15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

# Parsing Floor Plan Images

Samuel Dodge
Arizona State University

Jiu Xu
Rakuten Institute of Technology

Björn Stenger
Rakuten Institute of Technology

## Abstract

*This paper introduces a method for analyzing floor plan images using wall segmentation, object detection, and optical character recognition. We introduce a challenging new real-estate floor plan dataset, R-FP, evaluate different wall segmentation methods, and propose fully convolutional networks (FCN) for this task. We explore architectures with different pixel-stride values and more compact ones with skipped pooling layers. An FCN-2s with a 2-pixel stride layer achieves state-of-the-art performance, obtaining a mean Intersection-over-Union score of 89.9% on R-FP, and 94.4% on the public* CVC-FP *data set. Using OCR and object detection, we estimate room sizes. Finally, we show applications in automatic 3D model building and interactive furniture fitting.*

## 1 Introduction

Architectural floor plans are scaled drawings of apartments or building layouts. They contain structural and semantic information, *e.g.* room types and sizes, and the location of doors, windows, and fixtures. Floor plans are a common tool for real estate agents in selling or renting out a space. Parsing such images automatically has a number of applications, such as similarity search [1], CAD model generation, and 3D model creation for rendering and interactive walkthroughs [6, 8, 10]. Floor plan analysis has been an active research topic in the area of document processing, but has mainly focused on processing high-resolution scans. In this paper we instead focus on readily available floor plan images from real estate websites. These plans were created in a number of different drawing styles and are at lower resolution than standard architectural documents. Previous methods that rely on image binarization fail on most of these images. We propose a learning-based approach to segment walls and detect objects. Deep networks have been shown to perform well on semantic segmentation tasks [2, 12, 14]. Specifically, we train fully convolutional networks (FCN), explore different network architectures, and evaluate their performance, comparing them with baseline methods. The main contributions are: (1) a single method using an FCN for segmenting walls in different drawing styles with state-of-the-art performance, (2) a new dataset of 500 labeled floor plan images from a real-estate website, and (3) applications in automatic 3D model generation and interactive furniture fitting. Having extracted the sizes of walls from OCR, we are able to place furniture items into the model at the correct scale.

## 2 Prior work

Early systems were designed for converting 2D floor plans to 3D interactively, *e.g.* [6, 9, 10]. Two survey papers provide a good review of methods for generating models from architectural plans [7, 15]. Many solutions share a common pipeline which includes binarization, structure and text separation, symbol recognition and vectorization. Within the *ScanPlan* project, Maće *et al.* detect walls and doors using the Hough transform [11]. Wall polygons are partitioned iteratively into rooms, assuming convex room shapes. Ahmed *et al.* [1] process high-resolution images by segmenting lines according to their thickness, followed by geometrical reasoning to segment rooms. Doors are detected using SURF descriptors. In our case, the input images are of lower resolution and standard binarization fails on many images. De las Heras *et al.* [4] proposed segmenting walls using the assumption of them being a repetitive element, modeled by straight parallel lines. This method performs well on high-resolution images in different graphical styles. Similarly, Gimenez *et al.* [8] binarize the floor plan images and detect walls by looking for parallel lines separated by a distance within a predefined range. Wall thickness is determined automatically by clustering the distance values. Unfortunately, the assumption of consistent wall appearance does not hold in many of our input images. In [3] De las Heras *et al.* propose classifying image patches using a bag-of-words (BoW) model. This BoW model is tuned to each particular graphical style in the CVC-FP data set, and is trained on each subset of wall types with different parameters. Our method, by contrast, is learned only once for all graphical styles in the training set.

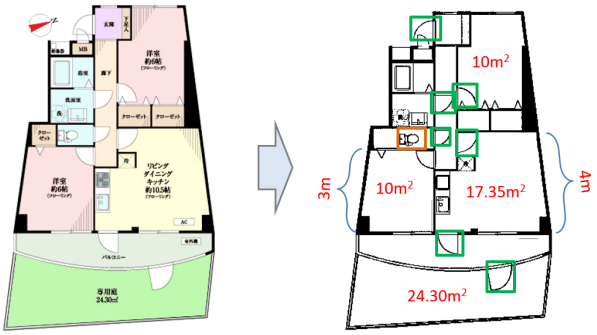


**Figure 1: Parsing a floor plan image:** *The goal is to obtain a model by segmenting walls, recognizing objects, and estimating dimensions using OCR.*

## 3 Parsing floor plans

We combine three methods to extract geometric and semantic information: Wall segmentation, object detection, and optical character recognition (OCR).

**Wall segmentation.** We use fully convolutional networks (FCN) for segmenting wall pixels, following [14]. We train models with different pixel-stride value and compare FCN architectures for different final stride layers to find the best stride value. Starting with a FCN-32s with a 32-pixel stride, initialized with VGG-16 parameters [14], in sequential training each model is initialized with the parameters of the

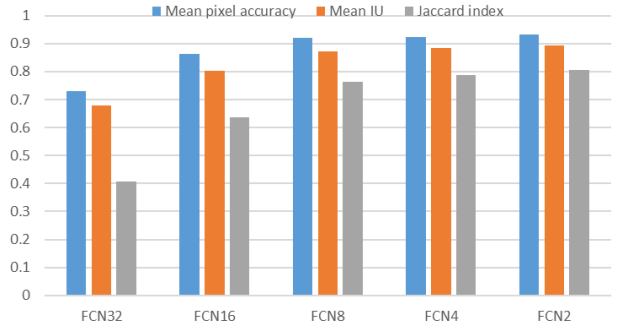


**Figure 2: FCN with different stride values:** *Adding layers with reduced stride increases performance according to all three measures.*

previous one. Following [14] all models are trained by stochastic gradient descent with momentum (value 0.9, weight decay of $5^{-4}$) and a batch size of 1. The learning rates are set to $10^{-8}$, $10^{-9}$, $10^{-10}$, $10^{-10}$, and $10^{-11}$ for FCN-32s to FCN-2s, respectively. These values are chosen by running two training procedures with different learning rates for 10 epochs and chosing the rate that reduces the error more rapidly. To introduce a layer with smaller stride (by a factor of two) we need to fuse two streams within the network: a pooling layer, and a deconvolutional layer. For this we ensure that the sizes and outputs of the layers is consistent. For overviews of the network architectures, please see the supplementary video. The FCNs are compared with these baseline methods: The first baseline uses a global threshold (GT) with the threshold value determined using Otsu's method. We additionally use morphological closing to remove small lines that are not walls (GT+MC). Finally, we remove text regions from the image using the Google Vision API (GT+MC+TR). We also implement a patch-based approach to wall detection, including a BoW approach suggested in [3]. We sample $10 \times 10$ pixel patches from the image such that the patches have at least one dark pixel (other pixels are assumed to be non-wall). We take the PCA of the training patches (1000 per image) and take the principal components that describe 95% of the variance of the data for the CVC-FP data set and 50 components for the R-FP dataset. We compare three binary classifiers to determine which patches are wall or non-wall: random forest (RF), linear support vector machine (SVM), and BoW. During testing we sample the image at a stride of 3 pixels. The final prediction for each pixel is the mean of the predictions from the overlapping boxes that contain the pixel.

**Object detection.** We use the Faster R-CNN framework from [13] for object detection and recognition, using a light-weight ZF network [16] for feature map computation. A subset of the R-FP images was annotated for training and testing, with 6 different object classes (doors, sliding doors, kitchen stoves, bath tubs, sinks, and toilets). The model is trained on 144 images for 150 epochs with ZF networks using top-2000-score Region Proposal Networks.

**Optical character recognition.** Our input images may contain both English and Japanese text. We use the Google Vision API for text detection and character recognition, which handles multiple languages.

### 3.1 Experimental Framework

**Data sets.** For evaluation, we collected a new dataset, *R-FP*, of 500 floor plan images from a public real estate website. These images have different sizes, with side lengths in the range of 156-1427 pixels. The images were created as a tool for real-estate agents and show some degree of artistic freedom, such as the use of different color and shading schemes and decorative elements. Compared with other public datasets, this dataset exhibits a significant amount of variation. Fig. 3 shows example images. We also evaluate on the publicly available *CVC* floor plan data set [5]. This set includes 122 high-resolution images in four different drawing styles.

**Metrics.** As evaluation metrics we use the mean pixel accuracy and mean Intersection-over-Union (IoU), as previously introduced in [14] as well as the Jaccard Index (JI) for wall pixels as proposed in [3].

## 4 Results

**Wall segmentation using FCN with different strides.** Figure 2 shows the increase in mean accuracy, mean IoU, and Jaccard index when progressively adding new layers with smaller stride values. Note that a stride value of 2 is lower than in previous work on semantic segmentation, where a stride value of 8 was suggested [14]. In our layout prediction task, fusing even shallower layers achieves further improvement. These layers contain features representing local details such as corners and edges, which are important for parsing floor maps.

**Wall segmentation on R-FP dataset.** Wall segmentation results on the new R-FP dataset are shown in Table 1. The patch-based random forest classifier slightly outperforms the baseline methods in terms of mean IoU. FCN-2s networks, trained either in stages or at once, perform best. Figure 3 shows sample results by different segmentation methods. FCN-2s at-once training results in comparable performance as FCN-2s trained in stages, while reducing the training time (20h for 400 iterations compared with 35h for staged training, both on a GeForce 1080 GPU). The FCN-2s network segments one image in approximately 0.14s.

**Skip-layer architectures.** For further investigating the importance of fusion by different pooling layers, we test the performance of two alternative network architectures, both skipping particular upsampling streams. In the first, FCN-2s-skip-2-4, *pool2* and *pool4* layers are skipped, which means two $8\times$ and one $2\times$ interpolation layers are placed before the final output layer. In the second architecture, FCN-2s-skip-2-3-4, additionally *pool3* layer is skipped for training, which means one $16\times$ and one $2\times$ interpolation layers are placed before the final output layer. The results in Table 1 show that skipping *pool2* and *pool4* layers results in similar performance to FCN-2s staged training, however, skipping all layers *pool2*, *pool3*, and *pool4* layer leads to poorer performance. The performance of training with skip layers infers some feature redundancy in stream fusion. *FCN-2s-skip-2-4* has a reduced training time of 18h and requires approximately 0.12s for segmenting one image.

**Evaluation on CVC-FP dataset.** Wall segmentation results on the public CVC-FP dataset are shown

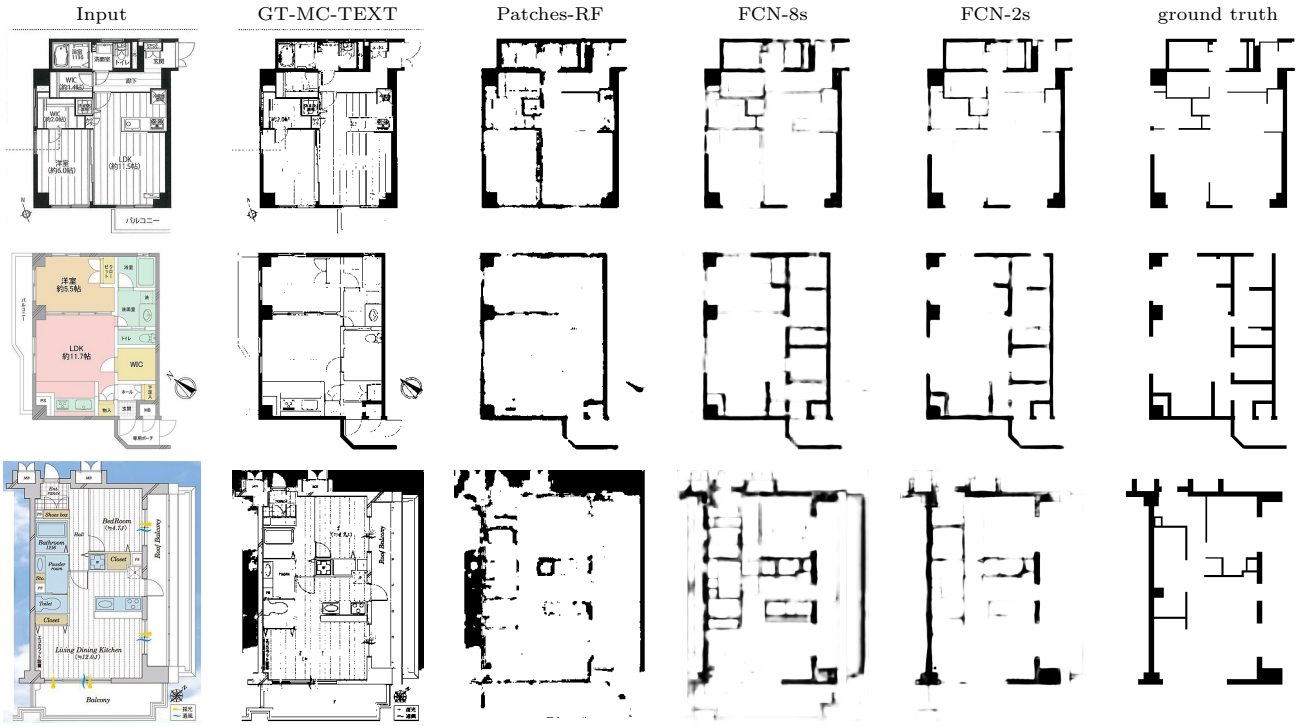


**Figure 3: Wall segmentation on images from the new R-FP data set.** *Global thresholding (GT-MC-TEXT) does not handle decorative lines and symbols. The patch based segmentation (Patches-RF) cannot handle different drawing styles with the same parameter settings. Segmentations by FCN-8s [14] are more blurred than those by the proposed FCN-2s. The bottom row shows difficult example on which all methods perform poorly.*

| Method | Mean acc. | Mean IoU | JI |
|---|---|---|---|
| Global Thresh. (GT) | 90.1 | 73.7 | 53.0 |
| GT+MC | 90.1 | 73.7 | 53.0 |
| GT+MC+TR | 90.3 | 76.9 | 58.3 |
| Patches - RF | 89.7 | 77.6 | 59.3 |
| Patches - SVM | 82.1 | 72.3 | 49.6 |
| Patches - BoW | 89.5 | 75.8 | 56.3 |
| **FCN-2s** | **94.0** | 89.7 | 80.9 |
| **FCN-2s at-once** | 92.9 | **89.9** | **81.2** |
| FCN-2s-skip-2-4 | 93.6 | 89.7 | 81.0 |
| FCN-2s-skip-2-3-4 | 91.4 | 87.3 | 76.6 |

**Table 1: Wall segmentation results on the new real estate floor plan dataset (R-FP)**

| Method | Mean acc. | Mean IoU | JI |
|---|---|---|---|
| GT | 78.7 | 69.6 | 41.5 |
| GT+MC | 95.1 | 80.1 | 61.9 |
| GT+MC+TR | 95.3 | 85.0 | 71.2 |
| Patches - RF | 92.8 | 89.3 | 79.2 |
| Patches - SVM | 92.6 | 89.0 | 78.5 |
| Patches - BoW | 93.0 | 87.9 | 76.6 |
| **FCN-2s** | **97.3** | **94.4** | **89.2** |

**Table 2: Wall segmentation results on the public CVC-FP dataset [5]**

| Training | Test | Mean acc. | Mean IoU | JI |
|---|---|---|---|---|
| CVC | R | 82.7 | 76.1 | 56.0 |
| R + CVC | R | 94.0 | 90.5 | 82.5 |
| R | CVC | 84.2 | 81.7 | 64.7 |
| R + CVC | CVC | 96.0 | 92.9 | 86.3 |

**Table 3: Cross-dataset generalization.** *An FCN-2s model trained on one data set generalizes poorly when testing on the other. Training on the union of the training sets boosts performance.*

in Table 2. FCN-2s is also the top performing method on this data set. Our patch-based baseline does not achieve as good performance as reported by De Las Heras *et al.* [3], however the main difference is that we perform training with a single set of parameters, whereas [3] choose different parameters for different image subsets. Another difference is the evaluation protocol. We perform 5-fold cross validation, whereas [3] perform leave-one-out cross validation for certain subsets of the dataset.

**Cross-dataset performance.** Table 3 shows the performance of the FCN-2s model trained on one dataset and tested on the other. Performance is poor, largely owing to the differences in the drawing styles. The R-FP dataset has many different types of wall images. Furthermore, the R-FP dataset is of floorplans in Japan and the CVC-FP dataset is of floorplans from Europe. The CVC-FP dataset consists of high quality architectural floorplans, whereas the R-FP dataset consists of floorplans from an internet real-estate website. We also train a FCN-2s on the combined training data, resulting in significantly improved results on both test sets, in particular achieving better performance on R-FP than the model trained on R-FP alone.

**Object detection performance.** The average precision evaluated on 25 test images is 96.0% for doors, 35.9% for sliding doors, 76.2% for kitchen ovens, 95.8% for bath tubs, 69.2% for sinks, and 70.8% for toilets,

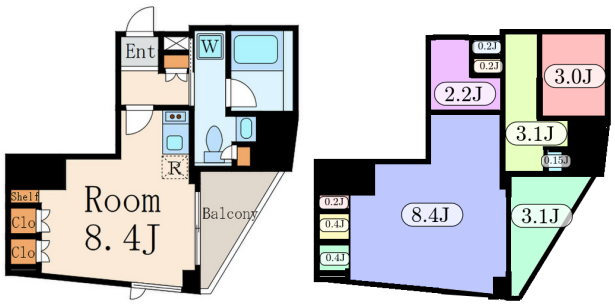


**Figure 4: Room size estimation example:** *room sizes read with OCR in the input image (left) are propagated to unlabeled rooms (right).*

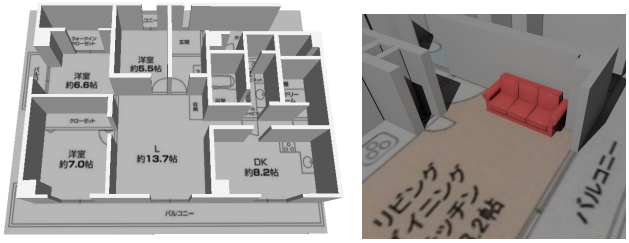

**Figure 5: 3D model creation:** *Wall segmentation allows automatic 3D model creation from floor plans (left). OCR of room measurements allows inserting 3D furniture models scaled to the scene (right).*

for an IoU value of 50%. Using 300 region proposals, the average detection time is about 50ms on a GeForce 1080 GPU.

**Text detection performance.** We labeled 20 images from the R-FP dataset with text locations and values. For our applications (3D model generation and furniture fitting), the most important text is the room size. The OCR is able to detect and recognize 74.1% of the room size annotations. Note that many floor plans contain multiple size annotations.

### 4.1 Applications: 3D model creation and furniture fitting

Using the wall segmentation from Section 3, we can create an approximate 3D model of home layout *solely* from the floorplan. A true 3D reconstruction of the layout requires a-priori knowledge of the wall height. In absence of this information, we arbitrarily choose a height. To create the 3D model, we consider the wall segmentation result as an elevation map, where all of the walls are of uniform height. Figure 5 (left) shows an example of fully automatic reconstruction.

Secondly using the segmentation information and text information we can infer the area of rooms, as well as the length of walls (Figure 4). Rooms are delimited by walls, (sliding) doors and windows. For each segmented room, we query the text information for the Japanese room measurement unit (*Jo*). We compute the relationship between the room size in physical units and pixels to compute the pixel density (in pixel/J). This pixel density can be used to compute an estimate of the area of rooms that are not labeled with physical units, and well as can be used to compute wall length. In addition, with the room size information we can estimate the fit of furniture in a room (Figure 5, right). Please see the supplemental video for further results.

## 5 Conclusion

We show that an FCN with stride of 2 achieves better performance than a stride of 8 as in [14]. The smaller stride is effective because some walls can be easily described by low level features. The FCN model combines features from subsequent layers that describe contextual information and texture information with these low level features. Secondly, the smaller stride allows us to predict more precise wall locations. Walls are often very thin structures that cannot be precisely predicted at larger strides.

From the floor plan we extract a parsed representation of wall locations, objects, and size information. We show two example applications using this information, 3D model creation and furniture fitting.

## References


[1] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel. Automatic room detection and room labeling from architectural floor plans. In *Int. Workshop on Document Analysis Systems*, pages 339–343, 2012.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for scene egmentation. In *TPAMI*, 2017.

[3] L.-P. De Las Heras, D. Fernández, E. Valveny, J. Lladós, and G. Sánchez. Statistical segmentation and structural recognition for floor plan interpretation. In *IJDAR*, pages 1–17, 2013.

[4] L.-P. De Las Heras, D. Fernández, E. Valveny, J. Lladós, and G. Sánchez. Unsupervised wall detector in architectural floor plans. In *ICDAR*, 2013.

[5] 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. In *IJDAR*, 2015. `http://dag.cvc.uab.es/resources/floorplans/`.

[6] P. Dosch, K. Tombre, C. Ah-Soon, and G. Masini. A complete system for the analysis of architectural drawings. In *Int. J. Document Analysis and Recognition*, volume 9, pages 102–116, 2000.

[7] L. Gimenez, J.-L. Hippolyte, S. Robert, F. Suard, and K. Zreik. Review: reconstruction of 3D building information models from 2D scanned plans. In *J. Build. Eng.*, volume 0, pages 48–56, 2015.

[8] L. Gimenez, S. Robert, F. Suard, and K. Zreik. Automatic reconstruction of 3D building models from scanned 2D floor plans. In *Automation in Construction*, volume 63, pages 48–56, 2016.

[9] H. Goto, K. H. Law, and G. Brickey. Knowledge-based creation of an architectural 3-D model from 2-D drawings. Technical Report TR-59, Stanford University, 1991.

[10] R. Lewis and C. Séquin. Generation of 3D building models from 2D architectural plans. In *Computer-Aided Design*, volume 30, pages 765–779, 1998.

[11] S. Macé, H. Locteau, E. Valveny, and S. A. Tabbone. A system to detect rooms in architectural floor plan images. In *Int. Workshop on Document Analysis Systems*, 2010.

[12] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *CVPR*, 2015.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[14] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional models for semantic segmentation. In *TPAMI*, 2016.

[15] X. Yin, P. Wonka, and A. Razdan. Generating 3D building models from architectural drawings: A survey. In *IEEE CGA*, pages 20–30, 2009.

[16] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.