Group 22: SpaceXYZ - Analysing Floor Plan Images

Qinjin Jia, Rongyu Wang, Robert LeBourdais, Chi Yu Yeh
{ qjia@bu.edu, wrongyu9@bu.edu, rlebourd@bu.edu, petery@bu.edu}

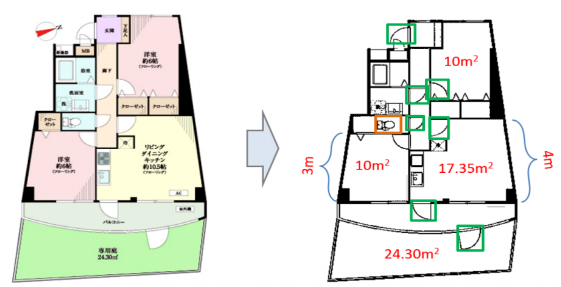


Figure 1. Illustration of the project[1]

1. Project Task

Architectural floor plans are scaled drawings of building layouts. Floor plan analysis has been an active research topic and has a number of applications (e.g. similarity search) [2]. The aim of the project is to parse such floor plans using wall segmentation, object detection and optical character recognition (OCR). More specifically, we aim to recognize the number of main rooms, the type of each room and the surface area of the room from floor plans (see Figure 1) with different resolutions. There are also several optional goals: determining the number of openings (i.e. doors and windows) per room, evaluating the widths and orientations of these openings per the scale and compass in each floor plan, and calculating the perimeter of each room. Constructing an efficient fully convolutional networks (FCN) for wall segmentation, training an R-CNN for object detection, and using OCR for different languages are challenges.

2. Related Work

The project involves three techniques: wall segmentation, object detection, and optical character recognition (OCR). For image preprocessing, Ahmed et. al proposed a Text/Graphics separation method to remove the elements that resemble text characters [6]. In [3], the authors proposed segmenting walls modeled as repetitive elements. A fully convolutional network with a 2-pixel stride layer (FCN-2) proposed in [1] achieves state-of-the-art performance for wall segmentation. [1] employs FCN32, FCN16, FCN8, FCN4, FCN2 deep networks. In sequential training each model is initialized with the parameters of the previous one. Ren et. al. propose a highly performant R-CNN model for object detection [4]. Google's Vision API provides a framework for OCR.

3. Approach

The project involves four techniques: wall segmentation, object detection, our own merging-and-splitting algorithm described below, and optical character recognition (OCR). We train a fully convolutional network(FCN-32) for wall segmentation and a Faster R-CNN object-detection network for doorway identification [1, 2]. Each network takes a floor plan image as input; the first network outputs a corresponding image that contains only walls, and the second network will output an image that contains only doorways and windows. With our merge-and-split algorithm, we then merge these outputs to partition the apartment unit into rooms. In the final algorithm stage, OCR extracts each isolated room's textual label to identify each room type.

3.1 FCN for wall segmentation

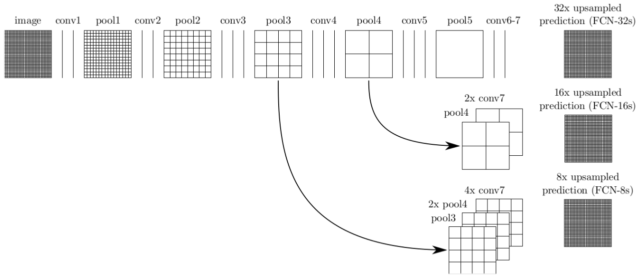


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

Figure 2. Illustration of FCN [7]

"Fully convolutional" networks are very deep models, including both a convolutional network and a deconvolutional network; by virtue of the deconvolutional network, a fully convolutional network can take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. These networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation. The architecture of FCN is shown in Figure 2 above. The pretrained VGG16 model [8] is employed to initialize the FCN32 model and the training points will be used for fine tuning the VGG16 model. Finally, we can extract the wall from the floor plan images with the trained FCN model.

3.2 Faster RCNN for door/window detection

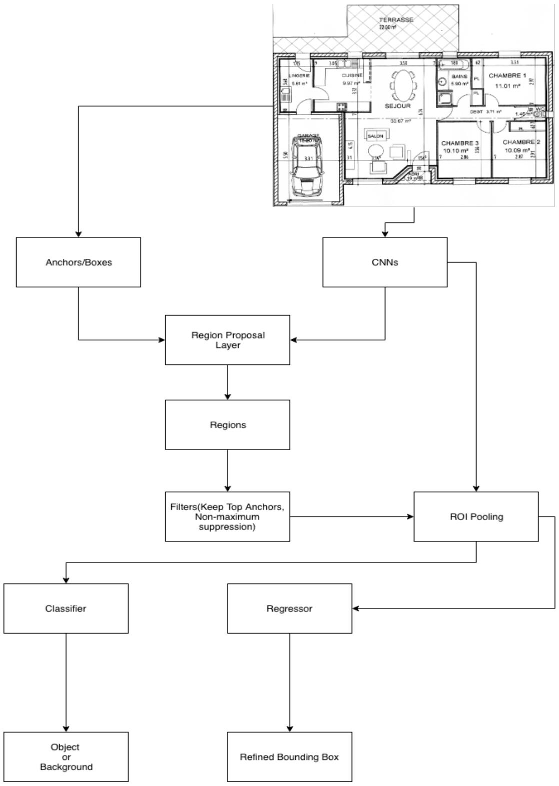


Figure 3. Block diagram of the Faster R-CNN model.

Faster R-CNN represents the state-of-the-art in object detection networks. In our project, we use Faster RCNN to detect objects such as doors, windows, and other non-wall room separators.

As the block diagram in Figure 3 shows, the floor plan image is provided as an input to a convolutional network which provides a convolutional feature map. Instead of using selective-search algorithm on the feature map to identify the region proposals, a separate network is used to predict the region proposals. The predicted region proposals are then reshaped using a RoI pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes.

3.3 CNN with exhaustive search for door localization

To investigate the effectiveness of an alternative method for doorway localization, we also evaluated the performance of a simpler, CNN-based, exhaustive-search algorithm. With the python class *DoorLocalizer* defined in *cnn-door*-posneg.py, we trained and compared the accuracy of several CNN architectures for binary classification of an 80x80 sub-image with classes {Door, Not Door}, in order to determine the most effective network topology for doorway recognition. With the *FindDoors* instance method of the *DoorLocalizer* class, we then applied the most accurate binary classifier in a sliding-window fashion—with a displacement of 5 pixels from one window to the next—to a 1094x905 floor-plan image to estimate door locations in that image, which we then compared to the corresponding ground-truth image. Figure 4 provides a high-level schematic of the exhaustive-search algorithm.

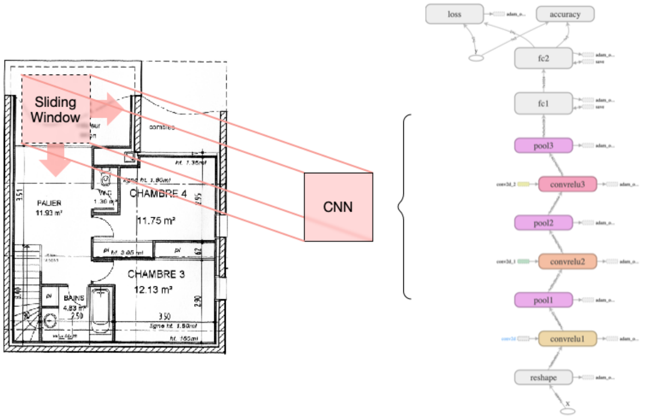


Figure 4. Schematic of the CNN-based, exhaustive-search algorithm for door localization

In order to train the binary classifier for this task, we had to generate suitable training and testing datasets from the CVC-FP dataset. For 80 (*door-ground-truth-image*, *original-image*) pairs from the CVC-FP dataset, the *build_door_data.m* MATLAB script uses the *ground-truth-door-image* to mask the *original-image* and then extract 80x80 pixel example images of doors (centered on the corresponding ground-truth door's centroid) and not doors; to expand the dataset even further, each door image is shifted by 10 pixels in each direction (up, down, left, and right) and rotated by 0, 90, 180, and 270 degrees, allowing us to generate 36 unique door images from each door in the dataset with the goal that the CNN will better learn to recognize doors at different orientations and viewed with different offsets with respect to the door's centroid. With this approach, we were able to generate 16,698 example images, split evenly between positive and negative examples. These images were further split divided into two groups, with 80% of the examples dedicated toward training and 20% dedicated toward testing. The CNN was then trained in mini-batches of size 50.

3.4 Optical character recognition

Room names and sizes are extracted from the floor plan images with OCR API provided by Google Cloud.

3.5 Merging and Splitting

Once we have the wall-prediction and doorway/window prediction outputs from the FCN and R-CNN networks, respectively, we combine them via simple image addition; any narrow gaps between doorways and walls are then eliminated via a morphological close operation, with an example result shown at the left of Figure 5. We then flood-fill the background of that image and invert the result to produce an image like the one to the right in Figure 5. Finally, working from the image at the right of Figure 5, we flood-fill each white region with a different color, and then separate the different rooms based on color. The *flood_fill.m* MATLAB script performs the merge-and-split operation just described.

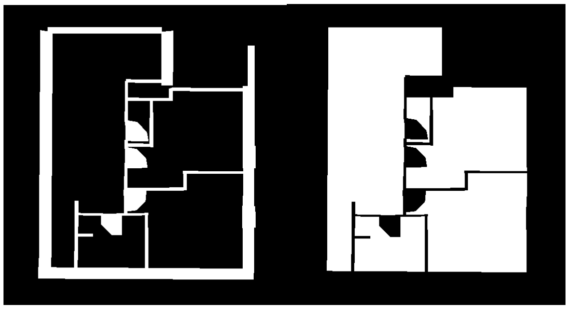


Figure 5. Example of intermediate outputs in the merge-and-split algorithm. *Left*: the sum of the outputs from the FCN and R-CNN networks after morphological closing. *Right*: The result of flood-filling the background of and then inverting the image at the left. Rooms are then easily separable.

4. Dataset and Evaluation Metric

The database for structural floor plan analysis (CVC-FP) is employed for the project [9].

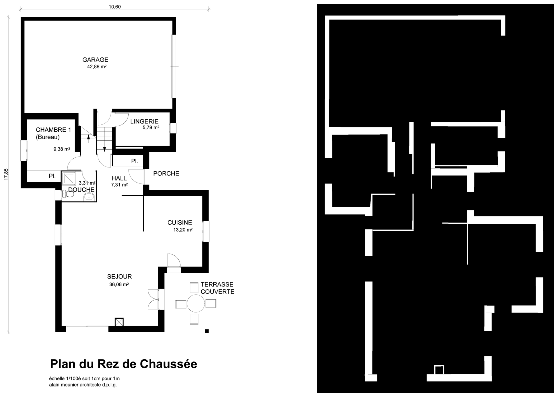


Figure 6. Training Set and Ground Truth Image(wall)[1]

The collection consists of 122 scanned floor plan documents divided in 4 different subsets regarding their origin and style. It contains documents of different qualities, resolutions, and modeling styles, which is suitable to test the robustness of the analysis techniques.

The dataset is fully ground-truthed in *.svg format for the structural symbols: rooms, walls, doors, windows, parking doors, and room separations.

The group converts .svg format ground truth images to *.png format so these can be feed into our deep neural networks.

Table 1. Dataset summary

Training	Test
100/122	22/100

Metric:

The project has three parts: wall segmentation, door/window detection, OCR. There is no metric for the system-level output, but we will maximize the performance of each part with the intent of finding a global optimum just like the “Greedy Algorithm”.

Intersection Over Union (Jaccard index) is employed for evaluating the performance of each part. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.

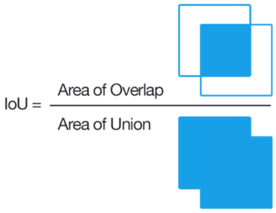


Figure 7. Illustration of the Jaccard index metric

For wall segmentation, IoU=area of overlap of ground truth wall area and predicted wall area / area of union of ground truth wall area and predicted wall area. For door/window detection, IoU=area of overlap of ground truth bounding box and predicted bounding box / area of union of ground truth bounding box and predicted bounding box.

5. Results

5.1 FCN for wall segmentation

The groups employed FCN-32s model proposed by J. Long [7] and used VGG16 pretrained model [8] to initialize the deep network and then utilized preprocessed CVC-FP dataset for fine tuning the model.

The outputs are shown in Figure 8 below. The group trained two different models with the same parameters but different input.

As the VGG 16 model takes 3-channel RGB images as input, so the 1-channel grayscale images are converted to 3-channel RGB images The input images are centralized(Mean-subtraction) for faster and more stable learning. The FCN parameters were updated with Adam optimizer. The learning rate was set to 1e-5. The first model was trained with the white background input and the second model was trained with the black background, which is inverted from the previous inputs. The black background inputs leaded to a better performance.

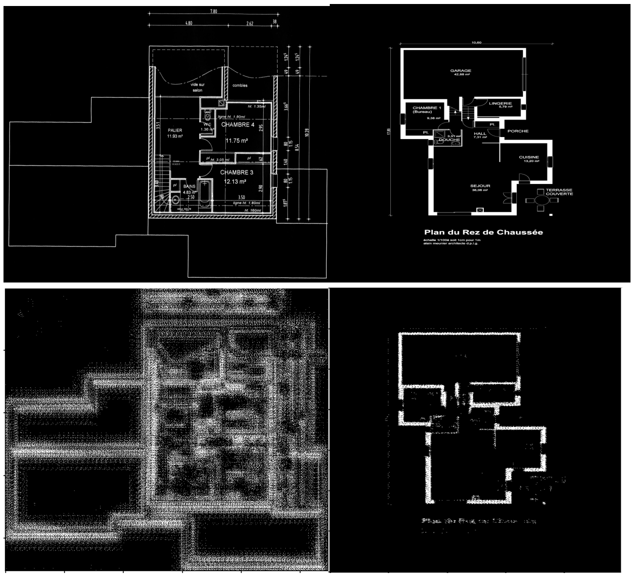


Figure 8. FCN inputs and outputs

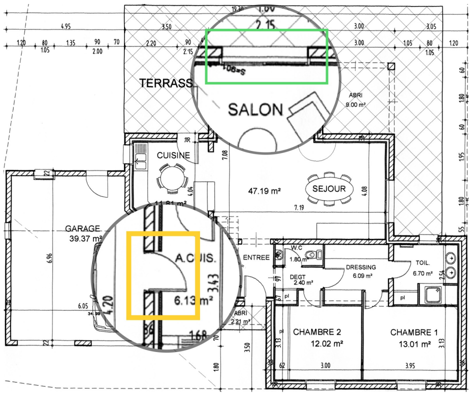
Left-top and Right-top shows the preprocessed floor plan images, Left-bottom shows the output before tuning parameters, Right-bottom show the output after tuning parameters.

State-of-Art Intersection Over Union	Our Test Intersection Over Union
89.7% [1]	60.5%

The performance is not perfect and still can be improved. In the future, the group will try to improve the performance of FCN by employing FCN32, FCN16, FCN8, FCN4, FCN2 deep networks. In sequential training each model is initialized with the parameters of the previous one.

5.2 Faster RCNN for door/window detection

In order to train the region proposal network, we initially make a training dataset which contains anchors we get from previous process and the groundtruth boxes. We label the anchors into “door”/”window”/ ”sink”/”toilet”/”bathtubs” in the image which are objects to be detected later. After obtaining anchors, these anchors go through RoI pooling layer and then form the classifier to do further classification.



From the above image, the doors and windows are predicted and labeled in test dataset through the region proposal network. As the network is very deep and dataset is quite large. The model has not been trained out yet.

For further improvement, we can combine the anchors of “sink” or “bathtub” to help predict room type such as toilets and chambers more precisely.

5.3 CNN with exhaustive search for door localization

Table 1. Architecture of the most effective door-localization CNN

Layer	Filter Count	Kernel Size	Stride Length
Con with ReLU 1	32	(3,3)	(2,2)
Max pool 1	N/A	(2,2)	(2,2)
Con with ReLU 2	64	(3,3)	(1,1)
Max pool 2	N/A	(2,2)	(2,2)
Con with ReLU 3	128	(3,3)	(1,1)
Max pool 3	N/A	(2,2)	(2,2)

For the CNN architecture (shown above in Table 1) that exhibited the highest test accuracy for doorway classification, Figure 9 shows the training and testing accuracy versus training epoch; for a 2.67 GHz Intel Core i7 processor with 16 GB of RAM and a Radeon Pro 560X 4096 MB graphics card, the network took about 30 seconds per epoch to train on all 13,500 examples in the training set. As shown in the figure, the network’s classification accuracy for the testing dataset plateaued around 95% after 130 epochs.

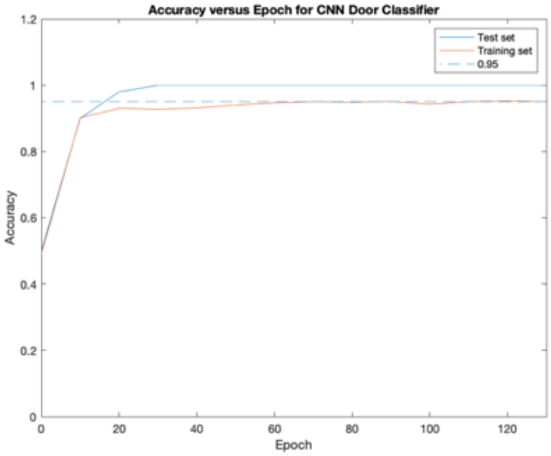


Figure 9. Training and testing accuracy versus epoch for the doorway classification CNN

Despite the high performance of the binary classifier and the relatively large size of the dataset, the exhaustive-search algorithm did not exhibit as strong intersection-over-union performance. An example output from the exhaustive-search algorithm is shown alongside the ground-truth image in Figure 10. Because the exhaustive search algorithm involves repeated application of the binary classifier, the algorithm requires about 2 minutes to classify a given input, meaning it is not suitable for real-time classification on a standard laptop—in contrast to methods like R-CNN as the literature has shown.

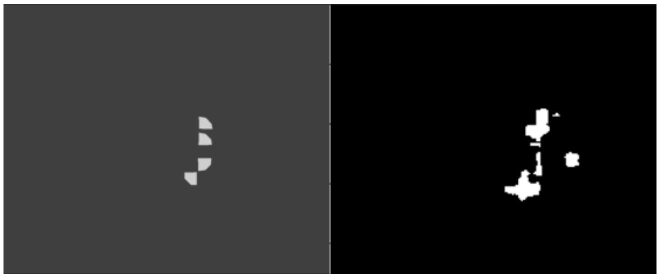


Figure 10. Ground-truth image (left) and predicted doorways (right)

In the future, we would like to train deeper networks with skip connections; both of these modifications may lead to greater classification accuracy and thus greater door localization. Finally, we would like to explore different ways of combining the output from the CNN classifier to generate the doorway localization map.

5.4 Optical character recognition

OCR with the Google Cloud Vision API for extracting room names from the floor plan images in .png format has been implemented. The output of the OCR will be like the text shown below.

```
CHAMBRE 3
10.02 m2
CHAMBRE 2
10.20 m2
```

(*CHAMBRE means bedroom in French)

In the future, the group will attempt to introduce Google translation API so the floor plan in any language can be proceed and translated into English automatically.

5.5 Merging and Splitting

Figure 11 shows output from the merge-and-split algorithm implemented in flood_fill.m, previously discussed in the Approach. Once the floor-image mask (top left of Figure 11) is generated as described in the Approach, our algorithm flood-fills each white region of the floor image mask with a different color in order to then trivially separate the room masks by color. The ratio of the number of white pixels in a given room mask to the number of white pixels in the corresponding floor mask immediately gives the percent area of the given room relative to the area of the apartment unit.

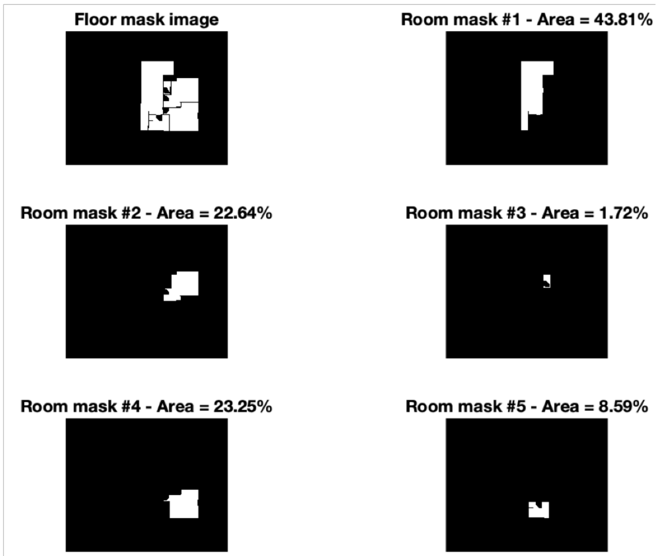

Floor mask image
Room mask #1 - Area = 43.81%
Room mask #2 - Area = 22.64%
Room mask #3 - Area = 1.72%
Room mask #4 - Area = 23.25%
Room mask #5 - Area = 8.59%

Figure 11. Final result of the merge-and-split algorithm

6. Timeline and Roles

Task	Deadline	Lead
Image Processing	Nov 7	Qinjin
OCR for room labeling	Nov 11	Peter
R-CNN for door/window detection	Nov 23	Rongyu
FCN for wall segmentation	Nov 23	Qinjin
Merge FCN, R-CNN and OCR to segment rooms	Nov 30	Robert
CNN for door/window detection	Dec 3	Robert

References

1) S. Dodge et. al. Parsing Floor Plan Images, MVA 2017

2) S.Ahmed et. al. Automatic room detection and room labeling from architectural floor plans. Int. Workshop on Document Analysis Systems, pp 339-343, 2012)

3) De. Las Heras, CVC-FP and SGT: a new database for structural floor plan analysis and its groundtrything tool, IJDAR 2015.

4) S. Ren et.al. Faster R-CNN: Towards real-time object detection with region proposal networks. NIPS 2015

5) D. Vargas, Wall Extraction and Room Detection for Multi-Unit Architectural Floor Plans.

6) Ahmed et. al Text/graphics segmentation in Architectural Floor Plans

7) J. Long, Fully convolutional network for semantic Segmentation.

8) K. Simonyan, Very Deep Convolutional Networks for Large-Scale Visual Recognition

9) L. Heras, Database for structural floor plan analysis http://dag.cvc.uab.es/resources/floorplans/