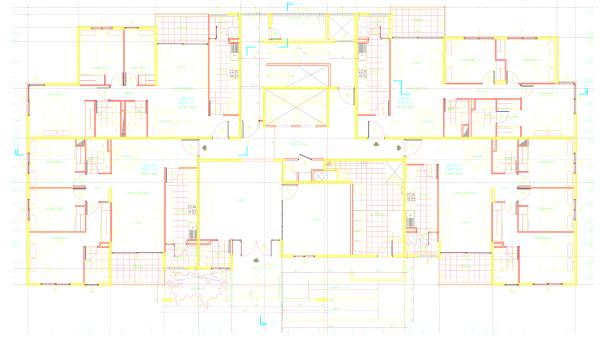


(b)

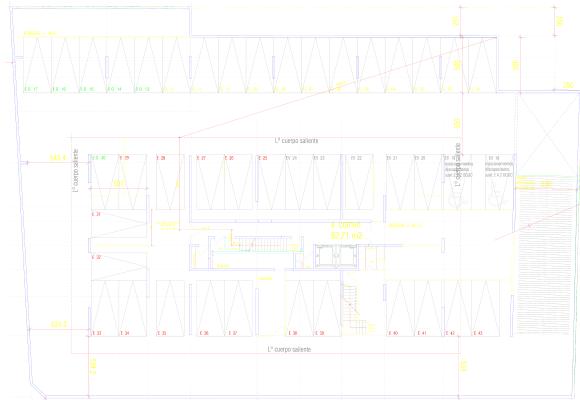


(c)

Fig. A1: Three examples of floor plans in the CVC-FP dataset.



(a)



(b)

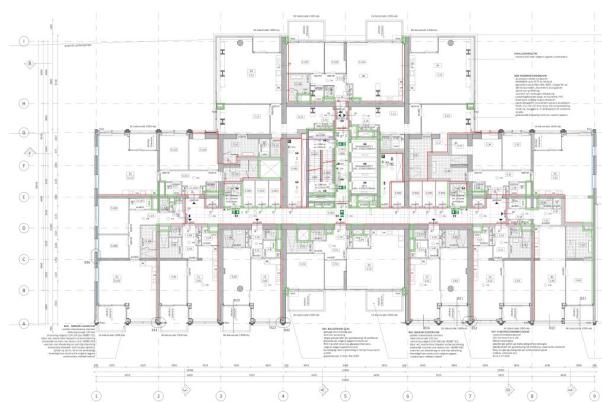


(c)

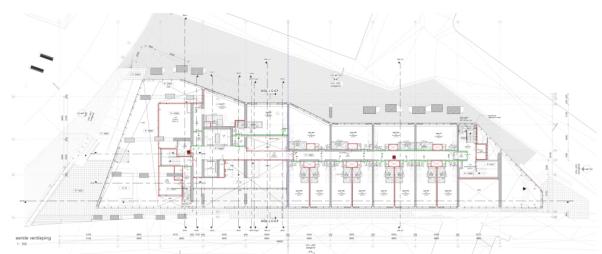
Fig. A2: Three examples of floor plans in the MLSTRUCT-FP dataset.



(a)

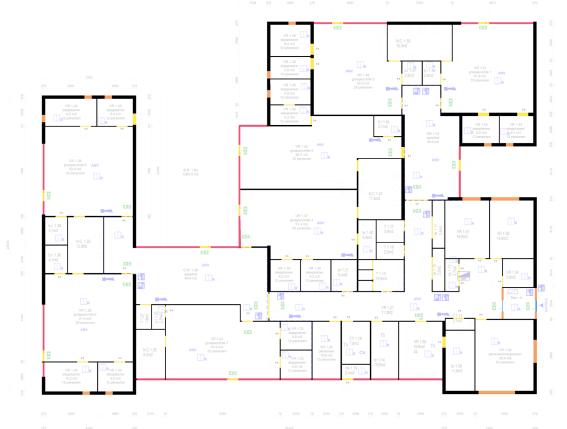


(b)



(c)

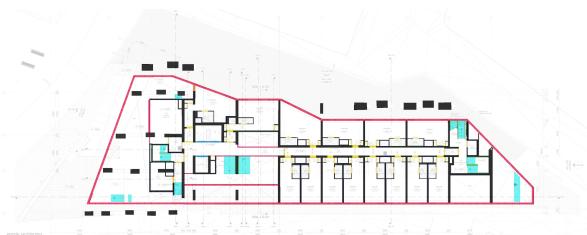
Fig. A3: Three examples of original multi-unit floor plans of varying complexity in the MURF dataset.



(a)



(b)



(c)

Fig. A4: Three examples of ground-truth labels superimposed on multi-unit floor plans in the MURF dataset.

Appendix B Qualitative comparison of generated 3D models

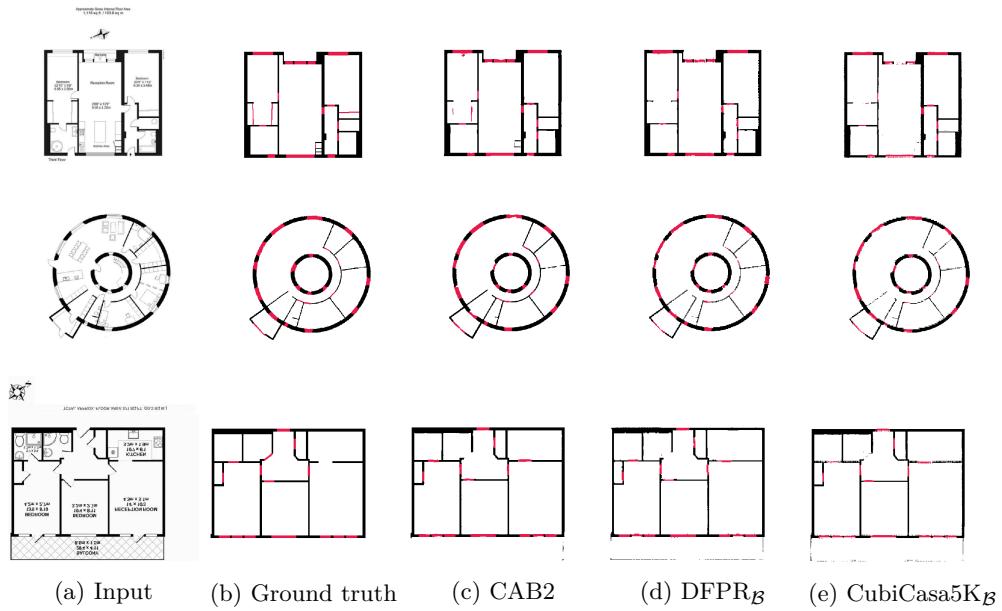
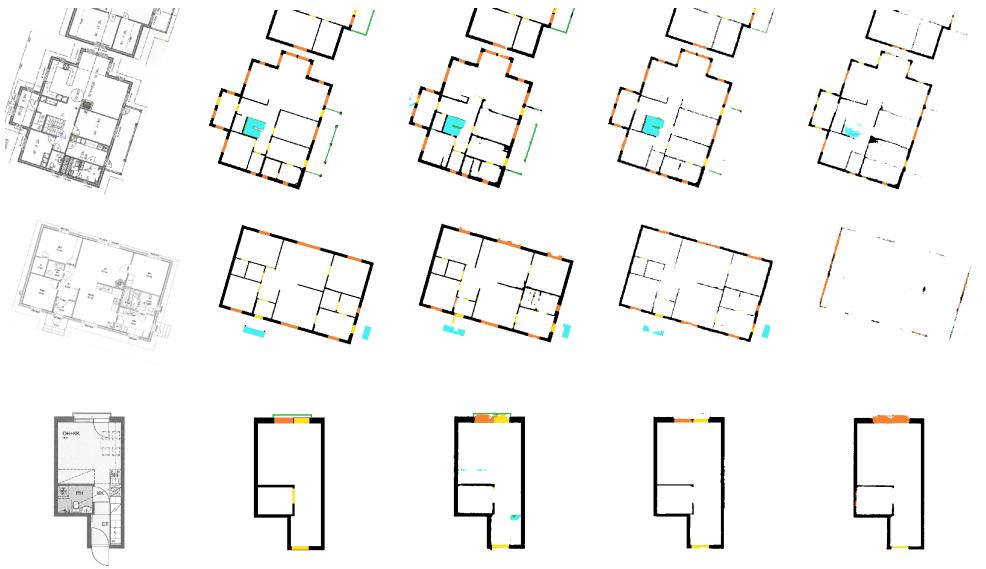
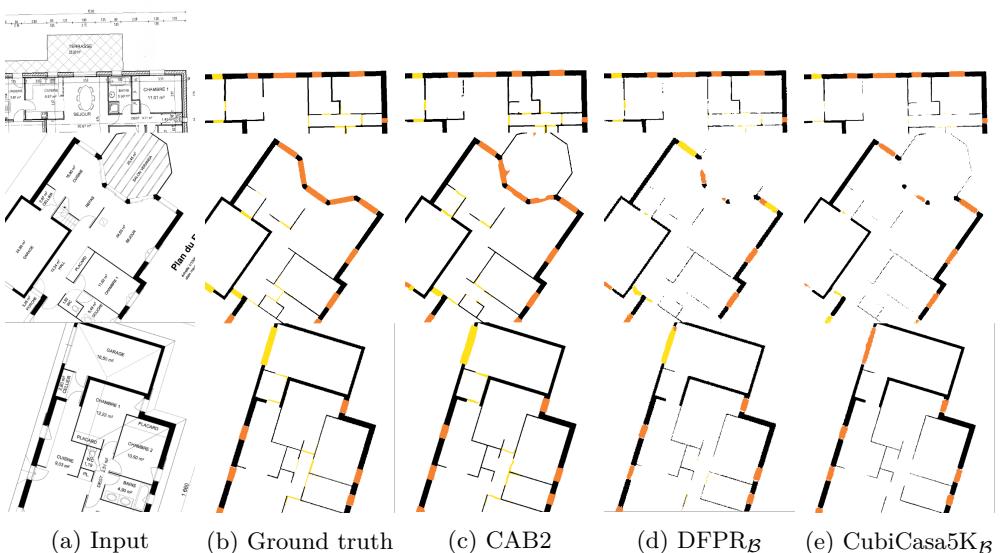


Fig. B5: Qualitative comparison of segmentation masks from R3D.



(a) Input (b) Ground truth (c) CAB2 (d) DFPR_B (e) CubiCasa5K_B

Fig. B6: Qualitative comparison of segmentation masks from CubiCasa.



(a) Input (b) Ground truth (c) CAB2 (d) DFPR_B (e) CubiCasa5K_B

Fig. B7: Qualitative comparison of segmentation masks from CVC-FP.

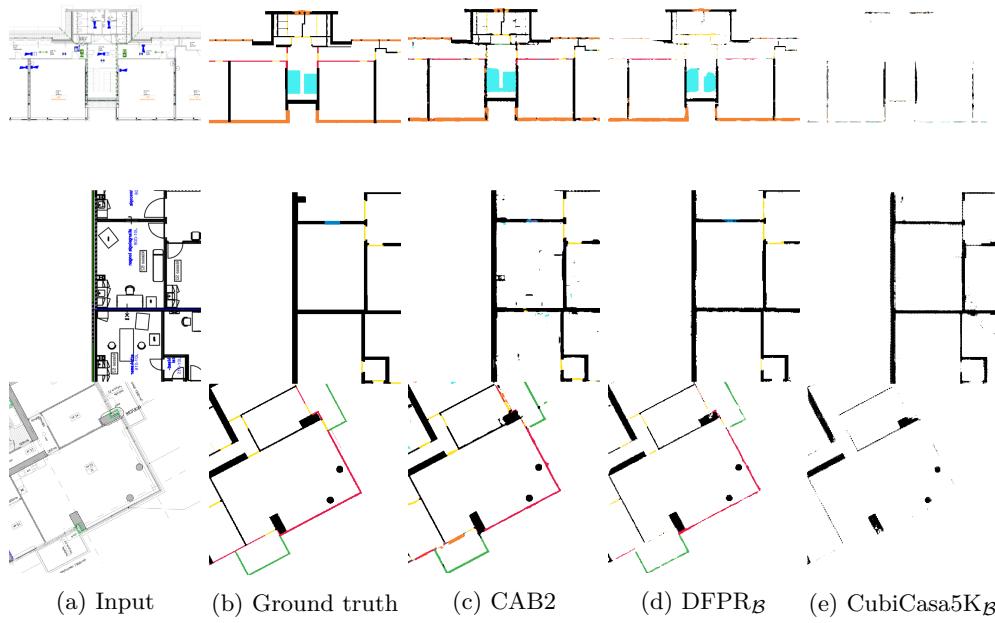


Fig. B8: Qualitative comparison of segmentation masks from MURF.

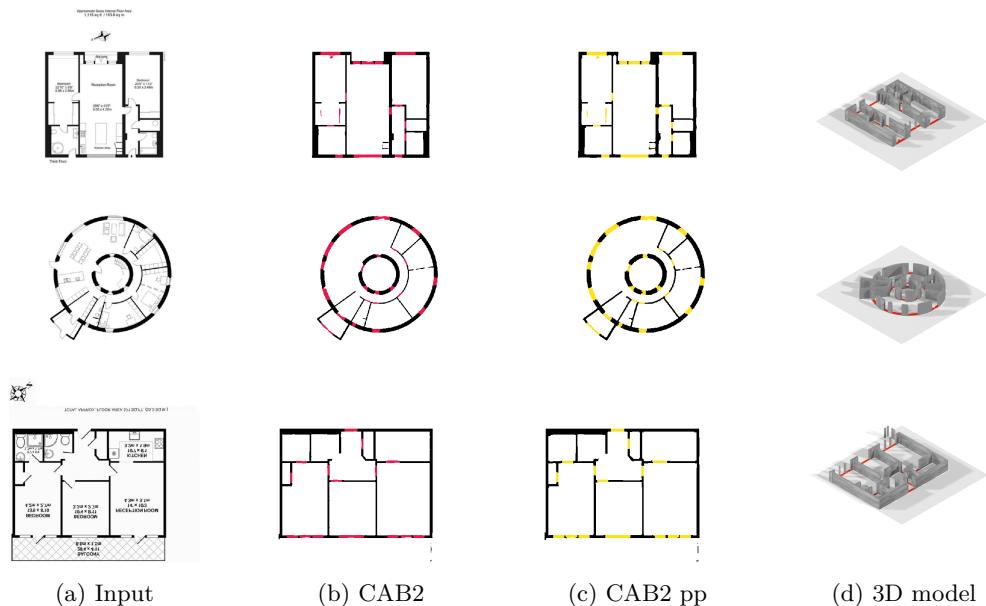


Fig. B9: Qualitative comparison of generated 3D models by proposed CAB2 from R3D.

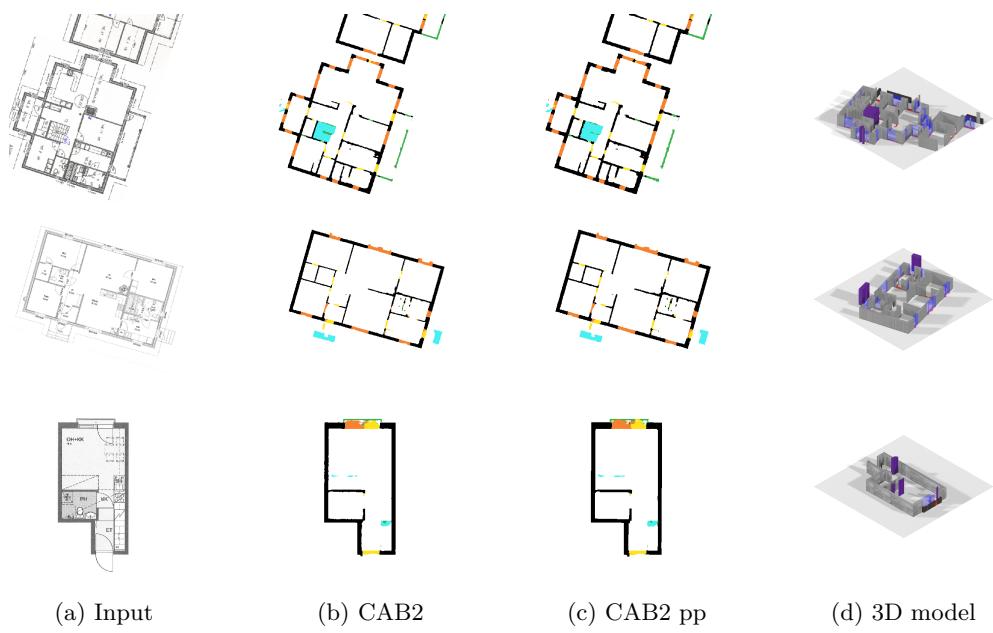


Fig. B10: Qualitative comparison of generated 3D models by proposed CAB2 from CubiCasa.

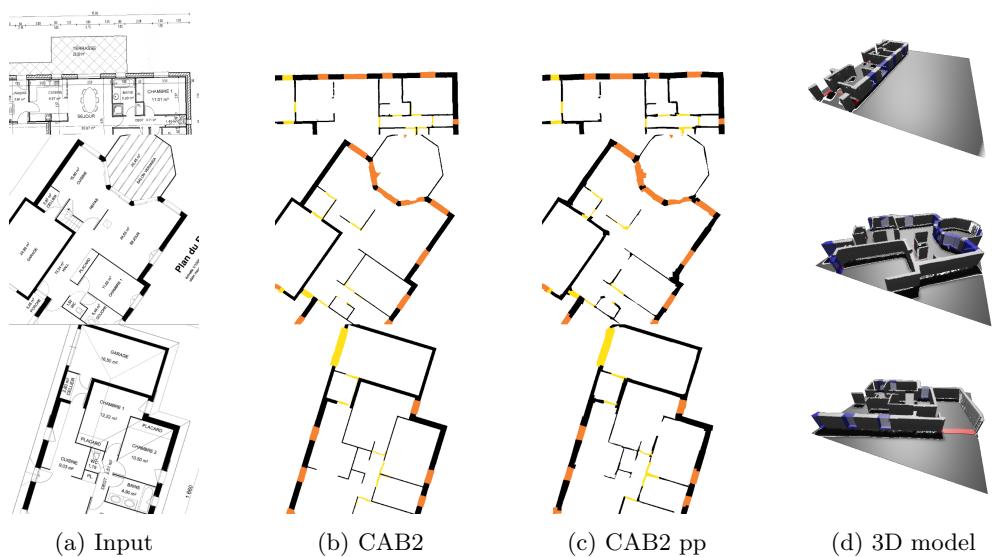


Fig. B11: Qualitative comparison of generated 3D models by proposed CAB2 from CVC-FP.

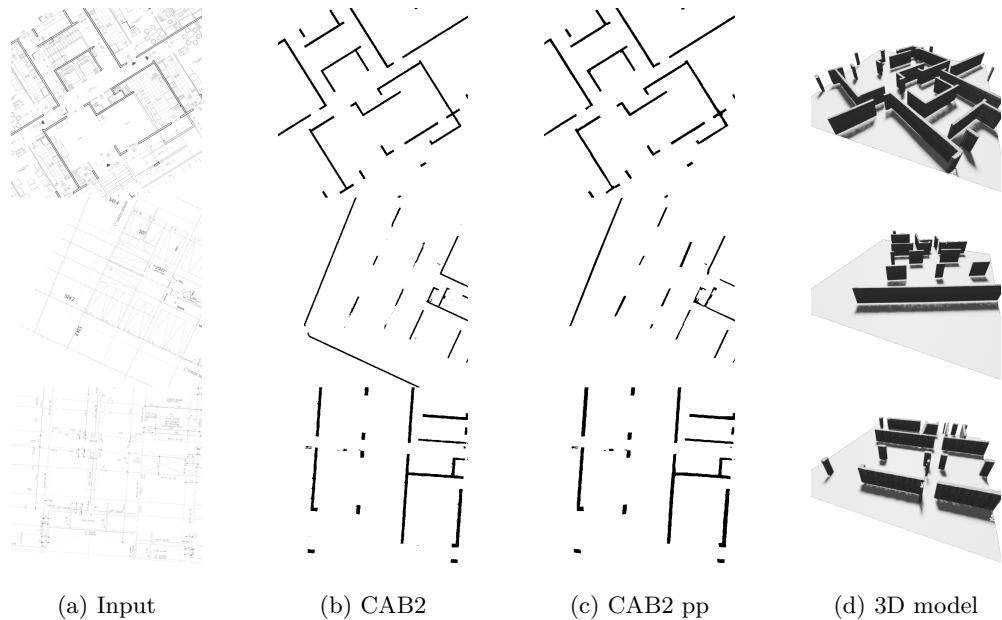
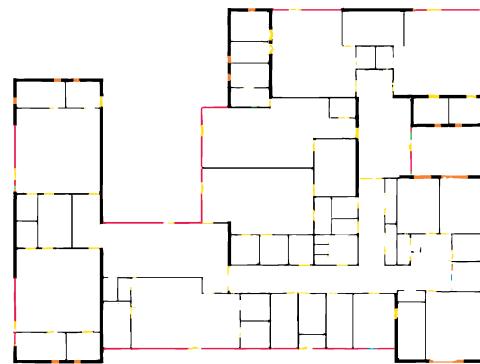


Fig. B12: Qualitative comparison of generated 3D models by proposed CAB2 from MLSTRUCT-FP.

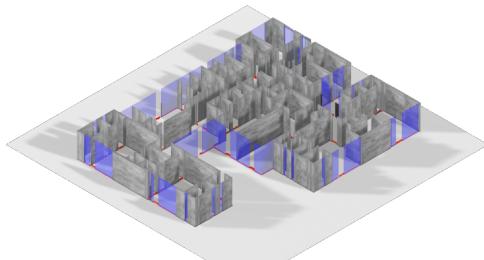
Appendix C Example of the 3D model from the full-sized MURF dataset



(a) Input floor plan



(b) Post-processed segmentation mask



(c) 3D model

Fig. C13: 3D model generated from full-sized MURF dataset.

References

- [1] Desa, U.: World urbanization prospects: the 2018 revision, key facts. New York: NY. Available online at: <https://population.un.org/wup/Publications/> (Accessed December 20, 2018) (2018)
- [2] LuVivi, Q., Kumar, P., WoodallPhilip, Don, R., HeatonJames: Developing a dynamic digital twin at a building level: using cambridge campus as case study. In: International Conference on Smart Infrastructure and Construction 2019 (ICSIC), pp. 67–75 (2019). <https://doi.org/10.1680/icsic.64669.067>
- [3] Khajavi, S.H., Motlagh, N.H., Jaribion, A., Werner, L.C., Holmström, J.: Digital twin: Vision, benefits, boundaries, and creation for buildings. IEEE Access **7**, 147406–147419 (2019) <https://doi.org/10.1109/ACCESS.2019.2946515>
- [4] Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B.: Characterising the digital twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology **29**, 36–52 (2020) <https://doi.org/10.1016/j.cirpj.2020.02.002>
- [5] Kratochvíla, L.: Point cloud obstacle detection with the map filtration. In: Proceedings I of the 29th Conference STUDENT EEICT 2023 General Papers. 1, pp. 455–459. Brno University of Technology, Faculty of Electrical Engineering and Communication, Brno (2023)
- [6] Fan, Z., Zhu, L., Li, H., Chen, X., Zhu, S., Tan, P.: Floorplancad: A large-scale cad drawing dataset for panoptic symbol spotting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10128–10137 (2021). <https://doi.org/10.1109/ICCV48922.2021.00997>
- [7] Gaitanis, A., Lentzas, A., Tsoumakas, G., Vrakas, D.: Route planning for emergency evacuation using graph traversal algorithms. Smart Cities **6**(4), 1814–1831 (2023) <https://doi.org/10.3390/smartcities6040084>
- [8] Koutamanis, A., Mitossi, V.: Automated recognition of architectural drawings. In: 1992 11th IAPR International Conference on Pattern Recognition, vol. 1, pp. 660–663. IEEE Computer Society, Los Alamitos, CA, USA (1992). <https://doi.org/10.1109/ICPR.1992.201647>
- [9] Dosch, P., Tombre, K., Ah-Soon, C., Masini, G.: A complete system for the analysis of architectural drawings. International Journal on Document Analysis and Recognition **3**(2), 102–116 (2000) <https://doi.org/10.1007/PL00010901>
- [10] Or, S.-h., Wong, K.-H., Yu, Y.-k., Chang, M.M.-y., Kong, H.: Highly automatic approach to architectural floorplan image understanding & model generation. Pattern Recognition, 25–32 (2005)
- [11] Ahmed, S., Liwicki, M., Weber, M., Dengel, A.: Improved automatic analysis of

- architectural floor plans. In: 2011 International Conference on Document Analysis and Recognition, pp. 864–869 (2011). <https://doi.org/10.1109/ICDAR.2011.177>
- [12] Macé, S., Locteau, H., Valveny, E., Tabbone, S.: A system to detect rooms in architectural floor plan images. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. DAS '10, pp. 167–174. Association for Computing Machinery, New York, NY, USA (2010). <https://doi.org/10.1145/1815330.1815352>
 - [13] Ahmed, S., Liwicki, M., Weber, M., Dengel, A.: Automatic room detection and room labeling from architectural floor plans. In: 2012 10th IAPR International Workshop on Document Analysis Systems, pp. 339–343 (2012). <https://doi.org/10.1109/DAS.2012.22>
 - [14] Sarker, S., Sarker, P., Stone, G., Gorman, R., Tavakkoli, A., Bebis, G., Sattarvand, J.: A comprehensive overview of deep learning techniques for 3d point cloud classification and semantic segmentation. Machine Vision and Applications **35**(4), 67 (2024) <https://doi.org/10.1007/s00138-024-01543-1>
 - [15] Bostik, O., Valach, S., Horak, K., Klecka, J.: Segmentation method overview for thermal images in matlab computational environment. MENDEL **25**(1), 43–50 (2019) <https://doi.org/10.13164/mendel.2019.1.043>
 - [16] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25 (2012). <https://doi.org/10.1145/3065386>
 - [17] Haskins, G., Kruger, U., Yan, P.: Deep learning in medical image registration: a survey. Machine Vision and Applications **31**(1), 8 (2020) <https://doi.org/10.1007/s00138-020-01060-x>
 - [18] Kannala, J.: Cubicasa5k: A dataset and an improved multi-task model for floor-plan image analysis. In: Image Analysis: 21st Scandinavian Conference, SCIA 2019, Norrköping, Sweden, June 11–13, 2019, Proceedings, vol. 11482, p. 28 (2019). https://doi.org/10.1007/978-3-030-20205-7_3. Springer
 - [19] Liu, C., Wu, J., Kohli, P., Furukawa, Y.: Raster-to-vector: Revisiting floor-plan transformation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2195–2203 (2017). <https://doi.org/10.1109/ICCV.2017.241>
 - [20] Lv, X., Zhao, S., Yu, X., Zhao, B.: Residential floor plan recognition and reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16717–16726 (2021)
 - [21] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder

- with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 801–818 (2018). https://doi.org/10.1007/978-3-030-01234-2_49
- [22] Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020) <https://doi.org/10.48550/arXiv.2004.10934>
 - [23] Zeng, Z., Li, X., Yu, Y.K., Fu, C.-W.: Deep floor plan recognition using a multi-task network with room-boundary-guided attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9096–9104 (2019). <https://doi.org/10.1109/ICCV.2019.00919>
 - [24] Park, S., Kim, H.: 3dplannet: Generating 3d models from 2d floor plan images using ensemble methods. *Electronics* **10**(22) (2021) <https://doi.org/10.3390/electronics10222729>
 - [25] Yang, B., Jiang, T., Wu, W., Zhou, Y., Dai, L.: Automated semantics and topology representation of residential-building space using floor-plan raster maps. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **15**, 7809–7825 (2022) <https://doi.org/10.1109/JSTARS.2022.3205746>
 - [26] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations, ICLR 2015 (2015)
 - [27] Liu, C., Schwing, A.G., Kundu, K., Urtasun, R., Fidler, S.: Rent3d: Floor-plan priors for monocular layout estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3413–3421 (2015). <https://doi.org/10.1109/CVPR.2015.7298963>
 - [28] Yang, B., al.: Vectorfloorseg: Two-stream graph attention network for vectorized roughcast floorplan segmentation. In: CVPR, pp. 1358–1367 (2023)
 - [29] Asmar, D., Daher, R., Hawari, Y., Khoury, H., Elhajj, I.H.: Automated building and evaluation of 2d as-built floor plans. *Machine Vision and Applications* **33**(3), 36 (2022) <https://doi.org/10.1007/s00138-022-01289-8>
 - [30] Amer, A., Lambrou, T., Ye, X.: Mda-unet: A multi-scale dilated attention u-net for medical image segmentation. *Applied Sciences* **12**(7), 3676 (2022) <https://doi.org/10.3390/app12073676>
 - [31] Li, R., Duan, C., Zheng, S.: Macu-net semantic segmentation from high-resolution remote sensing images. arXiv preprint arXiv:2007.13083 (2020)
 - [32] Qin, C., Wu, Y., Liao, W., Zeng, J., Liang, S., Zhang, X.: Improved u-net3+ with stage residual for brain tumor segmentation. *BMC Medical Imaging* **22**(1), 1–15

- (2022) <https://doi.org/10.21203/rs.3.rs-898744/v1>
- [33] Ding, X., Guo, Y., Ding, G., Han, J.: Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1911–1920 (2019). <https://doi.org/10.48550/arXiv.1908.03930>
 - [34] Zhang, Y., He, Y., Zhu, S., Di, X.: The direction-aware, learnable, additive kernels and the adversarial network for deep floor plan recognition. arXiv preprint arXiv:2001.11194 (2020) <https://doi.org/10.48550/arXiv.2001.11194>
 - [35] Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018). <https://doi.org/10.48550/arXiv.1807.06521>
 - [36] Li, R., Zheng, S., Zhang, C., Duan, C., Wang, L., Atkinson, P.M.: Abcnet: Attentive bilateral contextual network for efficient semantic segmentation of fine-resolution remotely sensed imagery. ISPRS journal of photogrammetry and remote sensing **181**, 84–98 (2021)
 - [37] Ma, Z., Chang, D., Xie, J., Ding, Y., Wen, S., Li, X., Si, Z., Guo, J.: Fine-grained vehicle classification with channel max pooling modified cnns. IEEE Transactions on Vehicular Technology **68**(4), 3224–3233 (2019)
 - [38] Qian, L., al.: Multi-scale context unet-like network with redesigned skip connections for medical image segmentation. Computer Methods and Programs in Biomedicine **243**, 107885 (2024)
 - [39] Yeung, M., Sala, E., Schönlieb, C.-B., Rundo, L.: Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. Computerized Medical Imaging and Graphics **95**, 102026 (2022) <https://doi.org/10.1016/j.compmedimag.2021.102026>
 - [40] Bulat, A., Tzimiropoulos, G.: Human pose estimation via convolutional part heatmap regression. In: European Conference on Computer Vision, pp. 717–732 (2016). Springer
 - [41] Ke, L., Chang, M.-C., Qi, H., Lyu, S.: Multi-scale structure-aware network for human pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 713–728 (2018)
 - [42] Nibali, A., He, Z., Morgan, S., Prendergast, L.: 3d human pose estimation with 2d marginal heatmaps. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1477–1485 (2019). IEEE
 - [43] Bolelli, F., Allegretti, S., Baraldi, L., Grana, C.: Spaghetti labeling: Directed

acyclic graphs for block-based connected components labeling. *IEEE Transactions on Image Processing* **29**, 1999–2012 (2019) <https://doi.org/10.1109/TIP.2019.2946979>

- [44] Suzuki, S., *et al.*: Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing* **30**(1), 32–46 (1985)
- [45] Liebel, L., Körner, M.: Auxiliary tasks in multi-task learning. arXiv preprint arXiv:1805.06334 (2018)
- [46] Ke, T.-W., Hwang, J.-J., Liu, Z., Yu, S.X.: Adaptive affinity fields for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 587–602 (2018). https://doi.org/10.1007/978-3-030-01246-5_36
- [47] Toussaint, G.T.: Solving geometric problems with the rotating calipers. In: Proc. IEEE Melecon, vol. 83, p. 10 (1983)
- [48] Leung, L.: TF2DeepFloorplan License: GPL v3. Open source software available from <https://github.com/zcemycl/TF2DeepFloorplan> (2023)
- [49] Kannala, J.: CubiCasa5k. Open source software available from <https://github.com/CubiCasa/CubiCasa5k> (2019)
- [50] Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: 36th International Conference on Machine Learning, ICML 2019, pp. 6105–6114 (2019)
- [51] Heras, L.-P., Terrades, O., Robles, S., Sánchez, G.: Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool. *International Journal on Document Analysis and Recognition (IJDAR)* **18** (2015) <https://doi.org/10.1007/s10032-014-0236-5>
- [52] Dodge, S., Xu, J., Stenger, B.: Parsing floor plan images. In: 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), pp. 358–361 (2017). <https://doi.org/10.23919/MVA.2017.7986875>. IEEE
- [53] LIFULL Co., L.: LIFULL HOME'S Dataset (collection) (2015)
- [54] Swaileh, W., Kotzinos, D., Ghosh, S., Jordan, M., Vu, N.-S., Qian, Y.: Versailles-fp dataset: Wall detection in ancient floor plans. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) *Document Analysis and Recognition – ICDAR 2021*, pp. 34–49. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86549-8_3
- [55] Pizarro, P.N., Hitschfeld, N., Sipiran, I.: Large-scale multi-unit floor plan dataset for architectural plan analysis and recognition. *Automation in Construction* **156**, 105132 (2023) <https://doi.org/10.1016/j.autcon.2023.105132>

- [56] Lu, Z., Wang, T., Guo, J., Meng, W., Xiao, J., Zhang, W., Zhang, X.: Data-driven floor plan understanding in rural residential buildings via deep recognition. *Information Sciences* **567**, 58–74 (2021)
- [57] Chinchor, N., Sundheim, B.M.: Muc-5 evaluation metrics. In: Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993 (1993). <https://doi.org/10.3115/1072017.1072026>
- [58] Jaccard, P.: The distribution of the flora in the alpine zone. 1. New phytologist **11**(2), 37–50 (1912)
- [59] Wu, Y., He, K.: Group normalization. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018). <https://doi.org/10.48550/arXiv.1803.08494>
- [60] Foundation, O.C.V.: mmcv. Open source software available from <https://github.com/open-mmlab/mmcv/tree/v1.6.2> (2022)
- [61] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>. Ieee