FCU ePaper

逢甲大學學生報告 ePaper

報告題名：

應用於建築平面圖之自動偵測睡房

Automatic Bedroom Detection on Architectural Floor Plan Images

作者：莊聖南
系級：電子工程學系
學號：D0616924
開課老師：梁詩婷老師
課程名稱：計算機視覺
開課系所：電子工程學系碩士班
開課學年：108 學年度 第一學期

中文摘要

本文提出了一種新穎的從建築平面圖數據庫中檢測房間的方法。因為現今只能透過人手的方式分析平面圖，這會非常浪費人力資源以及不合乎成本效益。所以基於本文提供的方法，在給予建築平面圖的情況下，本系統可以分析有關該設計圖的佈局細節。例如，可以使用圖像處理系統自動分析房間的數量，大小，類型，甚至房間的確切位置。 此外，它減少了分析時間並提高了業務工作效率。

本文利用傳統的圖像預處理方法和卷積神經網絡（CNN）來通過分割圖像與分類房間類型達到檢測睡室的效果。

最後能達到的分類睡室的最佳的準確率為 98.8%。這可以幫助客戶理解當前對住宅市場的趨勢和見解及該住宅的資訊，甚至能夠為房地產營銷上帶來優勢，可以更快速地幫助客戶做出決策。

關鍵字：卷積神經網絡、計算機視覺、深度學習、建築平面圖、圖像預處理

Abstract

This paper proposes a novel method to detect the room from an architectural floor plan database. Given a floor plan of a house, the details regarding the layout of the design can be determined, such as the number, size, type and even the exact location of the room can be interpreted automatically using the image processing methods. This can bring advantages to real estate marketing in identifying the current trends and insights to better assist the customers in making decisions. Besides, it reduces the analyzing time and increases the efficiency in business actions. This paper utilizes both the traditional image processing and convolutional neural networks (CNN) to detect the bedrooms by segmentation and classifications processes. The best bedroom classification accuracy achieved is 98.8%.

Keyword：floor plan · room · edge detection · CNN · classification

目次

內容

1 Introduction

When designing a building, the most indispensable tool for the architect is the floor plan, which provides safe building guidelines during the construction. A floor plan demonstrates the relationships between rooms, spaces, and other physical characteristics in a visual form. The floor plan usually specifies the basic layout dimensions (i.e., room size, height, length). Therefore, it is the most essential guide for the home buyer to consider purchasing the property [1]. Nevertheless, there is a lack of an automatic system to relate the architectural design to computer vision technology. Specifically, this automatic task is useful in assisting the buyers to quickly determine the number of bedrooms in each floor plan, classify the space according to the floor plan, analyze the amount of space in a floor plan, and determine the exact locations for each room.

Albeit the emergence of artificial intelligence, the existing literature studies in analyzing the architectural designs with this technology are manageably finite. For example, Bayer et al. [2] suggest a semi-automatic method to automatically generate the floor plans of specific buildings. Concretely, the Long Short-Term Memory (LSTM) [3] was utilized as predictive modeling to achieve this task. However, the trained model insufficiently describes the detailed contextual characteristics of the floor plan, such as the actual position of the basic building blocks (i.e., walls, doors, and windows).

On the other hand, Chen Liu et al [4] propose a novel convolutional neural network (CNN) architecture, namely FloorNet, to reconstruct the 3D floor plans by physically scanning the indoor spaces over a visual display. A triplebranch hybrid design is implemented to simultaneously process the 3D coordinate points, 2D floor plan and images captured, to form the final floor plan. De las Heras, Llu´ıs-Pere, et al. [5] design a system to first extract the main entities using statistical segmentation. Then the structural compositions of the building are identified using an image recognition technique. The authors claimed that the proposed algorithm is adaptable to any graphical representation, as it can extract the structural elements without prior knowledge of the modeling conventions of the plan.

Besides, [6] introduces a method to detect the elements of the walls and identify the key objects, as well as determine the characters from the floor plan images. The methodologies to accomplish this task are to adopt a fully convolutional network (FCN) model and an optical character recognition (OCR) technique. In brief, OCR is to retrieve meaningful room labeling. Although the size of the rooms can be obtained, the text might be failed to be recognized if the resolution of the image is low.

To improve the quality of the image, Sheraz et al. [7] suggest to segment the wall before performing the text recognition. The goal is to identify the text/ graphic from the floor plan. A promising result of recall of 99% is exhibited when evaluating on ~90 floor plans. On another note, they [8] point out that the wall thickness (especially thin wall) greatly affect the wall detection. The following year, the same research group [9] presents an automatic system to analyze and label the architectural floor plans by structural and semantic information. This paper extends the previous paper by dividing the rooms into several sub-partitions,if several rooms share the same room.

Recently, Goyal et al. [10] propose SUGAMAN (Supervised and Unified framework using Grammar and Annotation Model for Access and Navigation) to briefly describe the indoor environment in natural language from the building floor plan images. They represent the room features by adopting a local orientation and frequency descriptor (LOFD). Then, a single-layer neural network with 10 neurons is employed to learn the room annotations for room classification. To examine the effectiveness of the proposed algorithm, experiments are conducted on a dataset with more than 1000 image samples. Results demonstrated that the proposed method outperformed the stateof-the-arts by attaining higher classification accuracy when identifying the decor items (i.e., bed, chair, table, sofa, etc.).

Thus far, there are limited publicly available datasets that contain the architectural floor plan images. For instance, these four datasets: CVC-FP [11], SESYD [12], Robin [13], and Rent3D [14] usually served as the experimental data in the academic studies. The detailed information of these datasets is shown in Table 1 and the sample images are illustrated in Figure 1. It is observed that the databases are largely limited on their own. Concisely, the floor plan images may vary in different aspects: (1) building types (i.e., theater, school, house, museum, etc.); (2) multidimensional images (i.e., 2D and 3D); (3) representation types (i.e., sketches and computer-aided design), and; (4) furniture layouts (i.e., walls, windows, doors, sofa, stairs, etc.). Particularly, Rent3D and CVC-FP are the scanned images. The contents are mostly in text, rather than displaying the furniture icon. On the other hand, Robin and SESYD comprised of the computer-generated floor plan with lesser image noise, compared to the other two datasets. However, the wall thickness of Automatic Bedroom Detection on Architectural Floor Plan Images 3 SESYD is relatively thin. Therefore, it may lead to some errors during the wall segmentation process. Based on the pros and cons of the datasets discussed above, only Robin dataset is considered as the experimental data in this paper.

To the best of our knowledge, this is the first attempt to comprehensively address and analyze the details of the rooms from the floor plans. In this paper, a benchmark framework is provided to automatically determine the location and the number of bedrooms from a floor plan.

The main contributions of this paper are listed as follows:

1. Application of a series of pre-processing techniques to improve the image quality, such as noise removal, wall thickness adjustment and image scaling.

2. Adoption of several well-known pre-trained neural networks to learn the characteristics of bedroom features for bedroom classification.

3. Comprehensive experimentation on the dataset to verify the effectiveness of the algorithms evaluated. Both the qualitative and quantitative results are presented.

The remaining of the paper is organized as follows: Section 2 describes the complete framework in detail, including the pre-processing method, configuration of experiment settings and performance metrics for result validation. Section 3 reports and analyzes the experimental results. Finally, conclusions are drawn in Section 4.

2 Proposed Method

The proposed algorithm is targeted to determine the existence of the bedroom and their respective location as well as to compute the number of the bedroom from an architectural floor plan image. Figure 2 depicts the conceptual framework of the proposed method. In brief, it incorporates four primary stages: wall extraction, wall thickening and door gap closing, room partitioning and decor item retrieval, bedroom classification. The details of each step are discussed in this section by providing greater detail in terms of their respective mathematical derivations and pseudocodes. Note that, the dataset involved in the experiment is Robin because the decor items shown on the images are clear and do not contain any text information.

2.1 Wall extraction

The first step is to acquire the wall lines whilst removing the decor items. Prior to that, all the images are converted from RGB colorspace to grayscale. Specifically, Otsu's algorithm is employed to perform the thresholding technique to enhance image contrast. It is noticed that the decor items are appearing in thinner lines compared to the walls. Therefore, a simple morphological image processing method, viz, closing, is adopted to remove the objects with thin lines. The closing operation (1) is defined as:

$$A \cdot B = (A \oplus B) \ominus B$$

By doing so, the decor items can be eliminated, in the meantime, preserving the wall lines, as illustrated in Figure 2 (b). Moreover, Figure 3 (a)-(c) show the image outputs generated in this stage.

2.2 Wall thickening and door gap closing

As there are noticeable door gaps after performing the wall extraction in the previous step (i.e., Figure 2 (b)), the gaps are filled by thickening the walls to facilitate the gaps connection. First, the dilation morphological process is performed to improve the visual appearance of the wall lines. The dilation operation is expressed as:

$$A \oplus B = \bigcup_{b \in B} A_b$$

Next, the pairs of the door gaps coordinates for the linkage are identified. It can be achieved by applying Hough transform [15] and FAST algorithm for edge detection. Subsequently, the width and the exact position of the door gaps are obtained.

Fig. 1: Sample images from the four datasets: (a) CVC-FP; (b) SESYD; (c) Robin, and; (d) Rent3D;

The algorithm details to acquire the wall lines information is shown in Algorithm 1. In brief, there is three inputs required to link the door gaps, viz, the position, width and the total number of the door gaps. Line 1 and 2 in Algorithm 1 scan all the pixels in the image. Line 3 compares two coordinates to decide if they are the pair of the door gap, depending on the pre-defined width parameter. Lastly, line 4 links the pair of door gap coordinates.

Table 1: Detailed information of the four floor plan databases

		CVC-FP	SESYD	Robin	Rent3D
Presence of text		✓	✗	✗	✓
Total number of image		122	10	510	250
Item	Table	✗	✓	✓	✗
	Chair	✗	✓	✓	✗
	Sink	✓	✓	✓	✓
	Toilet bowl	✓	✓	✓	✓
	Bathtub	✓	✓	✓	✓
	Bed	✗	✓	✗	✗
	Cabinet	✗	✓	✗	✗
	Sofa	✗	✓	✓	✗
Fake/ Generated		✗	✓	✓	✗
Real/ Scan		✓	✗	✗	✓

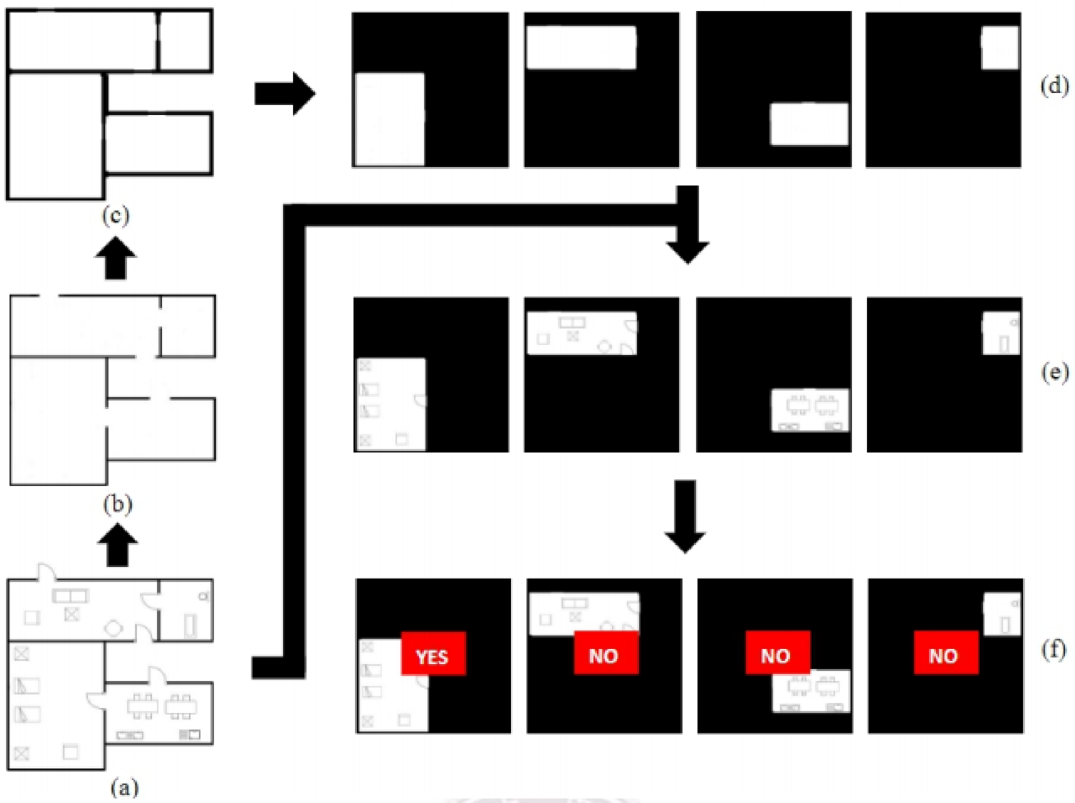

Fig. 2: Overall process flow illustration: (a) original dataset; (b) wall extraction; (c) wall line thickening and door gaps linking; (d) room partitioning; (e) decor items retrieval, and; (f) bedroom identification

Figure 2 (c) shows the output of the door gaps linking after increasing the thickness of the wall. Furthermore, Figure 3 (d)-(g) show the image outputs produced in this stage.

2.3 Room partition and decor items retrieval

This step is to split the multiple rooms from each floor plan into individual rooms. Since the floor plan images in Robin are generated by the computer, it is found that there is some absurdly small size of rooms in a few floor plan images. Therefore, if the detected room region occupies less than 300 pixels, it will be eliminated from the next processing step. Figure2 (d) illustrates the segmented rooms from a floor plan.

After identifying and filtering the rooms, the rooms are overlapped with the original floor plan image, such that the decor items appear in every single room. Figure 2 (e) illustrates the insertion of the original decor items on the respective partitioned rooms. The pseudocode to realize the room partition and decor items retrieval steps is shown in Algorithm 2. In brief, lines 1 to 27 detects the single rooms. Lines 28 to 33 determines if the room satisfies the room size requirement. Finally, line 34 stacks the decor items to the individual detected rooms.

2.4 Bedroom classification

This stage differentiates the rooms detected into bedroom/ non-bedroom categories. The Convolutional Neural Network (CNN) is employed as both the feature extractor and classifier for the bedroom classification. Intuitively, the shape, characteristics, patterns of a bed are the key features to decide if the image is a bedroom/ non-bedroom. Thus, CNN architectures are expected to learn the features of the bed in order to make the correct predictions.

Several pre-trained neural networks (i.e., AlexNet [16], ResNet [17] SqueezeNet [18] and GoogLeNet [19]) are utilized with slight modification. Concretely, these network architectures comprised of five types of operation: convolution, ReLU, pooling, fully connected and dropout. The bedroom images are first standardized to a certain size (i.e., $\aleph \times \aleph$).

1. Convolution and ReLU: The image performs a dot product between a kernel/ weight and the local regions of the image. This step can achieve blurring, sharpening, edge detection, noise reduction effect. ReLU is an element-wise activation function and is usually applied as the thresholding technique to eliminate the neurons that are playing a vital role in discriminating the input and is essentially meaningless. Each e_{ij} pixel in the image is defined as:

$$e_{ij}^{l} = \{f^{l}(x_{ij}^{l} + b^{l}) | i = 1, 2, ..., \aleph, j = 1, ..., \aleph\},$$
$$\text{where } x_{ij}^{l} = \Sigma_{a=0}^{m-1} \Sigma_{b=0}^{m-1} w_{ab}^{l} y_{(i+a)(j+b)}^{l-1}$$

l indicates the layer, x_{ij}^{l} is the pixel-value vector correspond to e_{ij} pixel, f^{l} is the ReLU activation function, w l is the coefficient vector and b^{l} is the bias determined by the feature map. Thus for an input x, the ReLu function can be indicated as:

$$f(x) = \max(0, x)$$

2. Pooling: To downsample the image along the spatial dimensions (i.e., width and height). This allows dimension reduction and enables the computation process to be less intensive. The $k\text{-}th$ unit of the image feature in the pooling layer is expressed as:

$$Pool_{\ k} = f(down(C) * W + b)$$

where W and b are the coefficient and bias, respectively. down(·) is a subsampling function:

$$down(C) = \max\{C_{s,l} | s \in Z+, l \in Z + \leq m\}$$

where $C_{s,l}$ is the pixel value of C in e and m indicates the sampling size.

3. Fully connected: All the previous layer and the next layer of neurons are linked. It acts like a classifier based on the features from the previous layer.

4. Dropout: The neurons are randomly dropped out during the training phase. This can avoid overfitting phenomena and enhance the generalization of the neural network trained.

Figure 2 (f) shows the categorization of each partitioned room to bedroom or non-bedroom after adopting the CNN method.

Fig. 3: Output generated after performing each image transformation process: (a) original image; (b) pixel binarization; (c) wall extraction; (d) image dilation; (e) image resize to 400 × 400; (f) edge detection using FAST algorithm, and; (g) door gaps linking

3 Experiment Results and Discussion

3.1 Performance Metric

There are two evaluation metric to validate the performance of the proposed framework, viz, accuracy and F1-score:

$$Recall := \frac{TP}{TP + TN},$$

$$Precision := \frac{TP}{TP + FP},$$

$$F1 - score := 2 \times \frac{Precision \times Recall}{Precision + Recall},$$

$$Accuracy := \frac{TP + TN}{TP + FP + TN + FN}$$

where:
– TP (true positive): The model correctly classified the bedroom.
– TN (true negative): The outcome where the model correctly predicts that is not a bedroom.
– FN (false negative): The event where the model does not predict the bedroom correctly, while in fact, it is a bedroom.
– FP (false positive): The test result indicates that it is a bedroom, but it is not.

3.2 Classification Performance and Analysis

All the 510 floor plan images from the Robin dataset are used as the experimental data. The bedroom classification results using five different CNN architectures (i.e., AlexNet, GoogleNet, SqueezeNet, ResNet-101 and ResNet-50) are shown in Table 2. Note that the number of epoch for each CNN is set to the range of 10 to 100. Table 2 reports one of the highest classification performance obtained among the epoch range. It is observed that the highest accuracy exhibited is produced by GoogLeNet, in which the accuracy and F1-score obtained are 98.40% and 98.80%, respectively. From the table, it can be seen that although ResNets are the two largest architectures among all the CNN methods, the classification results are the lowest (i.e., Accuracy=81.40% and F1-score=88%). This implies that learning the bedroom features leads to an overfitting phenomenon when a huge CNN is employed. Nevertheless, this binary classification task with limited data samples has been demonstrated that the transfer learning technique is capable to train the discriminant features on small size data.

To provide further insights into the classification performance, a detail investigation regarding some correct and incorrect predicted bedroom images. Table 3 tabulates the results of ten floor plan samples. Among the samples, 8 cases produce the F1-score of 100%, whereas 2 cases generate 66.67%. Aside from the numerical results, the qualitative visualization is shown in Figure to provide further classification context clues. Figure 4 depicts the activations of GoogLeNet. This particular activation layer selected is allocated in the third quarter of the network. There are noticeable brighter intensity pixels when there is a bed at the respective position. Besides, the activations of ResNet for the layer at a similar location (i.e., third quarter of the network) are shown in Figure 5 for a fair comparison. Apparently, Figure 5 (a) denotes that the bottom left corner should have a bedroom, as this particular in the activation image has bright pixels. However, that location does not have any bedroom, which means that the bedroom features learned in this network are insufficiently precise.

Table 2: Performance % of the bedroom classification when utilizing five convolutional neural networks, in terms of Accuracy (*Acc*) and F1-score (*F1*)

Method	Layers	Epoch	Acc	F1
AlexNet	25	60	90.5	97.5
GoogLeNet	144	30	**98.4**	**98.8**
ResNet-50	177	60	81.8	88
ResNet-101	347	30	81.8	87.2
SqueezeNet	68	70	96.6	97.9

Table 3: Detailed analysis of ten floor plan samples with GoogLeNet

Image	TP	TN	FP	FN	F1-score (%)
Rob_001	2	1	0	0	100
Rob_002	1	1	0	1	66.67
Rob_003	1	1	0	1	66.67
Rob_004	2	1	0	0	100
Rob_005	2	1	0	0	100
Rob_006	2	1	0	0	100
Rob_007	2	1	0	0	100
Rob_008	2	1	0	0	100
Rob_009	3	1	0	0	100
Rob_010	2	1	0	0	100

Algorithm 2: Room Segmentation and Decor Items Retrieval

Input:
$A(i,j) \leftarrow$ coordinate point of an independent room image
$m \leftarrow$ x-axis image size of an independent room
$n \leftarrow$ y-axis image size of an independent room
Output:
$N \leftarrow$ Number of independent rooms
$v = 1$;
$T = 1$;
while $v \neq 0$; ▷ `v = 1 indicates that a new area has been found`
do
$v = 0$;
$b = 0$;
for $i = 1; i \leq n; i++$ **do**
for $j = 1; j \leq m; j++$ **do**
if $A(i,j) == 255$; ▷ `Pixel intensity of 255 indicates white`
then
$A(j,k) = T$;
$v = 1$;
break;
$Q = 1$;
while $Q \neq 0$; ▷ `Q = 1 indicates there may be missing points`
do
$Q = 0$;
for $i = 1; i \leq n; i++$ **do**
for $j = 1; j \leq m; j++$ **do**
if $A(i,j) == T$ **then**
Convert the pixel values to 255 in the top, bottom, left and right directions of points $A(i,j)$ into T
This process is terminated when it encounters 0.
if *a pixel is converted* **then**
$Q = 1$;
else if $A(i,j) == 255$ **then**
Determine whether there is T before 0 is encountered in the four directions of point $A(i,j)$.
$T = T + 1$
for $i = 1; i \leq T; i++$ **do**
if *there is i on the boundary or the total number of pixels with a value of i is less than 300* ; ▷ `These two cases are not considered as a room`
then
continue;
else
Convert all i to 0 and convert other values to 255.
Stack the decor items on the respective detected rooms.

4 Conclusion

This paper presents a novel framework to automatically identify the location and the number of bedrooms from the floor plan images. Some traditional and data-driven image processing techniques are applied. In brief, Otsu's thresholding and morphological operations are employed to pre-process the image. Then, the rooms are extracted using the Hough transform and FAST algorithm. Finally, some popular convolutional neural network architectures are utilized to determine if the detected room is the bedroom.

As future work, this framework can be extended to recognize other function rooms, such as the bathroom, dining room or living room, etc. Besides, the binary classification task and the limited data samples point towards the studies on the new design of the shallow neural network. On the other hand, it is also worth investigating the optical character recognition (OCR) technique on other types of floor plan images.

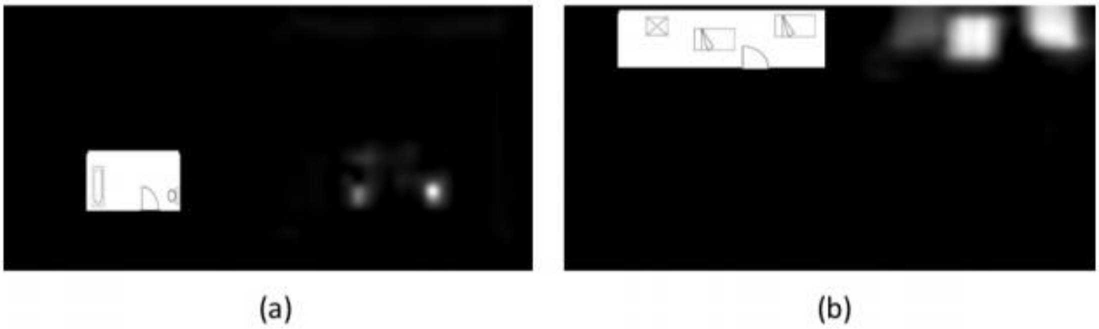

Fig. 4: Activations of GoogLeNet when the image: (a) does not contain a bed, and; (b) contains a bed;

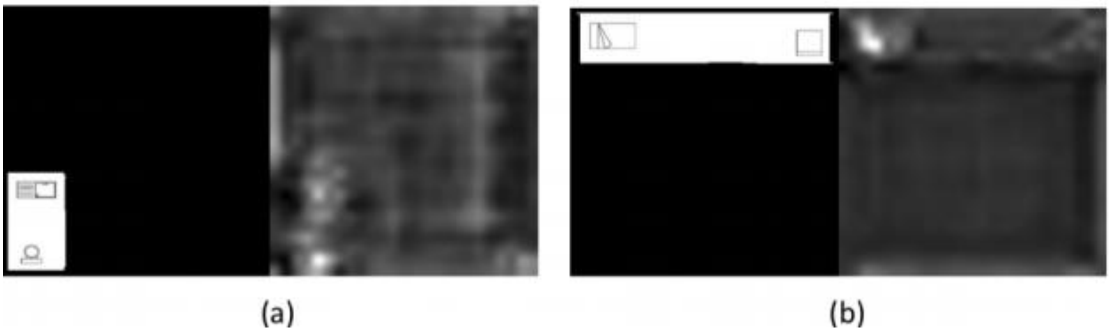

Fig. 5: Activations of ResNet when the image: (a) does not contain a bed, and; (b) contains a bed;

參考文獻

1. Williamsburg, URL https://mrwilliamsburg.com/why-are-floorplans-not-used-in-real-estate-marketing/
2. J. Bayer, S.S. Bukhari, A. Dengel, in International Conference on Pattern Recognition Applications and Methods (Springer, 2018), pp. 27–43
3. S. Hochreiter, J. Schmidhuber, Neural computation 9(8), 1735 (1997)
4. C. Liu, J. Wu, Y. Furukawa, in Proceedings of the European Conference on Computer Vision (ECCV) (2018), pp. 201–217
5. L.P. de las Heras, S. Ahmed, M. Liwicki, E. Valveny, G. S´anchez, International Journal on Document Analysis and Recognition (IJDAR) 17(3), 221 (2014)
6. S. Dodge, J. Xu, B. Stenger, in 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA) (IEEE, 2017), pp. 358–361
7. S. Ahmed, M. Weber, M. Liwicki, A. Dengel, in 2011 International Conference on Document Analysis and Recognition (IEEE, 2011)
8. S. Ahmed, M. Liwicki, M. Weber, A. Dengel, in 2011 International Conference on Document Analysis and Recognition (IEEE, 2011), pp. 864–869
9. S. Ahmed, M. Liwicki, M. Weber, A. Dengel, in 2012 10th IAPR International Workshop on Document Analysis Systems (IEEE, 2012), pp. 339–343
10. S. Goyal, S. Bhavsar, S. Patel, C. Chattopadhyay, G. Bhatnagar, IET Image Processing (2019)
11. L.P. de las Heras, O.R. Terrades, S. Robles, G. S´anchez, International Journal on Document Analysis and Recognition (IJDAR) 18(1), 15 (2015)
12. M. Delalandre, E. Valveny, and J.-Y. Ramel, "Recent con-tributions on the sesyd dataset for performance evaluationof symbol spotting systems."
13. D. Sharma, N. Gupta, C. Chattopadhyay, S. Mehta, in 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1 (IEEE, 2017), vol. 1, pp. 420–425
14. C. Liu, A.G. Schwing, K. Kundu, R. Urtasun, S. Fidler, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015), pp. 3413–3421
15. S. Mac´e, H. Locteau, E. Valveny, S. Tabbone, in Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (ACM, 2010), pp. 167–174
16. A. Krizhevsky, I. Sutskever, G.E. Hinton, in Advances in neural information processing systems (2012), pp. 1097–1105
17. K. He, X. Zhang, S. Ren, J. Sun, in Proceedings of the IEEE conference on computer vision and pattern recognition (2016), pp. 770–778

18. F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, arXiv preprint arXiv:1602.07360 (2016)
19. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, in Proceedings of the IEEE conference on computer vision and pattern recognition (2015), pp. 1–9

逢甲大學 忠勤誠篤