Group Project: Construction design recognition

Minghe Yang, Qiyu Xiao

January 3, 2022

Abstract

Labeling room blocks with different functionality on a building blueprint is a heavy task that is still mostly done manually by human nowadays. In this project, we explore the possibility of applying computer vision technique to this task. We use Detectron2[Det], an implement of Mask R-CNN[HGDG17], with different pre-trained model for performance comparison. High precision of room recognition is achieved but low room functionality recognition is acquired due to the limit of our dataset and the missing of optical character recognition(OCR) applications. The code of this project can be found at https://github.com/qyxiao/ComputerVision_final.

1 Introduction

Construction drawing is the general term used for drawings that form part of the production information that is incorporated into tender documentation and then the contract documents for the construction works.

During the process of making an operational construction drawing, different departments worked together to come up with the final design. Architecture apartment needs to come up with fancy designs, and the structure department needs to build a model to test if this design is practical or not. Time and efforts are consumed during this process, and sometimes the drawing is too complicated for them to spot every structure unit.

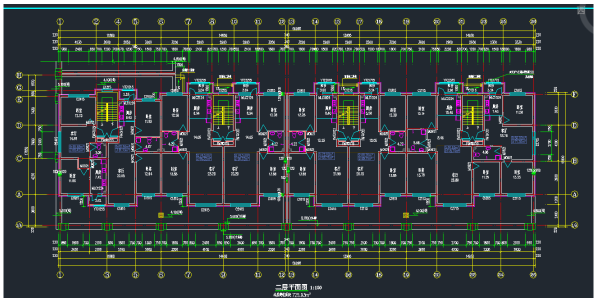

Figure 1: Construction drawing for a Residential building.

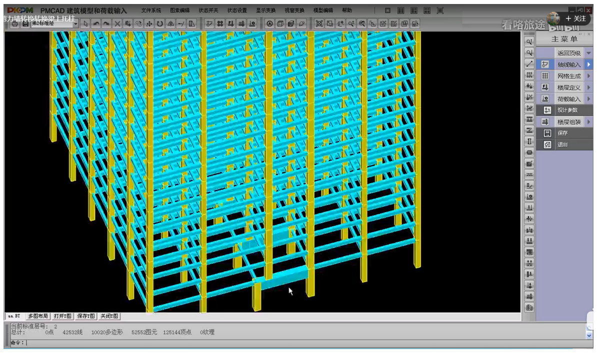

Figure 2: Modeling sample and strength detection.

For instance, if structure department came up with feedback that the downright part of this building is fragile and Vulnerable to potential earthquakes, the architecture department will come up with a second design which takes focus on that corner (Such as adding several pillars and beams enhance the strength), so the room layout will be different, and then structure apartment makes the second modeling based on this, 3rd... until everything's qualified.

The problem is, if a structure becomes to big, sometimes it could be hard for designers to spot every changes, though models are made to spot differences during comparison part, it takes a lot of time to rebuild the module and send the feedback. So, what our purpose is using computer vision to help with this. To Spot every room layout and compare them with the previous ones in order to help structure unit not to miss every unit. We used Mask RCNN NetWorks to help with the classification.

Since there's no current mature dataset for construction drawings, we have to build one for our own. During this report I'll first introduce how the dataset is created, then the classification and network structures.

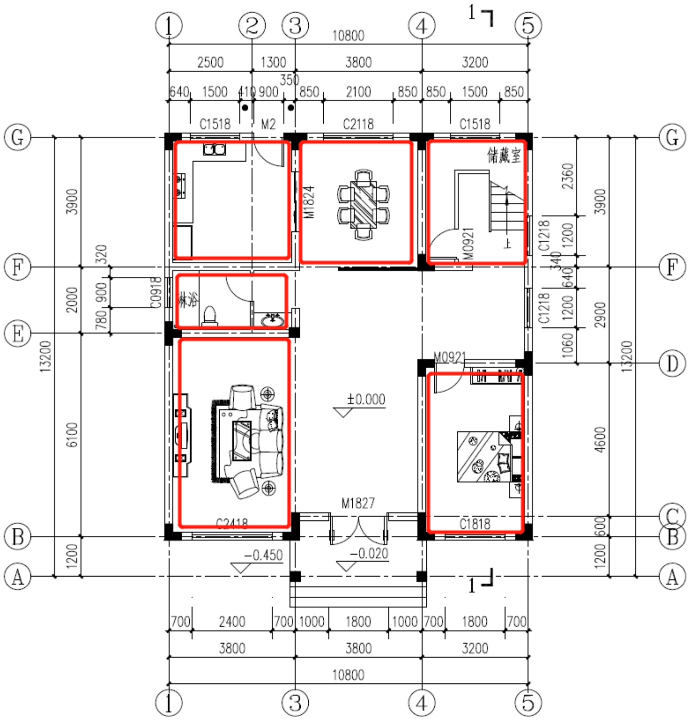

Figure 3: Example of a classification layout.

2 Dataset

2.1 Dataset Description

Since the design company usually signs a confidentiality agreement with first party, and the finished drawings are too complicated for training, we finally used blueprints of small villas in the country as our dataset due to their relatively small size and clear layout. Original CAD files from public school works are collected and transfer into readable png files. Labeling process uses Labelimg software to divide all rooms into 10 categories: bathroom, stairs, bedroom, living room, dining room, balcony, kitchen, garage, laundry, storage room divide by walls. The left top and right bottom corners are used to label the bounding box for each room. It would be ideal if we can also have labeled for the graphic representations or tokens of the room's functionality. However, except for 'bedroom' and 'stairs', other types of room don't have a uniform token, as shown in figure 4. And even for 'bedroom' and 'stairs', it is too much work to label the exact segmentation of their tokens. Thus in this project, we don't really distinguish the bounding box and the exact segmentation of the instance. Efforts are made during labeling and we finally got over 250 images for classification. We keep 20 as our test set and the rest as our training set. It's worth noticing that our sample number is small compared while the graphic representation of rooms with different functionality is fairly diversified. This problem can not be solved by simple data augmentation and imposes a large constrain on the task we expect the model to be able to handle, as will be further discussed in the next section.

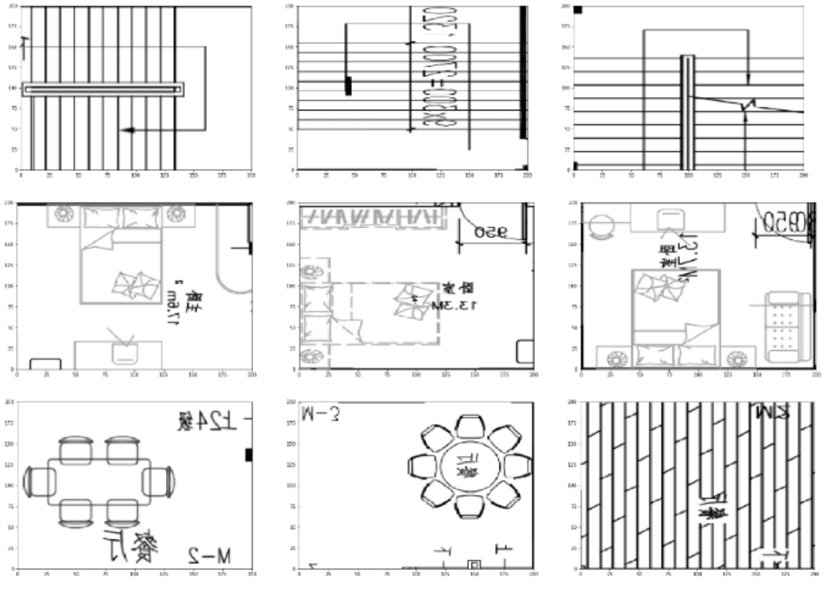

Figure 4: Examples of representations of 'stairs'(top), 'bedroom'(middle) and 'dinning room'(bottom) in blueprints. The latter apparently has more diversified representations than the other two.

2.2 Dataset Prepossessing

The original blueprint image has a size of 3508*2481 pixels, exceeding the capability of many devices to train and wasting computer power with a lot of irrelevant information to our task, such as blank spaces, dimensional lines and tables. To mitigate this problem, we crop the image to only keep information between (smallest label coordinate − offset) and (largest label coordinate + offset) for both dimensions, using (largest label coordinate - smallest label coordinate) $*$ 10% as the offset. We then resize the image with cubic interpolation to have same size of 512$*$512.

What's more, since an ideal model for our task should not be sensitive to the a rotation of the room or blueprints, we use rotation to augment our dataset. We rotate each figure in the training set by 90°/180°/270° and change the label coordinates accordingly. This way, our training set has around 1000 samples.

3 Network Structures

3.1 Mask R-CNN

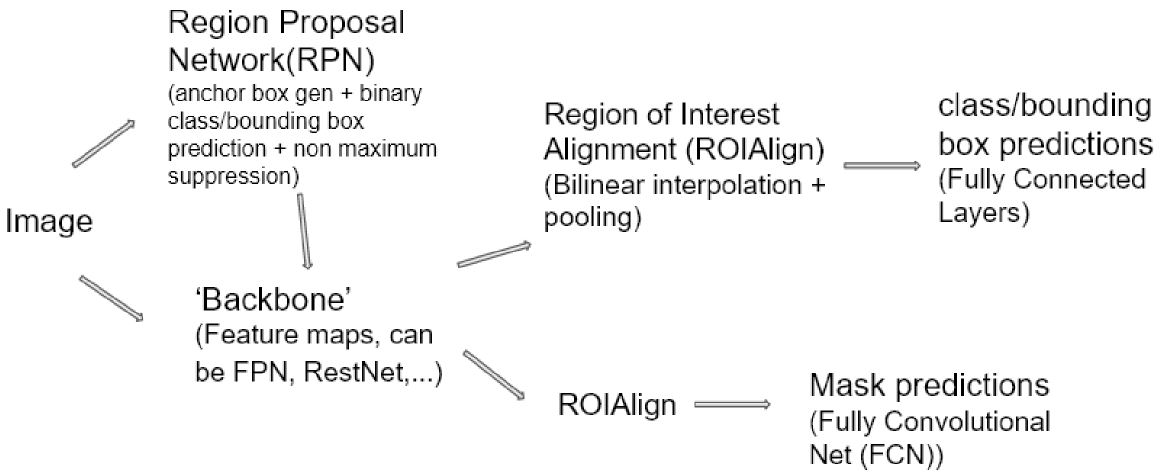
Image
Region Proposal Network(RPN)
(anchor box gen + binary class/bounding box prediction + non maximum suppression)
'Backbone'
(Feature maps, can be FPN, RestNet,...)
Region of Interest Alignment (ROIAlign)
(Bilinear interpolation + pooling)
class/bounding box predictions
(Fully Connected Layers)
ROIAlign
Mask predictions
(Fully Convolutional Net (FCN))

Figure 5: Structure of Detectron2 (Mask R-CNN) model.

Mask R-CNN is an extension of Faster R-CNN[RHGS15], which proceeds R-CNN[GDDM14] and Fast R-CNN[Gir15]. Its implementation in Detectron2 is shown in figure 5. First the image input is transformed through a 'backbone' layers and turn into feature maps. The 'backbone' here can use a modern CNN model like Feature Pyramid Network (FPN)[LDG+17], ResNet[HZRS16] or even combinations of these models with either pre-trained parameters or random initialization that's trained with the rest of model. On the other hand, there's a layer of Region Proposal Network, which proposes potential regions of interest for further investigation. This step tries to use CNNs to narrow down the number of bounding boxes for further investigations. This distinguishes two stage models like Mask R-CNN with one stage models like single shot multibox detector(SSD)[LAE+16] which always generate dense, fixed anchor boxes and detector each of them .

After this, the regions of interest will be delivered to the alignment layers with the features maps of the whole image. Instead of simply use proportional pooling to resize the feature maps in each proposed bounding box such that they suit for later layers, the Mask R-CNN creatively first use bilinear interpolation to get new sample points from the feature maps in these bounding boxes before applying pooling. Also notice that in contrast to previous models, Mask R-CNN has parallel networks for class/bounding box predictions and mask predictions for instance, instead of having mask predictions conditioned on class/bounding box predictions. At the end, the Mask R-CNN has fully connected layers for class/bounding box predictions and Fully Connected Layers[LSD15] for mask predictions.

3.2 FPN

Feature Pyramid Network (FPN) is a feature extractor designed for detecting objects in different scales with accuracy and speed in mind. It replaces the feature extractor of detectors like Faster R-CNN and generates multiple feature map layers (multi-scale feature maps) with better quality information than the regular feature pyramid for object detection.

FPN extracts feature maps and later feeds into a detector, says RPN, for object detection. RPN applies a sliding window over the feature maps to make predictions on the whether there's an object and the object boundary box at each location.

In the FPN framework, for each scale level (say P4), a 3 × 3 convolution filter is applied over the feature maps followed by separate 1 × 1 convolution for objectness predictions and boundary box regression. These 3 × 3 and 1 × 1 convolutional layers are called the RPN head. The same head is applied to all different scale levels of feature maps.

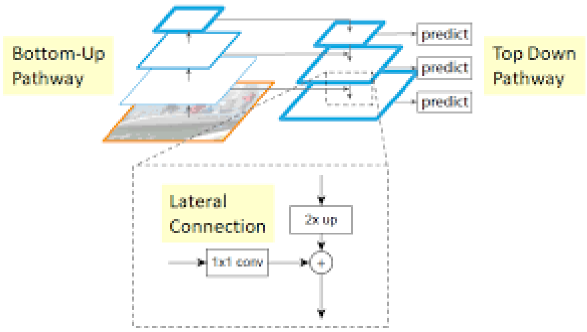

Figure 6: Structure of Feature Pyramid Network[FPN]

3.3 ResNet

Residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. In the context of residual neural networks, a non-residual network may be described as a plain network.

In this project we used Resnet50 as the backbone model and the structure is listed below.

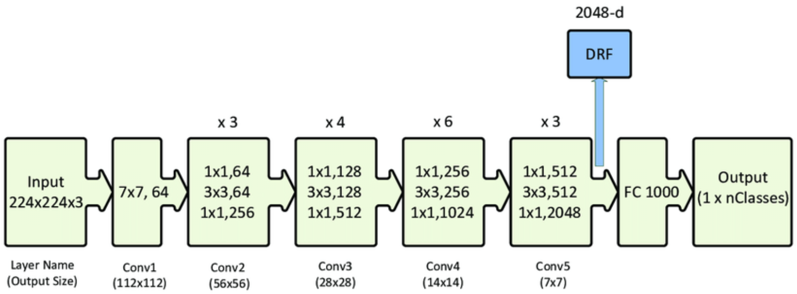

Figure 7: Resnet 50 model is used for convolution[Res]

4 Result and Discussion

As we briefly mentioned in Section 2, the limit of our dataset may impose a large constrain on the final performance of the model. Thus instead of directly applying the Detectron2 model for room detection and classification, we first attempt to only use this model for room detection. This should be a much simpler task since the basis of room detection should mostly be the existence of walls around a region instead of contents inside the region. After that, we also test the model on a task to distinguish 'stairs','bedroom' and 'other' types of rooms in the apartment. In this case the model need to look into the room. And we are only having three different categories because 'stairs' and 'bedroom' have relatively uniform representations while the rest of rooms do not. Following the practice in the original Mask R-CNN paper[HGDG17], we have used different pre-trained models mentioned in Section 3 for comparison of the performance and the result is shown below.

task/models	Res50+FPN	Res101+FPN	ResNeXt101+FPN
room detection	14.310	11.312	11.939
room classfication	49.285	45.755	46.269

Table 1: Average Precision for segmentation for different models and different tasks.

From Tabel 1, We can see that in contrast to the result in original Mask R-CNN paper, the simplest backbone model, Rest50 + FPN, works the best in both tasks. This may be because the data we have is too small and deeper networks not able to make all use of their capability. We can also notice that all models have higher precision in the seemingly more complicated task. To better understand this, we need to visualize the model predictions.

Let's visualize the room detections on a specific blueprint in figure 8. The left half shows result of a model only tries to predict if there is a room in that region or not while the right half shows a model that tries to find 'bedroom', 'stairs' and rooms of 'other' functionalities. On the left, We can see that overall the model does a decent job and have found all the rooms in the apartment. However a closer inspection reveals that even though this is the case, lines exists in the blueprint that are irrelevant to rooms separation may still mislead the model. The model wrongly categorizes blank regions surrounded by dimension lines in the figure as rooms, like the ones at the bottom left and middle right. And for the result on the right half, we see that even though the labelling is more accurate. The irrelevant lines no longer mislead the model because now the model also needs to look into the region and in this case will find there's nothing inside these 'rooms'. These model thus work better, but still far from perfect,=. For example it classified the bedroom in the middle top as 'other', even though it looks almost no different from any other bedrooms in these apartment to us humans.

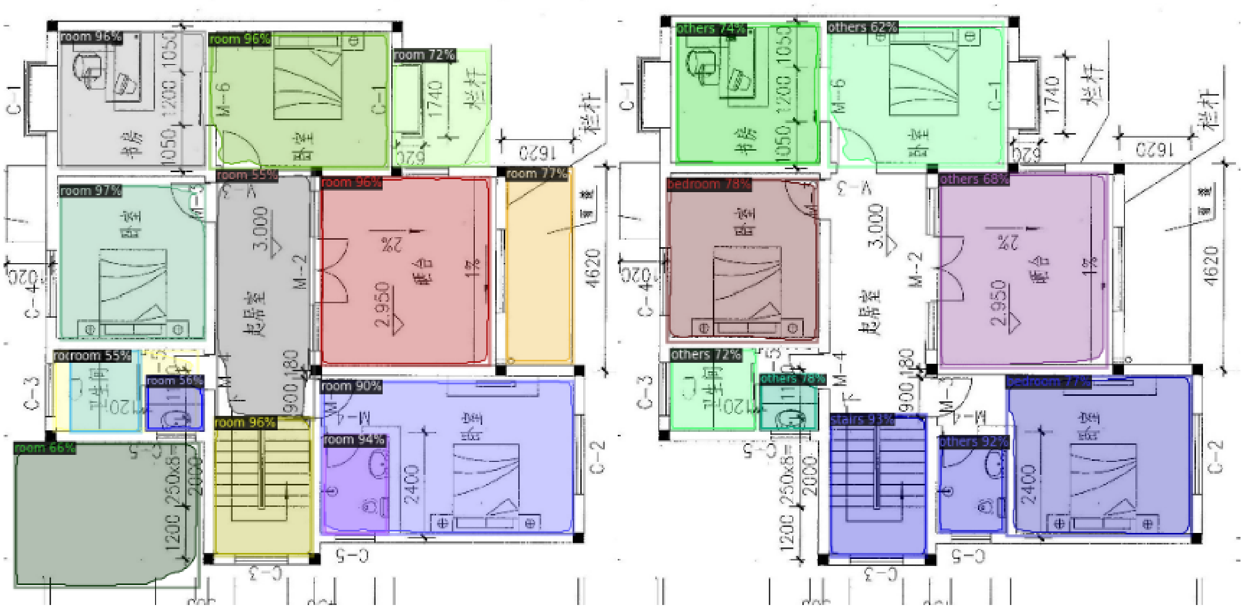

Figure 8: Model predictions on a specific blueprint. (left) Model only detects room existence and location. (right) Model tries to detect 'stairs', 'bedroom' and room of 'other' types.

For future improvement of this project, optical character recognition (OCR) is necessary as we can see that in many cases the functionality of room is not only represented by the graphic tokens but also text in the plot. And text when available tend to be much less ambiguous. And we also expect the performance of the model to improve if we can have dataset that has content within each rooms well labelled, instead of just bounding boxes being labelled. And in this project, we manually crop the image based on labels, but to achieve the goal of having fully automatic labeling we also need to train a model to crop image from original blueprints scan. Overall, we think this project has shed the light on the possibility of applying computer vision models on construction design recognition.

References

[Det] Facebook detectron2 github.

[FPN] Review: Fpn — feature pyramid network (object detection).

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[Gir15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[LAE$^+$16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[LDG$^+$17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[Res] Resnet-50 architecture.

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.