ST-RoomNet: Learning Room Layout Estimation From Single Image Through Unsupervised Spatial Transformations

Hatem Ibrahem, Ahmed Salem, Hyun-Soo Kang*
School of Information and Communication Engineering
Chungbuk National University, Cheongju, Korea
{hatem, ahmeddiefy, hskang}@cbnu.ac.kr

Abstract

Room layout estimation is an important task for the 3D reconstruction of indoor scenes and augmented reality applications. The layout of the room is usually estimated by predicting the keypoints of the room corners, room planer segmentation (floor, ceiling, right, left, and front walls), or line detection. In this paper, we propose a novel way to estimate the room layout from monocular RGB images using spatial transformation networks (STN). Since it is commonly known that the room has a cuboid layout, we train a convolutional neural network to predict unsupervised perspective transformation parameters that can transform a reference cuboid layout to the required room layout based on the deep features of the input room image. We show that the proposed method is simple and efficient in learning the room layout without the need to perform segmentation, line detection, or keypoint estimation. We test the proposed method on two challenging benchmarks; LSUN Room Layout and Hedau dataset showing that our method can achieve pixel accuracy errors of 5.24% on LSUN and 7.10% on Hedau at a speed of (10∼15 fps) outperforming the state-of-the-art methods in room layout estimation task. Paper's code is available at: https://github.com/lab231/ST-RoomNet/.

1. Introduction

The 3D construction of indoor scenes is an essential task in computer vision which requires a lot of information about the scene or multiple images of the scene. The room layout estimation is a primary task in the process of 3D construction or in modern augmented reality (AR) applications, in which the location information of the room walls, floor, and ceiling is extremely important. Traditional methods for 3D construction usually depend on multiple-view approaches, but recently, researchers pay more attention to

*Corresponding author

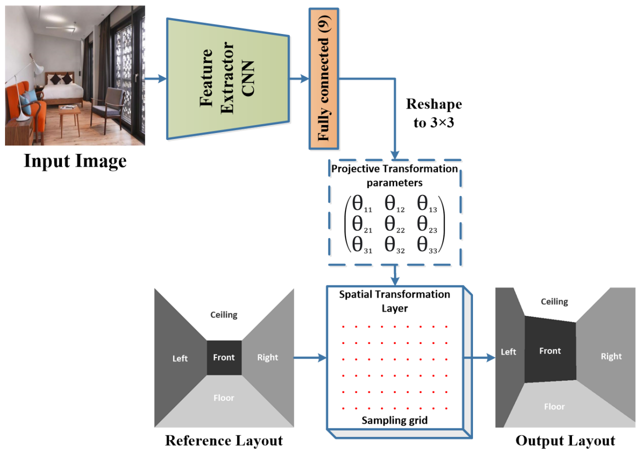
Input Image
Feature Extractor CNN
Fully connected (9)
Reshape to 3×3
Projective Transformation parameters
$\begin{pmatrix}\theta_{11} & \theta_{12} & \theta_{13}\\ \theta_{21} & \theta_{22} & \theta_{23}\\ \theta_{31} & \theta_{32} & \theta_{33}\end{pmatrix}$
Spatial Transformation Layer
Sampling grid
Ceiling
Left
Front
Right
Floor
Reference Layout
Output Layout

Figure 1. An overview of the proposed method for room layout estimation. A convolutional neural network is used to predict the persepective transformation parameters that are used to transform a reference layout scene to the input room's layout through a differentiable spatial transformation layer (STL).

the single-view geometry and how to exploit every piece of information in the image for the 3D perception of the scene. An indoor scene image includes much important semantic information which can be extracted through convolutional neural networks (CNN). That information includes low-level information about the boundary lines and corner points that exist in the scene and higher-level information such as what objects exist in the scenes and the alignment of those objects. The main objective of the room layout estimation is either to predict the segmentation mask for the main room planes (walls, floor, and ceiling) or to predict the keypoint coordinates of the corner of the room. The recent research on room layout estimation [1–9] depend on keypoints prediction approach or direct semantic segmentation of the different planes of the room, however, those approaches are inefficient since the room walls, floor, and ceiling are usually occluded by objects that make most of the wall plane not obvious, and in most cases, the corner points of the walls (keypoints) are occluded and not directly obvious. Some other research [10, 11] depend on line estimation approaches by segmentation and it also suffers from the occlusion of most of the line, which affects the final wall segmentation or keypoint estimation. In this paper, we propose a novel approach for room layout estimation using the commonly known spatial transformation network [12]. Spatial transformation networks are a special kind of CNN where the transformation layer can spatially transform the input image (or volume in 3D) depending on the localization features of the image. A CNN is employed to estimate deep representative features of the input room image in order to predict eight perspective transformation parameters following equation 1:

$$\begin{bmatrix} wx\prime \\ wy\prime \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

where x and y are the input pixel coordinates, $x\prime$, and $y\prime$ are the transformed pixels which can be obtained by normalizing the output vector by w. Noting that p_{33} equals 1 so the CNN model is designed to predict 8 parameters instead of 9. A spatial transformation layer [12] is then employed to transform the reference layout image into the required layout using grid-based 1-to-1 mapping. The proposed method can indirectly learn the transformation parameter for the reference layout based on end-to-end training that maps the input RGB image into the layout plane segmentation without any extra supervision. A detailed overview of the proposed method is shown in Figure 1. The contribution of this paper can be summarized as:

- We propose an end-to-end approach for the room layout estimation using spatial transformation networks.
- The proposed method simply predicts 8 parameters of transformation instead of the traditional complex semantic segmentation or keypoint estimation approaches.
- We propose two lightweight CNN architectures for room layout estimation that can process frames at relatively high speeds (15.1 and 10.0 fps).

The remaining of this paper is organized as follows: the related work (section 2), the proposed method (section 3) for the details of our method, datasets for training and testing (section 4), the experimental results that presents our experiments on room layout estimation (section 5), limitations and future work based on our method (section 6), and finally, the conclusion of the paper (section 7).

2. Related work

In the room layout estimation task, the classical methods [10, 11] aimed at modeling the structure of an environment using complete low-level features such as image edge features and superpixel features, then using hypotheses to generate the room layout, but those methods were limited in accuracy, in addition to the computational complexity of their implementations. The recent methods for room layout which are based on deep learning approaches either depend on walls, floor, and ceiling semantic segmentation approaches or keypoint estimation approaches. Long *et al.* [13] was the first to propose a fully convolutional network (FCN) for semantic segmentation which opened the way for the plane segmentation methods. Mallya and Lazebnik [1] used FCN to predict an edge map, and the layout ranking method is applied to the map to sample the vanishing points. Dasgupta *et al.* [2] directly used FCN to predict the semantic segmentation mask of the walls, ceiling, and floor. Then a hole-filling technique is used to refine the segmentation map followed by vanishing point estimation techniques based on the predicted segmentation to produce the layout. The plane segmentation-based methods are computationally expensive as they need many post-processing stages to get the final layout. This method of Dasgupta *et al.* [1], for example, takes 30 seconds to process a single frame. Zhang *et al.* [3] proposed a high-quality edge segmentation approach using an encoder-decoder architecture with a one-dimensional latent representation to aggregate the information of every region, and an adaptive sampling strategy is used to produce the layout. Ren *et al.* [4] proposed a Coarse-to-Fine Indoor Layout Estimation (CFILE) in which they combined the room contours segmentation and plane segmentation to produce robust layout results with surface smoothness and geometric constraints. Lin *et al.* [5] proposed a layout estimation method based on the combination of plane segmentation, line segmentation, and heatmap smoothing loss without the need for extra post-processing stages, and they could relatively improve the previous layout estimation results. Lee *et al.* [6] proposed RoomNet which is an encoder-decoder architecture for simultaneous segmentation of keypoints location and layout type prediction. The predicted layout type is used to reorder the estimated keypoints to ensure the correctness of the predicted layout. Hirzer *et al.* [7] used three segmentation hypotheses for the room layout estimation, they proposed to predict the locations of the corners in a room image and select one of the three hypotheses that is consistent with the layout estimated from the room corners. Zheng *et al.* [8] proposed a structural deep metric learning (SDML) to estimate the room layout by explicitly modeling the structural relations across different rooms. They trained a model to minimize the difference between the room image and its layout. Wang *et al.* [9] proposed Feature Pyramid Net-

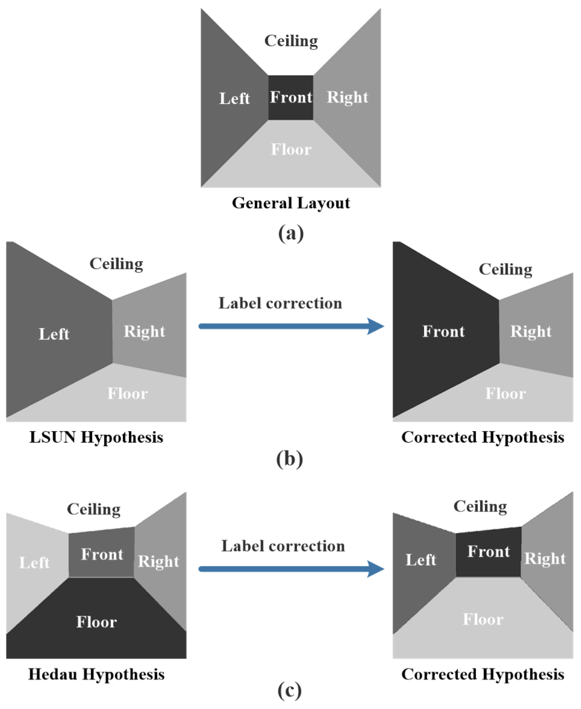

Figure 2. (a) The proposed general layout of the room. (b) An example of the label correction applied to the LSUN [15] layout hypotheses that neglect the real structure of the room. (c) An example of the label correction applied to Hedau [16] dataset.

works (FPN) [14] for room layout estimation by predicting coarse positions of the room keypoints preserving the keypoints order. Then moving the coordinates of each point to the nearest image boundary as a second refinement stage. Although the previously mentioned methods could achieve acceptable accuracies, most of them still suffer from complex implementation due to the extra post-processing and refinement stages. In our proposed method, we train an end-to-end CNN to directly predict the room layout without the need for extra post-processing stages as the model directly predicts the perspective (homography) transformation parameters by regression that can transform a reference room's layout to the required room's layout, in opposite to all the previously mentioned methods which either depend on semantic segmentation or keypoint estimation. Thanks to the use of a customized spatial transformation network, we could outperform the current method in room layout estimation, and we could perform the task at relatively high speed.

3. Proposed method

We model the room layout estimation problem as a general cuboid scene transformation through the spatial transformation networks. The details of the proposed method will be shown in the following subsections.

3.1. General hypothesis

In order to perform the room layout estimation task, we propose a reference cuboid representing the standard layout of any room as shown in Figure 2-(a), and since we do experiment our method on two common datasets (LSUN room layout [15] and Hedau [16]), we correct the plane labels of the room sides. In the case of the LSUN [15] dataset, the cases when there are only two walls in the room (in addition to the ceiling and floor) neglect the real structure of the room and label the two walls with left and right walls. Such labeling is confusing as the front wall is in the middle of the left and right walls. We correct such cases to follow the general hypothesis that we propose and the model can learn the relation between the reference layout and the required layout. An example of this case is shown in Figure 2-(b). In the case of Hedau [16] dataset, the original layout hypothesis follows our hypothesis but the labels of the room sides are different. We correct the labels of the room layouts in the original data to follow the labels of our proposal. An example of this case is shown in Figure 2-(c).

3.2. Modified spatial transformation networks

Spatial transformation networks (STN) [12] is a type of convolutional network that can manipulate the input image (or features) spatially, and can learn invariance to scale, translation, scale, and generic warping through a differentiable spatial transformer module. A grid is employed to be used for the features' spatial transformation and a bilinear interpolation is used to define the values of the transformed features. The original STN paper employed a CNN for feature extraction (i.e. localization) and then a fully connected layer is added to predict the six parameters of the affine transformation. The parameters are given to the spatial transformer layer to transform the input image and then a CNN classifier is added for the final prediction. In our proposed method, we follow a more advanced approach with some modifications. We used the perspective (homography) transformation layer instead of the simple affine layer in the original study as the transformation from the reference layout to the required layouts follow the homography transformation rules. Also in the training phase, we used bilinear interpolation, but in the inference phase, we used nearest neighbor interpolation to avoid the wrong labels in the edge pixels of the room sides. The general architecture of the proposed method is shown in Figure 3-(a).

3.3. Feature extractors

We employ two modern CNN architectures for the task of input image feature extraction. First, we employ ConvNext-Tiny [17] which is proven an efficient CNN architecture for image classification outperforming most of the modern computer vision models either convolutional-based or vision transformer-based [18] models. ConvNext-

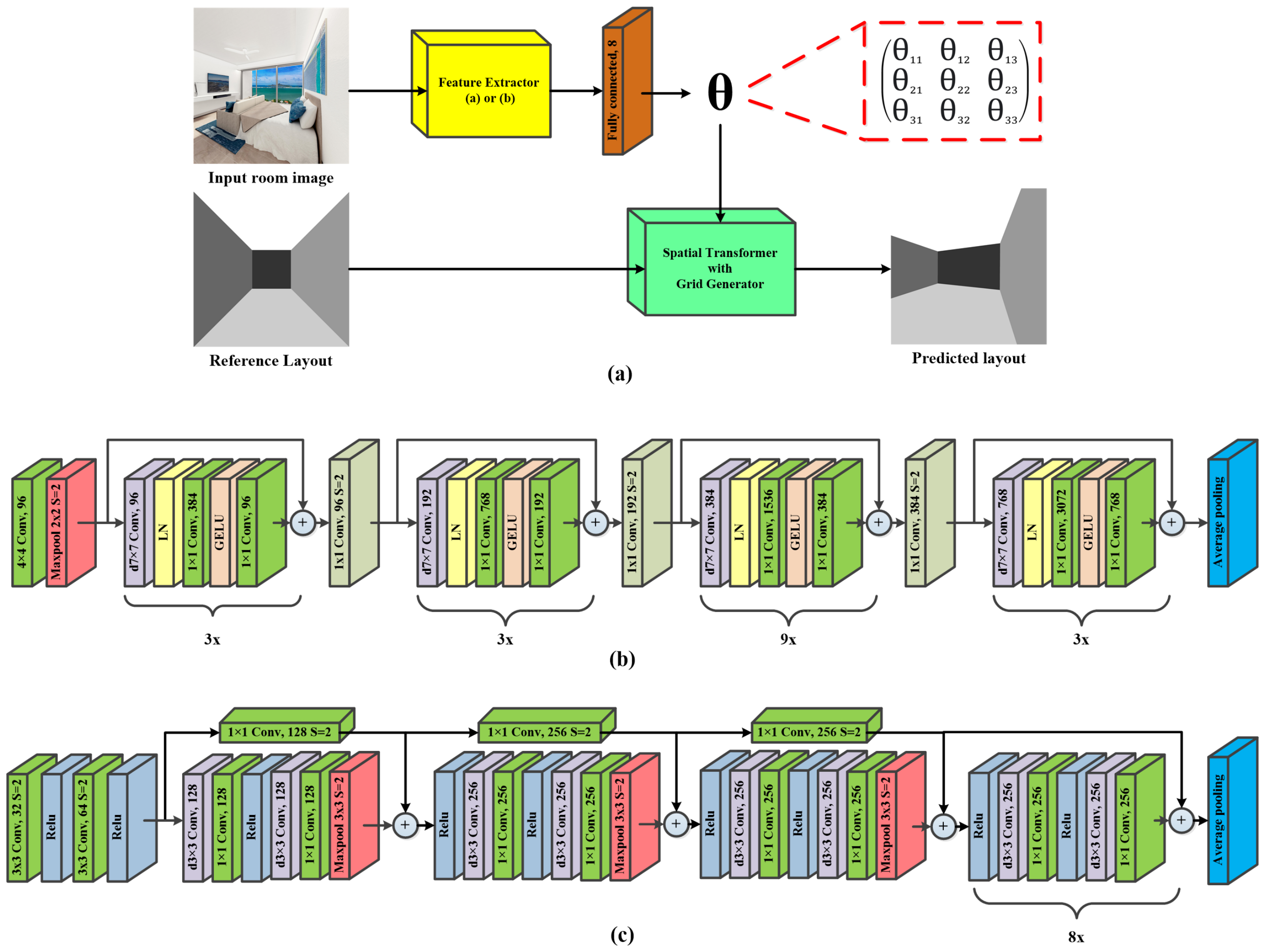

Figure 3. The proposed architecture of our method. (a) The general architecture of the proposed method consists of the feature extractor stage followed by a fully connected layer that predicts the transformation parameters needed for the perspective transformation of the reference layout. (a) Details of the ConvNext-Tiny feature extractor. (b) Details of the modified Xception (MXception) feature extractor.

Tiny architecture, shown in Figure 3-(b), depends mainly on image patch embedding using strided convolution layer which is proved to be beneficial for accuracy (originally inspired from the swin-transformer [19]) and large kernel size for convolutions. Also, the architecture consists of replicas of residual blocks with depthwise [20], pointwise [20] convolutions, and layer normalization [21] instead of the traditional batch normalization. The depthwise convolution is a special type of convolution where the convolution is applied for channel separately in the input features, and pointwise convolution is 1×1 convolution to project the input features to new feature count. The combination of depthwise and pointwise convolutions was first proposed by Challot [20] and proved to be efficient in learning representative features. The authors of ConvNext reported that using kernel size of 7×7 produces much more accuracy than smaller kernel sizes and even for larger kernels (9×9 or 11×11) in image classification, and the GELU activation showed better performance than RELU activation so we used the same kernel size and GELU activation through all the architecture as shown in Figure 3-(b). The second alternative architecture employed in our method is a modified version of Xception architecture (MXception) [22]. It is a reduced version of the original Xception [20] taking the output from the 13th Xception block and then adding average pooling to the final features. The architecture also consists of residual blocks of Xception where each Xception block consists of RELU, 3×3 depthwise conv, 1×1 conv, RELU, 3×3 depthwise conv, 1×1 conv, and a residual connection between the start of the block and the end of the block. In the early Xception blocks, this block has max-pooling at the end and 1×1 convolution in the residual connection. The modified Xception architecture is shown in Figure 3-(c). Although we tried to train different recent feature extractors, those ConvNext-

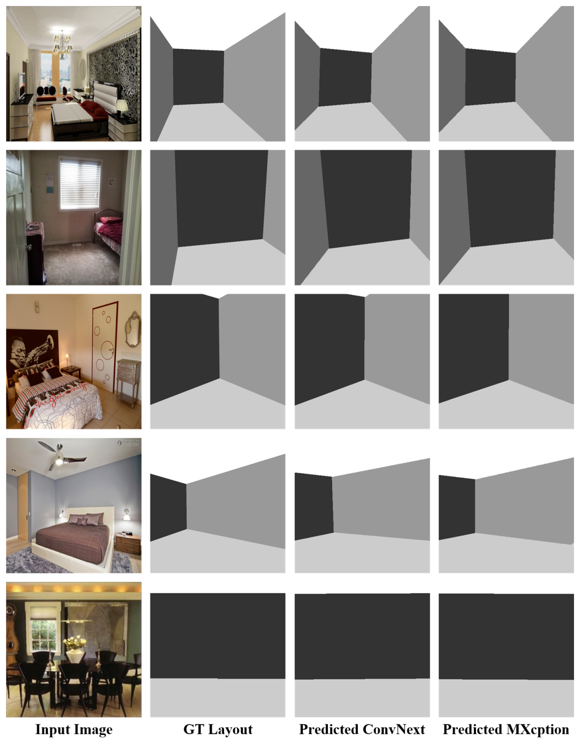

Figure 4. Qualitative results obtained by the proposed method on LSUN [15] using ConvNext and MXception feature extractors.

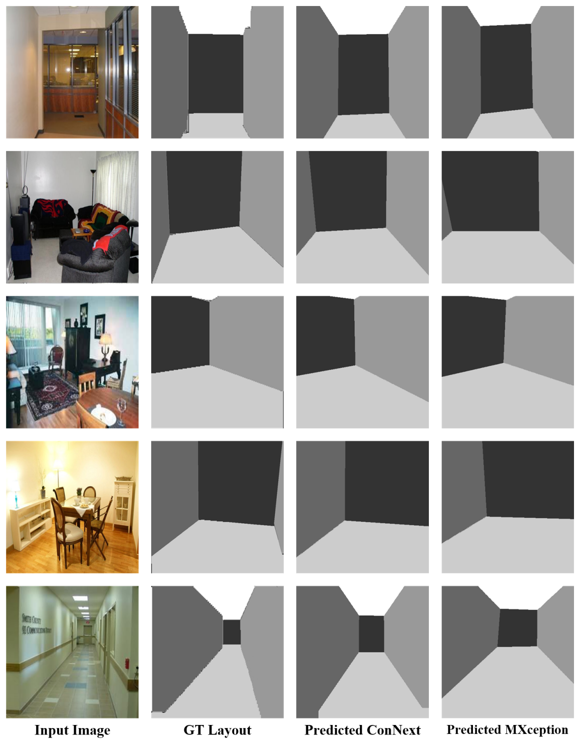

Figure 5. Qualitative results obtained by the proposed method on Hedau [16] using ConvNext and MXception feature extractors.

Tiny and MXception were the only two that could learn robust features for transformation as the trials with other feature extractors all failed which proves the efficiency of the depthwise and pointwise as they are the common modules between the two architectures.

3.4. Loss function

To train the proposed model, we employed Huber loss to regress the final layout image. Huber loss (which is either L1 or L2 loss depending on a threshold value) can be mathematically stated as :

$$L = \begin{cases} \frac{1}{2N}\sum_{n=1}^{N}(y-\tilde{y})^2, \text{if } |y-\tilde{y}|<t \\ \\ \frac{t}{N}\sum_{n=1}^{N}(|y-\tilde{y}|-\frac{1}{2}t), \text{otherwise} \end{cases} \tag{2}$$

where N represents the number of pixels, y is the ground truth pixel value and $\tilde{y}$ is the predicted pixel value, while the threshold value, t, is selected as 1 since empirically it speeds up the training process. We chose Huber loss over L1 and L2 loss since it is faster than either of them.

4. Datasets for training and testing

We train the proposed method on LSUN Room Layout Estimation [15] and Hedau [16] datasets. The LSUN Room Layout Estimation dataset consists of 4,000 training images, 394 validation images, and 1,000 testing images, while Hedau dataset consists of 209 training, 53 validation, and 105 test images. To expand the variety of scenes, we utilize random brightness during training by slightly altering the contrast of the input images and utilizing random horizontal flipping taking into consideration flipping the labels of the right and left walls. For evaluation, we evaluate the proposed method on the test sets of LSUN [15] and Hedau [16] dataset.

5. Experimental results

In this section, we report the experiments done using the proposed methods and the results obtained from each experiment on LSUN and Hedau datasets.

5.1. Experimental setup

To train and evaluate the proposed method, we used a desktop computer with Nvidia RTX3090 TI GPU, Intel core i7 @ 3.2GHz CPU, and 64 RAM. We resized the training and test images/layout masks to a fixed size of 400×400. We trained the models using the Tensorflow Keras environment with Adam's optimizer with weight decay (initial learning rate of 0.001 and a weight decay of 0.0001). The convNext-Tiny and Mxception feature extractors were initialized using ImageNet [23] weights, and trained equally

Method	PE(%) LSUN	PE(%) Hedau	Time
Hedau *et al.* [16]	24.23	21.2	-
Ramalingam *et al.* [24]	-	13.34	-
Zhao *et al.* [25]	-	14.5	-
Mallya and Lazebnik [1]	16.71	12.83	-
Dasgupta *et al.* [2]	10.63	9.73	30 s
Ren *et al.* [4]	9.31	8.67	-
Zhao *et al.* [26]	5.29	6.60	1.79 s
RoomNet (3-iter) [6]	9.86	8.36	166 ms
RoomNet (re-imp.) [6]	11.24	12.19	20 ms
Hirzer *et al.* [7]	7.79	7.44	86 ms
Lin *et al.* [5]	6.25	7.41	67 ms
Zhang *et al.* [27]	6.58	7.36	150 s
Zheng *et al.* [8]	6.95	7.21	170 ms
Wang *et al.* [9]	7.99	-	32 ms
ST-RoomNet-MXception	7.15	8.25	66 ms
ST-RoomNet-ConvNext	**5.24**	**7.10**	102 ms

Table 1. Quantitative results on LSUN and Hedau datasets reporting the PE(%) and the time per image.

for 150 epochs.

The metric used for the evaluation of the proposed models is the Pixel Error (PE%), where PE is the average pixel-wise error between the predicted layout mask and the ground truth layout mask averaged over all images.

5.2. Results on LSUN dataset

We evaluated the proposed method on LSUN [15]. The proposed method (ST-RoomNet) could attain PE% of 7.15% using MXception feature extractor, and a PE% of 5.24% using ConvNext feature extractor and a processing speed of 66 ms and 102 ms per image, respectively. The proposed ST-RoomNet using ConvNext feature extractor outperforms other state-of-the-art methods on LSUN room layout estimation as shown in Table 1. Sample qualitative results obtained by the proposed method on LSUN dataset are shown in Figure 4 showing the outstanding performance of the proposed method.

5.3. Results on Hedau dataset

In the evaluation of the proposed method on Hedau [16], the proposed method (ST-RoomNet) could attain PE% of 8.25% using MXception feature extractor, and a PE% of 7.10% using ConvNext feature extractor. Qualitative results shown in Figure 5 show that the proposed method can produce high-quality layouts even better than the ground truth data of Hedau, which contains layouts that do not follow the standard cuboid layout with non-refined edges. In such cases, the proposed method can produce smooth edges following the standard cuboid layout as shown in Figure 5. Table 1 shows that the proposed method using ConvNext feature extractor outperforms other state-of-the-art methods on Hedau Room estimation task.

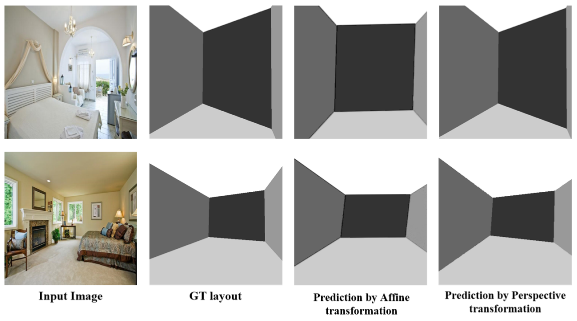
Input Image
GT layout
Prediction by Affine transformation
Prediction by Perspective transformation

Figure 6. Result comparison between the proposed method using Affine and perspective transformation. The affine transformation is limited and cannot learn room layout, however, the perspective transformation is efficient enough for learning the room layouts.

5.4. Ablation study on the spatial transformation layer

In this experiment, we train the proposed method using MXception feature extractor on LSUN dataset to predict the six parameters of affine transformation with an affine transformation layer, and we compare the results with our standard experiment in which the feature extractor predicts the eight parameters of the perspective transformation with a perspective transformation layer. We report the result in Table 2 showing that the affine transformation cannot be employed to learn the room layout problem since the PE% is large (17.56). Since most of the room planes have homography deformations instead of the simple affine deformation in which the parallel lines in the reference scene remain parallel after transformation, and that is not the case in the real layouts of the image. Figure 6 shows the difference between the results obtained by the affine and perspective transformation layer during training highlighting the limitations of the affine transformation and the robustness of the perspective transformation.

Transformation Layer	PE(%)
Affine	17.56
Perspective	**7.15**

Table 2. Comparison between the PE% obtained on LSUN dataset using affine and perspective transformation layers after training using MXception feature extractor.

F. extractor	Convergence	PE(%)
VGG16 [28]	No	-
ResNet50 [29]	No	-
ResNet50V2 [30]	No	-
InceptionV3 [31]	No	-
DenseNet121 [32]	No	-
Xception [20]	Weak	18.23
MobileNetV2 [33]	weak	33.51
MobileNetV3 [34]	weak	27.66
EfficientNetB0 [35]	weak	26.21
EfficientNetB1 [35]	weak	31.21
EfficientNetB2 [35]	weak	26.29
ConvNext-Small [17]	weak	22.74
MXception [22]	Yes	**7.15**
ConvNext-Tiny [17]	Yes	**5.24**

Table 3. Comparison between popular feature extractors (F. extractor) in terms of the ability to converge during training and PE (%) on LSUN dataset.

5.5. Ablation study on the feature extractor

In our experiments, we tried to train different feature extractor networks (popular classification networks) [20, 28–35] to study the effect of the architecture of the feature extractor on the pixel error of the room layout estimation. In each experiment, we replaced the final classification layer of the original classification network with a regression layer (fully connected layer with linear activation) with eight nodes followed by the spatial transformation layer. We found out that feature extractors such as VGG16 [28], Resnet50 [29], ResNet50V2 [30], and InceptionV3 [31] cannot converge at all as the loss explodes during training. Other architectures such as original Xception [20], MobileNetV2 [33], MobileNetV3 [34], EfficientNet (B0, B1, B2) [35], and ConvNext-Small [17] can weakly converge with relatively large pixel error values. They cannot be considered as perfectly fitted models, however, the proposed ConvNext-Tiny [17] and MXception [22] (Modified Xception) can perfectly converge and attain the best (very small) pixel error values. We noticed that both the architectures that could weakly and perfectly converge in the room estimation task are all based on Depthwise convolution which proves the robustness of the depthwise convolution and its strength to learn complex tasks, also the depth of the model plays a role in the convergence as we found that ConvNext-Tiny can perfectly converge during training however it has a reduced architecture of ConvNext-Small. The same happens in the case of MXception which is a reduced version of the original Xception architecture. Table 3 shows a comparison between the different feature extractors stated previously and the corresponding pixel error (PE%) for those that could converge.

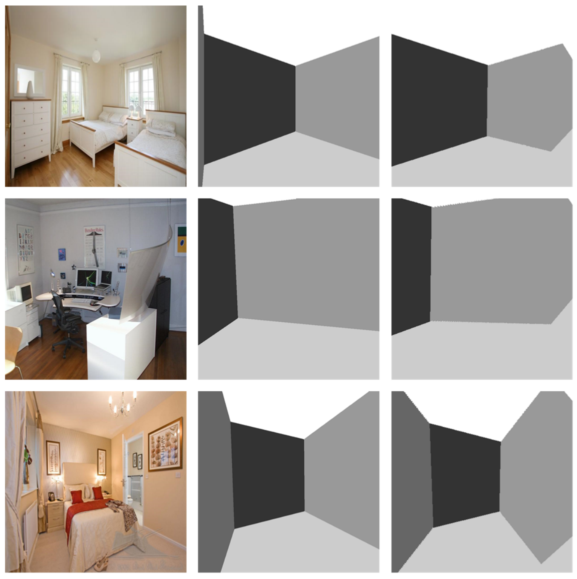

Figure 7. Failure cases of the proposed method due to wrong extrapolation.

6. Limitations and future work

Although the proposed method (ST-RoomNet) is very efficient in learning the room layout problem, we noticed a minor drawback of the proposed method which is the inability of the model to extrapolate the layout. The extrapolation is the task of predicting the values of the pixels outside the scene. This case rarely happens when the required layout has a perspective transformation with a small zooming value (zooming out) which requires extrapolation of the reference layout as shown in Figure 7. The current implementation of the spatial transformation layer simply pads the outside pixels with the nearest value and does not take into consideration the structure of the reference layout. This will be our future work to add extrapolation to the implementation of the spatial transformer. Our future work will also be focused on the room layout estimation of the 3D scenes instead of the 2D scenes as the current method can be extended for the 3D room layout estimation task with the use of the 3D spatial transformer.

7. Conclusion

The proposed method (ST-RoomNet) can efficiently learn the task of room layout estimation using an approach of spatial transformation of a reference room layout to the required layout. It can attain extremely small pixel error (PE%) values (5.24 on LSUN and 7.10 on Hedau) outperforming the state-of-the-art methods on LSUN and Hedau room layout estimation datasets. The proposed method can also process each image frame in 102 ms for the ConvNext-tiny feature extractor and 66 ms for the MXception feature extractor. The proposed method is novel since it does not depend on the traditional approaches of the planer segmentation or keypoint estimation that are employed in the state-of-the-art methods. It depends on the spatial transformation network, especially on the perspective transformation which could model the problem of the room layout estimation with the aid of robust feature extractors such as ConvNext and MXception typically employed as feature extractors.

Acknowledgement

This work was supported by the Research Projects of "Development of automatic screening and hybrid detection system for hazardous material detecting in port container" funded by the Ministry of Oceans and Fisheries, and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2023R1A2C1006944.

References

[1] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *ICCV*, pages 936–944, 2015.

[2] Saumitro Dasgupta, Kuan Fang, Kevin Chen, and Silvio Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *CVPR*, pages 616–624, 2016.

[3] Weidong Zhang, Wei Zhang, Kan Liu, and Jason Gu. Learning to predict high-quality edge maps for room layout estimation. *IEEE Transactions on Multimedia*, 19(5):935–943, 2017.

[4] Yuzhuo Ren, Shangwen Li, Chen Chen, and C.-C. Jay Kuo. A coarse-to-fine indoor layout estimation (cfile) method. In *ACCV*, 2016.

[5] Hung Jin Lin, Sheng-Wei Huang, Shang-Hong Lai, and Chen-Kuo Chiang. Indoor scene layout estimation from a single image. In *In International Conference on Pattern Recognition (ICPR)*, pages 842–847, 2018.

[6] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. In *ICCV*, pages 4875–4884, 2017.

[7] Martin Hirzer, Peter M. Roth, and Vincent Lepetit. Smart hypothesis generation for efficient and robust room layout estimation. In *WACV*, pages 2901–2909, 2020.

[8] Wenzhao Zheng, Jiwen Lu, and Jie Zhou. Structural deep metric learning for room layout estimation. In *ECCV*, 2020.

[9] Wang Aopeng, Wen Shiting, Gao Yunjun, Li Qing, Deng Ke, and Pang Chaoyi. Toward enhancing room layout estimation by feature pyramid networks. In *Data Science and Engineering*, pages 213—-224, 2022.

[10] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *CVPR*, pages 1849–1856, 2009.

[11] David C. Lee, Abhinav Kumar Gupta, Martial Hebert, and Takeo Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010.

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *Adv. Neural Inform. Process. Syst. (NIPS)*, pages 2017–2025, 2015.

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[15] Zhang Y, Yu F, Song S, Xu P, Seff A, and Xiao J. Largescale scene understanding challenge: Room layout estimation. In *Princeton University*, 2016.

[16] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *CVPR*, pages 1849–1856, 2009.

[17] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11966–11976, 2022.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, pages 11966–11976, 2021.

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002, 2021.

[20] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1800–1807, 2017.

[21] Ba J. L., Kiros J. R., and Hinton G. E. Layer normalization. In *NIPS*, 2016.

[22] Hatem Ibrahem, Ahmed Diefy Ahmed Salem, and Hyun-Soo Kang. Real-time weakly supervised object detection using center-of-features localization. *IEEE Access*, 9:38742–38756, 2021.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[24] Srikumar Ramalingam, Jaishanker K. Pillai, Arpit Jain, and Yuichi Taguchi. Manhattan junction catalogue for spatial reasoning of indoor scenes. In *CVPR*, pages 3065–3072, 2013.

[25] Yibiao Zhao and Song-Chun Zhu. Scene parsing by integrating function, geometry and appearance models. In *CVPR*, pages 3119–3126, 2013.

[26] Hao Zhao, Ming Lu, Anbang Yao, Yiwen Guo, Yurong Chen, and Li Zhang. Physics inspired optimization on semantic transfer features: An alternative method for room layout estimation. In *CVPR*, pages 870–878, 2017.

[27] W. Zhang, Wei Zhang, and Jason Jianjun Gu. Edge-semantic learning strategy for layout estimation in indoor environment. *IEEE Transactions on Cybernetics*, 50:2730–2739, 2019.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016.

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[32] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269, 2017.

[33] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

[34] Andrew Howard, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, Yukun Zhu, Ruoming Pang, Hartwig Adam, and Quoc Le. Searching for mobilenetv3. In *ICCV*, pages 1314–1324, 2019.

[35] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019.