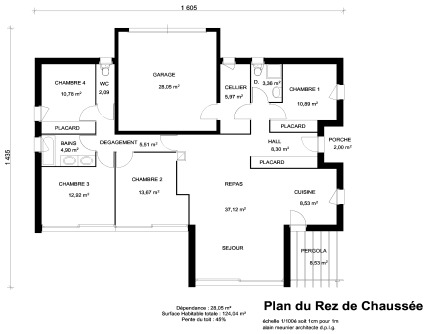
Deep convolutional neural network for wall segmentation in architectural floors

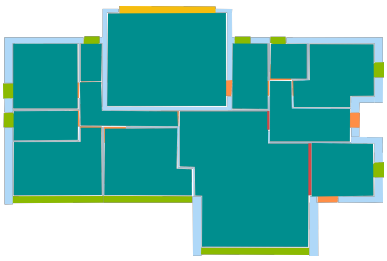
Cecilia Ferrando
School of Architecture
Carnegie Mellon University
Pittsburgh, PA 15213
cferrand@andrew.cmu.edu

Donghan Wang
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
donghanw@cs.cmu.edu

Matthew Hanczor
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
mhanczor@andrew.cmu.edu



(a) A floor plan



(b) The ground truth of the floor plan shows the structure. Objects such as walls, windows and doors are colored.

Figure 1: Segmenting walls in a floor plan image: The goal is to train a wall segmentation model using FCNs. This is a floor plan (LEFT) and its label (RIGHT) from the CVC-FP dataset.

1 Introduction

The automation of architectural document analysis is a niche but increasingly explored research field. Architectural floor plans are scaled graphical documents containing semantic information such as the location of walls, doors, windows and other structural or technical parts of a building.

The semantic segmentation of architectural floor plans has already been treated in previous research. The majority of the proposed approaches address the problem through image processing, and proceed with binarization, structure and text separation, symbol recognition and vectorization. Mace et al., for example, detect walls and doors by using the Hough transform, and assume convexity of the room shapes to iteratively isolate the rooms [1].

De las Heras [2] takes advantage of the assumption that walls are a system of straight parallel/perpendicular elements to perform wall segmentation. The same authors classify image patches using a bag-of-words (BoW) model for statistical segmentation of the floors. While this model proposed by De las Heras need to be trained over different types of representation styles, in Stenger et al. [3] this limitation is overcome.

Automating the analysis of floor plan images has several useful applications, such as

- vector 2D floor plan reconstruction from raster floor plan images
- 2D floor plan reconstruction from sketches [4]

10707 Class Project (2017).

- 3D reconstruction from a 2D floor plan [5]
- room area detection
- extraction of room relationship graphs [6]
- similarity searching among a database of architectural floor plans

2 Related work

Wall segmentation and room detection tasks belong to the growing research field in document recognition and processing. The diffusion of neural network research across fields has recently opened perspectives in the application of deep learning methods to problems of document recognition. However, the problem of automating the segmentation and recognition of architectural documents existed earlier and was addressed through more traditional computer vision methods. In [7], the authors segment lines according to their thickness, followed by geometrical reasoning to segment rooms. Doors are detected using SURF descriptors. The method has poor performance in low resolution. In 2013, De la Heras et al. [2] 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.

The following year, the same authors proposed classifying image patches using a bag-of-words (BoW) model [8]. 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. The model has two drawbacks. First, a separate model will have to be trained for each plan type. Second, trained models may not work well on types not seen in the training data.

As computer vision very often relies on manual tuning of its model parameters to fit specific types of images and problems, its limitations are significant when it comes to analyze and process big heterogeneous data sets of architectural floor plans. Deep networks have been shown to perform well on semantic segmentation tasks [9], [2], [10]. Recently [11] proposed a method of using FCNs for segmenting walls in different drawing styles and reported state-of-the-art performance.

More broadly, when considering spatial analysis and reconstruction as related to architectural spaces, deep learning has proved to have useful applications beyond floor plan segmentation. Some examples are 3D reconstruction of rooms from 2D plans ([12]), and also 2D reconstruction of plans from scene images. Trainable encoder-decoder networks have been applied for 2D layout estimation starting from 3D images [13].

In this project, our main reference is [11]. This paper explores different neural network architectures, trained and tested on two datasets, the Rakuten Floor Plan dataset (R-FP) and the publicly available CVC Floor Plan dataset (CVC-FP). The authors were able to achieve state-of-the-art performance of an 89.9% Intersection-over-Union (IoU) score on R-FP, and 94.4% on CVC-FP.

3 Development of the Method

The basis of our approach is a continuation of the work done by Dodge et. al. [3], who looked at the task of segmenting walls in architectural floor plan images. The authors here used fully convolutional network (FCN) architectures proposed by [10], where every layer in the model is a convolutional layer, and the model output is a pixel-by-pixel classification in the same resolution as the input image. By using this approach the authors were able to show it was possible to accurately extract only the wall information from architectural floor plans using a model trained end-to-end, pixel-to-pixel.

In this section we describe how FCNs are used for pixelwise semantic segmentation and our implementations using pre-trained models on the PASCAL VOC dataset.

Fully Convolutional Networks In [10], Shelhamer et. al. demonstrated traditional convolutional neural networks (CNN) that perform well on whole image classification tasks could be retooled to act as single pixel classifiers. By considering the fully connected layers of CNNs as convolutional kernels that span the entire input image, it is possible to see the output of these layers as a classification map of each individual pixel.

(a) Original image (b) FCN segmentation results

Figure 2: Segment images using the PASCAL trained FCN.

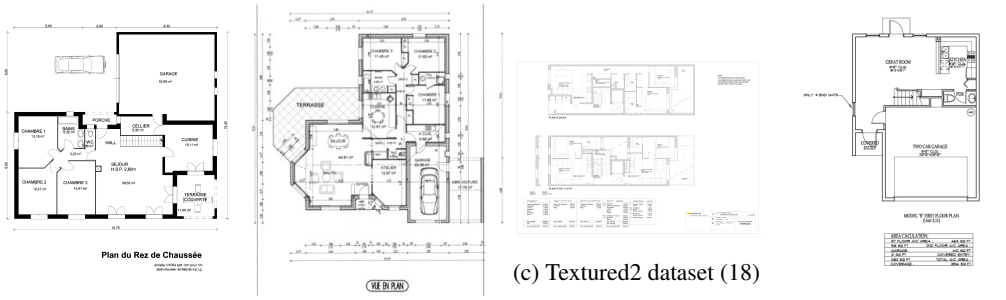
Most standard CNNs downsample the input images in convolutional or pooling layers. This leads to a coarse outputs or, in other words, a significant reduction in output resolution. To address this problem, upsampling is done by reversing the forward and backwards passes of standard convolution to form *deconvolution* layers. By utilizing these layers the network is able to more finely classify pixels. As with convolutional layers though, the stride of the filter can affect the output. Both prior works looked at several different strides for the output upsampling, with strides of 8 and 2 performing well for their respective tasks.

Implementing FCNs for Segmentation In our work we implemented the FCN versions of two popular convolutional neural network classification architectures, VGG-16 and AlexNet. These models both had pre-trained weights from the PASCAL dataset. After initial experiments we found the size and complexity of VGG-16 to be too slow to train and did not seem to perform notably better for our dataset. We therefore used AlexNet for the majority of our work. We choose these models as they showed a high mean accuracy and the best intersection over union accuracy on the PASCAL dataset. Using an FCN model allows for arbitrarily sized inputs since the layers themselves do not depend on a fixed input.

The output layer is the same spatial size as the input. To compute the loss function, we sum the loss over the entire spatial dimension of the final layer. While our inputs are binary or grayscale images with one channel, our output needs to be a softmax classification layer. In the original implementation the authors were trying to classify one of 21 image classes including a background class. In our case we are trying to classify wall or no wall which is a binary classification problem. We initially also expected to expand this to classify and segment several different classes such as a wall, window, door, partition, etc. but due to technical issues and long training times for the binary case we were unable to demonstrate the model with multiple classes. Cross entropy loss is used for the binary case, which allows this model to be easily expanded to the multi-class case later on.

Due to our limited dataset size (Section 4), we used a training dataset of 321 images, and held out a validation dataset of 35 images to train hyperparameters. We demonstrate results below on the validation data as we were not able to also hold out extra images for a test set due to size.

To start, we show that we were able to perform image segmentation using the Caffe version of the FCN AlexNet model provided by [10]. This model is pretrained using 9,600 images from the PASCAL training dataset [14]. We demonstrated the ability to segment images using the PASCAL classes that this model was trained to predict. Figure 2 shows the result of one such classification, where each color in the output image represents a classification from the 21 possible PASCAL classes.



(a) Black dataset (90)

(b) Textured dataset (10)

(c) Textured2 dataset (18)

(d) Parallel dataset (4)

Figure 3: The CVC-FP [15] dataset includes 122 high-resolution images in four types. The number in the parenthesis indicates the number of images.

4 Datasets and evaluation metrics

There were two main datasets we used for training and evaluation, CVC-FP [15] and R-FP [11], comprising a total of 622 labeled images.

4.1 CVC-FP dataset

We trained and evaluated the models first on the CVC-FP data [15]. This set includes 122 high-resolution images in four different drawing styles. It contains documents of different qualities, resolutions, and modeling styles. The dataset is fully labeled for the structural symbols: rooms, walls, doors, windows, parking doors, and room separations.

1. **Black dataset** consists of *90* floor plan binary images of good quality. The size of these images is $2,480 \times 3,508$ or $3,508 \times 2,480$ pixels depending on the orientation of the building.

   Concerning the structural symbols, walls are mostly modeled by black lines of three different thicknesses whether they are main, interior, or exterior walls.

2. **Textured dataset** consists of *10* poor quality and grayscale images whose resolutions can vary from $1,098 \times 905$ pixels the smallest to $2,218 \times 2,227$ the largest. Walls are modeled by two parallel lines with a diagonal line pattern in between for the exteriors, and a heterogeneous gray-dotted pattern for the interiors.

3. **Textured2 dataset** consists of *18* images of $7,383 \times 5,671$ pixels collected from a local architectural project in Barcelona. The singularity of this dataset is that the 18 floor plans belong to a single building of six floors. Walls are modeled similar to the Textured Dataset, this time with a higher frequency diagonal pattern between the two parallel lines.

4. **Parallel dataset** consists of *4* images of good quality and high resolution $2,550 \times 3,300$ pixels. Walls are modeled by simple parallel lines.

4.2 R-FP dataset

The R-FP dataset includes 500 labeled floor plan images from a real-estate website. These plans were created in a number of different drawing styles and are at lower resolution than standard architectural documents. Figure 4 shows an example.

The R-FP dataset is not a publicly available dataset and required working with the original creators in order to obtain a copy of this set. It was desirable to use this though since it was more than four times the size of the CVC-FP dataset. This dataset only became available to use near the end of the project, but still allowed some time to switch to this larger set for training.

(a) A floor plan (b) Ground truth of pixelwise wall annotation

Figure 4: Rakuten dataset includes 500 labeled floor plan images.

4.3 Data pre-processing

The CVC-FP dataset available to the public consists of high resolution PNG floor plan images and manually created segmentation ground truths. These ground truths are in the SVG file format but do not directly convert into the correct PNG size and resolution with standard methods. This is critical since we are training our network to perform pixelwise classification, and therefore the scale and resolution must be consistent between training data and labels. We ended up creating a function that would, for each SVG ground truth, take the file and then reference the input PNG image file to get the correct size and scale, then rescale the SVG correctly and save as a PNG output. Additionally, these ground truth labeled images were not consistent in terms of color used to represent each class. We therefore processed the images to create consistent color classes for all of the labels for training.

The CVC-FP dataset was very high resolution, and required that the images be scaled down to 20% of their original size to train.

In R-FP dataset, the label images had the same inconsistent color issue as CVC-FP. We processed the images to create consistent color classes for all of the labels for training.

5 Experiments and Results

5.1 Evaluation metrics

Our evaluation metric for the results obtained by training the FCN-32s net is the Sørensen–Dice coefficient (or Dice similarity coefficient). The Dice Similarity between two sets X and Y is defined as

$$s = 2\frac{|X \cap Y|}{|X| + |Y|}$$

In the case of two images, assuming that the two have been binarized so to have pixel value 1 for the class and 0 elsewhere, the Dice similarity is two times the sum of the intersecting pixel values over the sum of all pixel values in the two images.

The Dice similarity coefficient is commonly used in image segmentation, particularly in medical image segmentation. This metric is suitable for the evaluation of our binary problem of segmentation of the walls from non-walls (and similarly for other classes) provided that the test image and the prediction are the same size and are converted to binary images.

5.2 Experiments

One of the first steps in our experiments was to establish a baseline to compare to using more traditional methods. We created a computer vision based program in OpenCV to run on a portion of the dataset (the 90 full-black-walls from CVC-FP) to segment the walls from the rest of the markings on the drawing. The results are shown in Figure 5. This method works quite well for a baseline on the

Figure 5: Segment walls using computer vision based methods. Two results are shown. The original floor plan images are on the left. The detected walls are shown on the right.

90 images, but these images were all drawn from the same dataset and had similar features and line weights. This is important because the parameters of the filters used (Gaussian blurring, denoising, dilation and erosion) had hand tuned parameters that would not work as well in a different dataset, and generalize poorly.

We moved to implementing and training our pretrained version of AlexNet, implemented in Caffe. Initial training with the CVC-FP dataset showed us that our dataset was in fact too small to use as our model was not able to learn anything meaningful, and segmentation tests were not able to recognize walls. Additionally we tried training AlexNet from scratch but again the dataset size was prohibitive. This training was done without GPU resources (as our GPU was being used for fine-tune training) and did not produce any results after 8 hours of training.

Once we were able to get the larger R-FP dataset we moved to fine tuning AlexNet with that dataset. This is when we started to see results and the model began to classify wall sections correctly. Training was run for 200 epochs on the data, with 35 images held out for validation. Using a learning rate of 1e-4, moementum of 0.9, and a weight decay of 1e-6, the model was able to reduce both training and validation error during this time. This training took roughly 8 hours on a GPU. Training time was long in our case since we were using images of all different dimensions which precluded the use of batches. Augmenting the data by scaling or cropping were discussed but decided keep our resources focused on continuing to train our model.

5.3 Results

Visualizations of the results can be found in the appendix. Since the results after the first 200 epochs showed the model was learning we decided to continue training, though first experimented with several hyperparameters to try and improve training. We tried several tests of using different solvers (Adagrad, adaptive momentum), learning rates, and momentums, but found very little change after a few epochs training with each. We decided to maintain the previous learning rate but include adaptive momentum for the remainder of training. We allowed the model to run for another 300 epochs (10 hours). The plot of all 500 training epochs vs training and validation loss is shown in Figure 6. Table 2 summarizes the loss and accuracy on the training and validation data for 50, 200 and 500 epochs.

Epoch	Train Loss	Valid Loss	Valid Accuracy
50	0.236	0.173	92.83%
200	0.119	0.155	93.57%
500	0.143	0.162	93.88%

Table 1: Loss and accuracy on the training and validation data. Loss value is normalized.

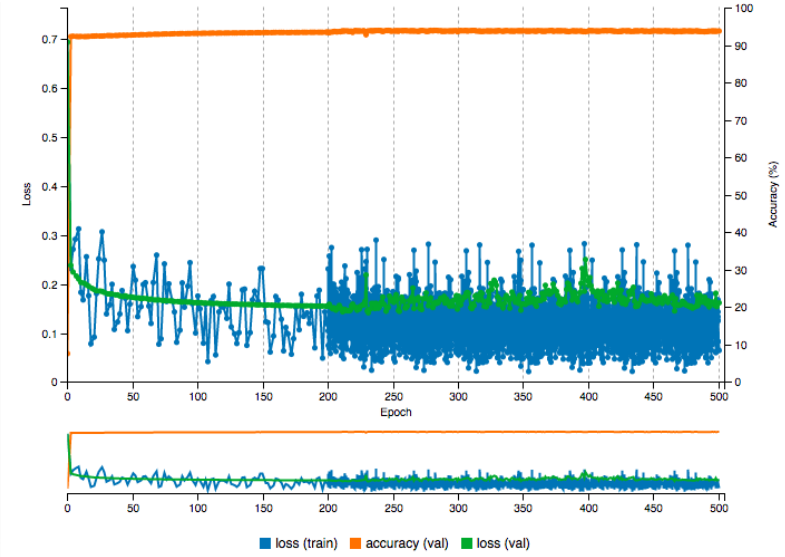


Figure 6: The loss and accuracy on training and validation dataset over 500 epochs. Training was resumed after the first 200 epochs with a more frequent update time, leading to a denser plot after 200 epochs

Results were evaluated through the average Dice Similarity of 20 randomly selected predictions at different epochs, obtaining the following performances:

Epoch	Dice Similarity
50	0.408
200	0.442
500	0.458

Table 2: Average Dice Similarity of test predictions at different epochs

6 Discussion and Analysis

When looking at the results in the appendix it is clear to see that the model is improving over time. At 50 epochs the model was either completely unable to detect walls, or else produced large and/or disjointed segments that did not well represent wall segments. However at 500 epochs the majority of the validation images had walls detected, and wall segments more closely matched the ground truth representations. It is likely that continued training would even further improve the model. Additionally, looking at the Dice similarity in Table 2 we can see that this increases with the number of epochs, showing the model is actually learning.

As mentioned this training was all done on top of the PASCAL pretrained AlexNet. Looking through the PASCAL images though it is clear to see that those images are drastically different than the floor plans we tried to segment. In the PASCAL dataset real world images involving complex shapes and colors are used, while in our case we deal with lots of straight lines and sharp angles. One potential future step would be to pretrain the model using data that would learn features that could be applicable to our dataset. The MNIST dataset could be a good choice for this. This dataset is large, readily accessible, and the content deals with handwritten numbers that produce features like lines

and angles that would be applicable to our work. This data could be augmented to add thin lines or other noise so the model could be pretrained to ignore certain features and only try and produce the noiseless MNIST digits. While not an exact match for our problem, intuitively it seems like this would produce features and filters that would be more useful than the ones from the PASCAL dataset.

The results here were initially intended to be the first in a line of results towards complete floor plan drawing segmentation. As demonstrated, traditional computer vision methods are well suited to remove all but just the wall structure, however they are not as useful in semantically segmenting an image into the more expansive set of classes we hoped to learn (doors, windows, etc.).

The next steps in this work would be to focus on improving wall segmentation accuracy, but in parallel we think progress could be made towards multi-class classification tasks. The way that we set up our approach and how the model was modified was in a way that it would be fairly simple to move to the multi-class segmentation scenario, however in the pursuit of quality results we focused on the binary classification shown here. In the multi-class case, we can base the performance evaluation for room detection on the metrics from common semantic segmentation and scene parsing evaluations that are variations on pixel accuracy and region intersection over union (IU) [10]:

- pixel accuracy: $\sum_i n_{ii} / \sum_i t_i$
- mean accuracy: $(1/n_d) \sum_i n_{ii}/t_i$
- mean IU: $(1/n_d) \sum_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$
- frequency weighted IU: $(\sum_k t_k)^{-1} \sum_i t_i n_{ii}(t_i + \sum_j n_{ji} - n_{ii})$

where n_{ij} is the number of pixels of class i predicted to belong to class j , there are n_{cl} different classes, and $t_i = \sum_j n_{ij}$ is the total number of pixels of class i.

Like most machine learning tasks though, this move to a more complex classification task would require much larger datasets than the ones we were currently working with. To supplement this for the near term however one possible method to try would be to augment the data we currently have. This could also produce more robust models as well if we introduce slight distortions to the image. Overall though the work done here seems promising, and continuing this to the multi-class case could provide a novel method of floor plan analysis.

References

[1] Sébastien Macé, Hervé Locteau, Ernest Valveny, and Salvatore Tabbone. A system to detect rooms in architectural floor plan images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 167–174. ACM, 2010.

[2] Lluís-Pere Heras, David Fernandez, Ernest Valveny, Josep Lladós, and Gemma Sánchez. Unsupervised wall detector in architectural floor plans, 08 2013.

[3] S. Dodge, J. Xu, and B. Stenger. Parsing floor plan images. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 358–361, May 2017.

[4] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE transactions on visualization and computer graphics*, 17(11):1624–1636, 2011.

[5] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. *arXiv preprint arXiv:1707.06375*, 2017.

[6] Qamer Uddin Sabri, Johannes Bayer, Viktor Ayzenshtadt, Syed Saqib Bukhari, Klaus-Dieter Althoff, and Andreas Dengel. Semantic pattern-based retrieval of architectural floor plans with case-based and graph-based searching techniques and their evaluation and visualization. In *ICPRAM*, pages 50–60, 2017.

[7] Sheraz Ahmed, Marcus Liwicki, Markus Weber, and Andreas Dengel. Automatic room detection and room labeling from architectural floor plans. In *Document Analysis Systems*, 2012.

[8] Lluís-Pere de las Heras, Sheraz Ahmed, Marcus Liwicki, Ernest Valveny, and Gemma Sánchez. Statistical segmentation and structural recognition for floor plan interpretation. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(3):221–237, Sep 2014.

[9] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

[10] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1605.06211, 2016.

[11] Samuel Dodge, Jiu Xu, and Björn Stenger. Parsing floor plan images. In *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*, pages 358–361. IEEE, 2017.

[12] Srinidhi Hegde, Saket Anand, and Ojaswa Sharma. Deep learning based 3d reconstruction of indoor scenes. 2017.

[13] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. *arXiv preprint arXiv:1703.06241*, 2017.

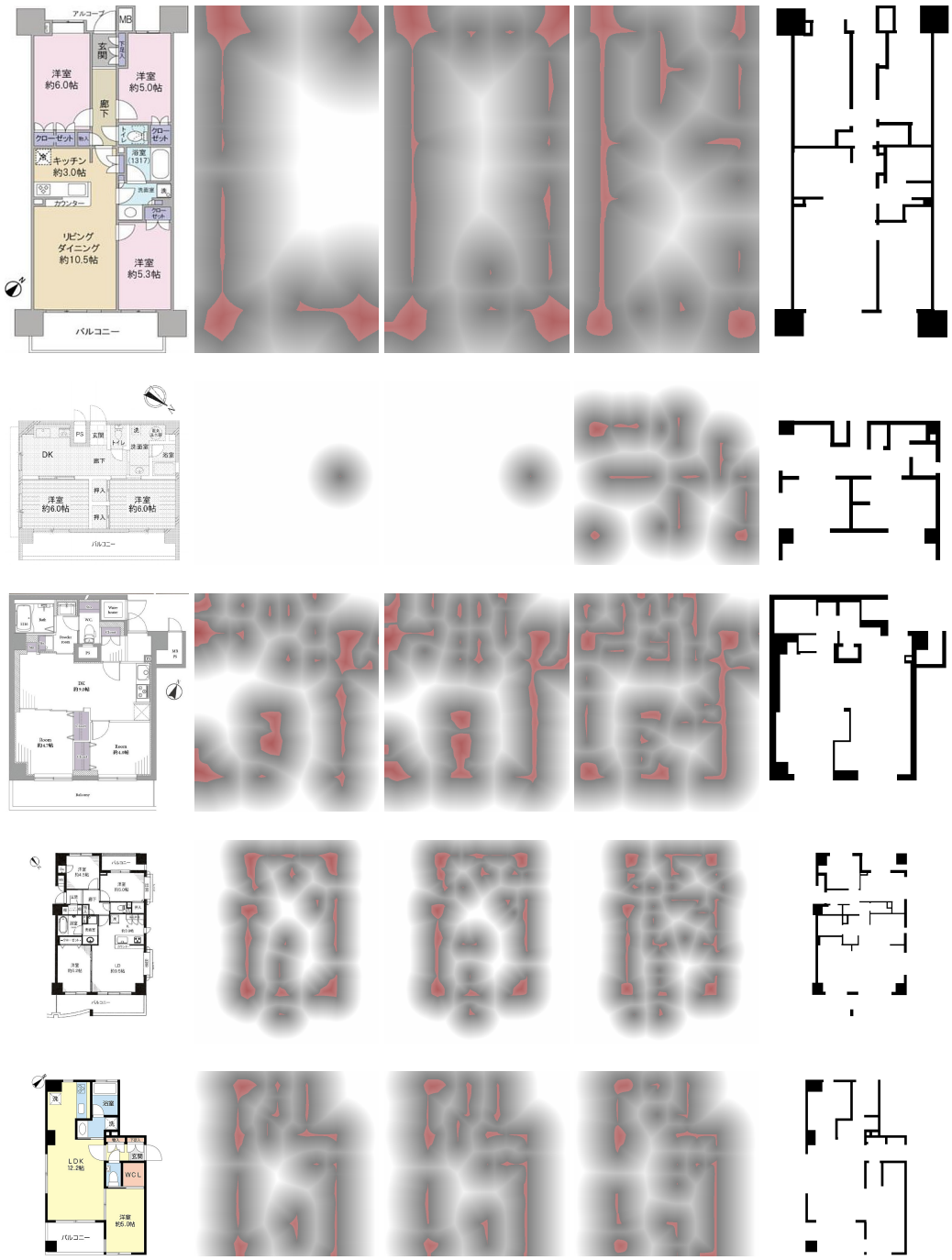
[14] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 991–998. IEEE, 2011.

[15] Lluis-Pere de las Heras, OriolRamos Terrades, Sergi Robles, and Gemma Sanchez. Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool. *International Journal on Document Analysis and Recognition*, 2015.

7 Appendix

7.1 FCNs experiments

Figure 7 depicts the wall segmentation on images from the R-FP data set using models from different epochs.



(a) Input (b) FCN-50epoch (c) FCN-200epoch (d) FCN-500epoch (e) Ground truth

Figure 7: Wall segmentation on images from the R-FP data set. Walls are in maroon; background is in black. The performance increases with more epochs. The edges are sharper and clearer. The improvement is significant for the second image.