Learning Informative Edge Maps for Indoor Scene Layout Prediction

Arun Mallya and Svetlana Lazebnik
Dept. of Computer Science, University of Illinois at Urbana-Champaign
{amallya2,slazebni}@illinois.edu

Abstract

In this paper, we introduce new edge-based features for the task of recovering the 3D layout of an indoor scene from a single image. Indoor scenes have certain edges that are very informative about the spatial layout of the room, namely, the edges formed by the pairwise intersections of room faces (two walls, wall and ceiling, wall and floor). In contrast with previous approaches that rely on area-based features like geometric context and orientation maps, our method attempts to directly detect these informative edges. We learn to predict 'informative edge' probability maps using two recent methods that exploit local and global context, respectively: structured edge detection forests, and a fully convolutional network for pixelwise labeling. We show that the fully convolutional network is quite successful at predicting the informative edges even when they lack contrast or are occluded, and that the accuracy can be further improved by training the network to jointly predict the edges and the geometric context. Using features derived from the 'informative edge' maps, we learn a maximum margin structured classifier that achieves state-of-the-art performance on layout prediction.

1. Introduction

Consider the task of finding the spatial layout of the indoor scene depicted in Fig. 1. In the widely accepted framework of Hedau *et al.* [10], this task is formulated as finding a 3D box, such as the one outlined with green lines, that best fits the room. Existing approaches to this problem focus on complex mid-level image features [10, 16, 20] or powerful inference procedures [21, 22, 26, 28]. The problem of layout estimation would be greatly simplified – indeed, made almost trivial – if we could directly find edges that are very informative about the 3D structure of the room, namely those between two walls, the walls and the ceiling, and the walls and the floor. Unfortunately, these edges are neither very prominent nor always visible. We can see these issues in Fig. 1: high-contrast "clutter" edges exist between the arm chair and the wall, and the TV and the shelf, while the wall and the ceiling are nearly of the same color. Moreover, the arm chair and the cupboard heavily occlude the edges between the walls and the floor. Thus, inferring 3D box edges directly from low-level pixel information seems very challenging. For this reason, most existing approaches rely on mid-level area-based features, such as Geometric Context [10] and Orientation Maps [16], as an intermediate step for layout estimation. At the same time, in the literature on image segmentation and perceptual organization, there has been successful work on learning to predict object contours from low-level pixel information [2, 6, 17, 31] and this work motivates us to directly predict informative edges for room layout estimation. Given an image, we determine its informative edge map and subsequently use it to predict the best-fit 3D box for the image, as depicted in the right half of Fig. 1. Our contributions are as follows:

Image with groundtruth box | Inferred edge map with best box estimate

Figure 1. An indoor scene (left) and a map of the edges that determine its 3D layout (right). We present novel techniques to obtain this informative edge map from an image and demonstrate its effectiveness for determining the spatial layout of a room.

1. We adapt two recently developed pixel labeling methods for learning to predict informative edge maps for room layout: Structured Forests for Edge Detection [7] and Fully Convolutional Networks (FCNs) [18]. We find that FCNs show an impressive level of performance on this task, especially when trained to jointly predict informative edges and geometric context (Sec. 3).
2. We propose a structured layout inference method that

ranks adaptively generated room layout candidates based on features computed from the informative edge map (Sec. 4). Our adaptive layout generation is simpler than the exact inference approach of Schwing and Urtasun [22] but works very well, while our edge-based features are simpler than standard Geometric Context [10] and Orientation Map [16] features.

3. We perform extensive quantitative assessments and demonstrate that our methods achieve state-of-the-art performance on the standard Hedau dataset [10] as well as the newly released and larger LSUN dataset [1] (Sec. 5).

2. Related Work

The idea of using a cuboidal box to approximate the 3D layout of indoor scenes was first introduced in 2009 by Hedau *et al.* [10]. By adapting the techniques of Hoiem *et al.* [13], the authors derived geometric context labels for indoor scenes, which assigned to each pixel a class from {*middle wall, right wall, left wall, ceiling, floor, object*}. Then they used features extracted from these labels to train a structured regressor to rank multiple box candidates and determine the best fitting box. In the same year, Lee *et al.* [16] introduced orientation map features which tried to reason about the geometric layout of the room based on line segments detected in the room. Their area-based features, along with structured regressors, have become a standard framework for determining the spatial layout [22, 21, 20]. More recently, Ramalingam *et al.* [20], have obtained improved results using cues from line junctions in the image. To the best of our knowledge, no one has so far attempted to specifically identify edges such as those between different faces (wall, ceiling, floor) of a room which directly determine the spatial layout of a room.

Several works have focused on developing better inference procedures to search the space of possible room layouts more effectively. Schwing et al. [22, 21] proposed a framework for exact inference of the best-fit 3D box based on integral geometry. Wang *et al.* [26] proposed a discriminative learning method that used latent variables to jointly infer the 3D scene and the clutter. In our work, we show that given better edge-based features, a simpler inference procedure is sufficient to obtain state-of-the-art performance.

Moving past 2D area-based features, some works have incorporated 3D information obtained from external sources. By using information from online furniture and appliance catalogs, Del Pero *et al.* [4] proposed a model that used Bayesian inference with 3D reasoning to simultaneously predict the geometry and objects present in a room. The authors emphasized in their work the importance of detecting cues such as faint wall edges to improve the layout estimation of a room, but did not have a method for explicitly targeting informative edges. Zhao *et al.* [29] also used 3D models obtained from the Google 3D Warehouse along with a stochastic scene grammar to recover the 3D geometry of indoor scenes. Unlike these works, we do not use any explicit 3D information in our inference.

3. Learning to Predict Informative Edges

As stated in the Introduction, we define 'informative' edges as the edges of the projected 3D box that fits the room. There are three types of such edges: those between two wall faces, between walls and the ceiling, and between walls and the floor. Fig. 2 illustrates three images with their groundtruth edge maps according to this definition. These maps are generated from the original groundtruth format of [10], which consists of polygons corresponding to the different room faces. We take the pixel masks of these polygons, dilate them by 4 pixels, and find their intersection, which ends up being about 4 pixels thick on average. All of the pixels in the resulting mask are considered to be positive examples of informative edges (even where the actual edges are occluded by clutter such as furniture), and all other pixels are considered as part of the background or negative class.

We explore two state-of-the-art dense pixel labeling methods for learning to predict informative edge maps. The first method, discussed in Section 3.1, is a Structured Forest [7], which operates on small input patches, and therefore utilizes only local context. The second method, discussed in 3.2, is a Fully Convolutional Network [18], which operates on the image as a whole and therefore has a good way of incorporating global context.

3.1. Structured Forest Edge Maps

Structured Forests were introduced by Dollár *et al.* [7] as an efficient means for generating contour maps and achieved remarkable performance on the BSDS500 segmentation dataset [2]. Exploiting the fact that edge labels within a small image patch are highly interdependent, this method trains an ensemble of trees, each of which operates on an input image patch and produces as output a patch corresponding to the edge labels in the input patch. An image is divided into multiple overlapping patches, and for each patch, the outputs are obtained from multiple trees. These outputs are then overlaid and averaged to produce an edge probability map for the entire image. One advantage of the structured forests is that they can be taught to detect specific types of edges and we exploit this ability in our work. In our experiments, we use the standard settings for the forest as specified in [7] and learn an ensemble of 8 trees, each to a maximum depth of 64, with an input patch size of 32×32 pixels. In addition to the color and gradient features of [7], we have also found it useful to include location features for our problem. In order to encode global location information of an edge, we divide the height and width of the image into

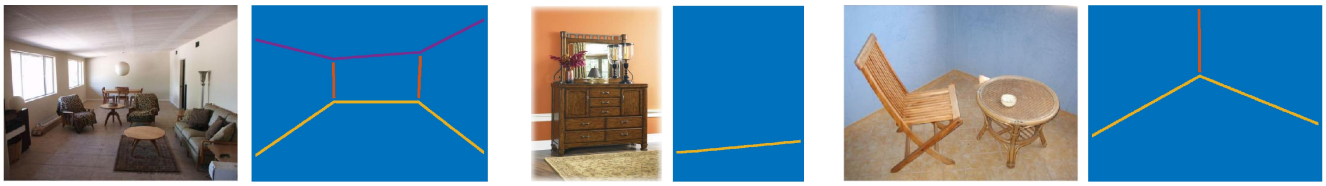

Figure 2. Examples of training images with their respective groundtruth informative edge maps. Orange, purple, and yellow lines indicate wall/wall, wall/ceiling, and wall/floor edges respectively. Blue indicates background containing uninformative edges. Note that these edges are marked by ignoring the occluding clutter, as if the room were empty.

10 bins, and use the bin index as the location feature for a pixel. Image patches which contain at least one groundtruth edge pixel are treated as positives examples and those that do not contain any groundtruth edge pixels are treated as negatives.

3.2. Fully Convolutional Network Edge Maps

Fully Convolutional Networks (FCNs) [18] have been shown to achieve state-of-the-art performance on pixel labeling tasks over multiple datasets including PASCAL VOC 2011/2012 and NYUDv2 [19]. These networks are obtained from image-level classification networks trained on the ImageNet dataset [5], such as the AlexNet [15] or VGG-16 [24], by converting each fully connected layer into a convolutional layer with a kernel covering the entire input region, and then fine-tuning for the pixel-level labeling task. The receptive field size of the last convolutional layer of the FCN is very large, typically around 400 pixels, resulting in low-resolution, coarse output maps. To obtain higher-quality label maps, a deconvolutional layer is used to upsample the coarse outputs to dense pixelwise outputs. FCN models are naturally suited for tasks that require contextual information from the entire image.

One innovation in our paper is the joint training of FCNs for two tasks: prediction of the informative edge map and prediction of geometric context labels [13, 10]. As stated in Section 2, geometric context labels correspond to the five labels of the different room faces plus an 'object' or 'clutter' label. On the other hand, our informative edge maps correspond to the boundaries between faces. Thus, the two types of labels are complementary and, the issue of clutter aside, one could in principle generate one label map from the other. In practice, however, face membership and face boundary prediction depend on different types of low-level cues, and it is interesting to see whether joint training can help to reinforce the quality of both map types. We perform joint training by sharing all layers of the FCN except for the deconvolutional layers which produce the softmax probability maps for the respective types of output. The total loss of the network is the sum of the two cross-entropy classification losses: one for informative edge label prediction, and one for geometric context label prediction. In Section 5, we provide evidence that joint loss optimization indeed helps to improve the accuracy of the edge maps.

In our work, we learn FCNs with the VGG-16 structure using Caffe [14]. To initialize the FCN weights for our task, we found it important to use a network pre-trained for segmentation on an indoor scene dataset. Specifically, we use the FCN with 32-pixel prediction stride (FCN-32) trained on the NYUDv2 RGBD dataset for the 40-class indoor semantic segmentation task of [9].[1] The original network from [18] has two input streams, one for the RGB input, and one for the depth feature inputs. We discard the layers corresponding to the depth inputs and use the remaining layers to initialize our FCN. We also tried initializing with the FCN-32 trained for semantic segmentation on the PASCAL dataset, but obtained very poor performance. We fine-tune our network with a base learning rate of 10^{-4} with high momentum of 0.99. We use a higher learning rate of 10^{-3} for the newly inserted final convolutional and deconvolutional layers. The best parameter settings and stopping iteration are tuned on the validation set.

Fig. 3 shows edge maps output by the Structured Forest and the FCN for images with varying degrees of clutter. The edge maps produced by the forests tend to be dense and noisy, as compared to sparser and cleaner maps produced by the FCN. Also, the FCN is better able to predict edges that are occluded by clutter. Further quantitative evaluation of edge map prediction will be given in Section 5.

4. Inference Model

We use the framework introduced in [10] for representing the spatial layout of an indoor scene, which consists of the following steps: (1) Estimate the vanishing points of the room (Section 4.1); (2) Based on the vanishing points, generate a number of candidate box layouts (Section 4.2); (3) Learn a structured regressor to rank the box layouts based on features derived from the informative edge maps and other image information (Section 4.3).

[1] This network is publicly available at https://gist.github.com/longjon/16db1e4ad3afc2614067 and was last accessed on 18 April 2015.

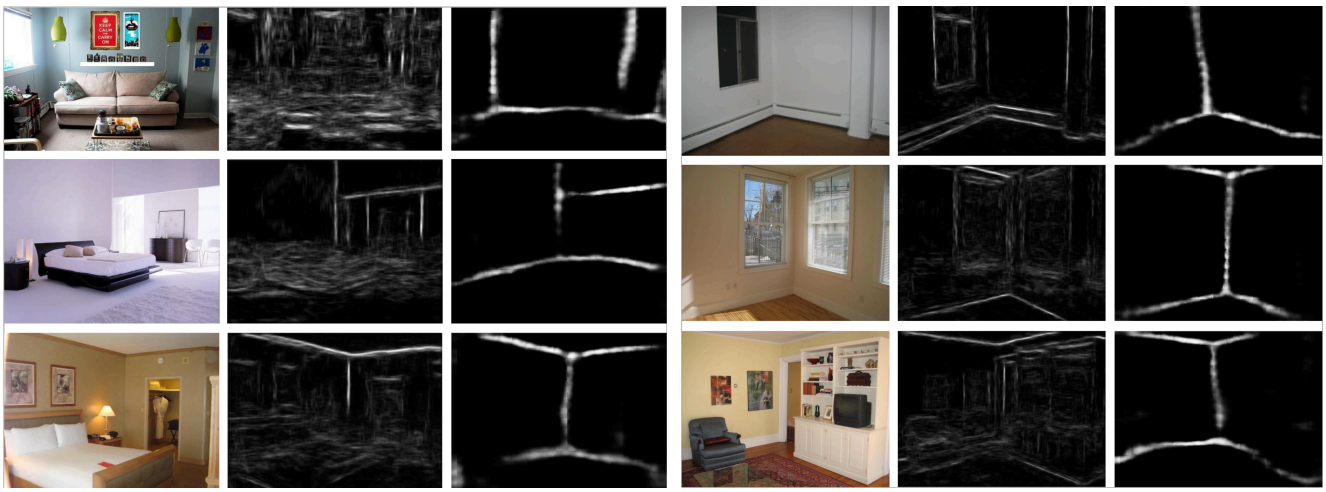

Figure 3. Comparison of Informative Edge Maps generated by the Structured Forest and the FCN. Each triplet of images from left to right, top to bottom shows the input indoor scene, the output of the 1 Class Forest trained on the SUNbox dataset (*Forest-1 Class-SUNbox*), and the 1 Class FCN trained on the Hedau+ dataset (*FCN-1 Class-Hedau+(Joint)*). Note that the FCN produces good edge maps even in the presence of clutter such as the couch and the bed in the images above.

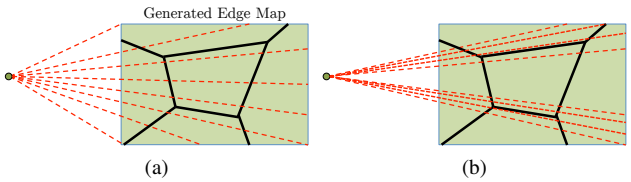

Figure 4. Adaptive Layout Generation. (a) Uniformly spaced sectors originating from the horizontal vanishing point are first generated. $K = 1$ sectors with the highest average edge strength per pixel are chosen, both above and below the horizontal line through the vanishing point. (b) $N = 2$ rays are sampled from each selected sector.

4.1. Vanishing Point Estimation

The first step in finding the best-fit layout is to estimate the vanishing points of the scene given the image. We use the approach of [10], which makes the *Manhattan World* assumption that there exist three dominant orthogonal vanishing points. This approach votes for vanishing points using edges detected in the image by using the Canny edge detector [3]. We tried substituting the Canny edges with our informative edge maps, but this resulted in bad estimates of the vanishing points. This is because edges that we consider 'not informative' for the sake of final layout estimation, such as those on furniture, tiling, windows, etc., are actually informative about the location of the vanishing points, and detecting a large number of such edges helps with robust vanishing point estimation.

4.2. Adaptive Layout Generation

In the next step, a family of 3D boxes is generated by sampling rays originating from the vanishing points. A single layout is generated by sampling two rays each from the vanishing points corresponding to the mostly horizontal and the mostly vertical lines in the image, and then connecting their intersections with rays originating from the third vanishing point. Hedau et al. [10] sampled 10 rays per direction uniformly by angle. Schwing *et al.* [21] used a denser sampling of about 50 rays per vanishing point to obtain a significant reduction in the layout estimation error. Other works performed adaptive ray sampling by taking into account edge and corner features [4] or edge junctions [20]. Finally, Schwing et al. [22] avoided explicit up-front ray sampling through their exact inference procedure.

Since the exact inference procedure of [22] is hard to implement, we came up with our own heuristic strategy that attempts to sample rays more densely in sectors of the image where the informative edge map has higher energy. This strategy is illustrated in Fig. 4. Once the informative edge maps and the vanishing points have been estimated, we draw uniformly spaced sectors (w.r.t. angle) originating from the horizontal vanishing points, as shown in red, in Fig. 4 (a). We then rank all resulting sectors by the average informative edge strength and retain top K sectors each in the upper and the lower parts of the image formed by drawing a horizontal line through the horizontal vanishing point.[2] Finally, we sample N rays uniformly from each of the selected sectors. For $K = 1$, and $N = 2$, Fig. 4 (b) shows the selected sectors and the final sampled rays. An analogous procedure is repeated for selecting rays originating from the vertical vanishing point.

[2] A sector is assigned to the upper or lower part of the image based on the angle of its top ray.

4.3. Ranking Box Layouts

Once multiple candidate layouts are generated, the best one is selected through the use of a max-margin structured regressor. Let us denote an image by x, and a 3D box layout by y. We wish to learn a function $f(x, y; w)$ parameterized by some weights w that ranks layouts according to their compatibility with the input image. For an image x_i with a best-fit box y_i, the function f should assign a high score to layout y that is *similar* to y_i. Given a previously unseen test image x, the predicted layout is given by

$$y^* = \arg\max_y f(x, y; w)\,. \tag{1}$$

The above structured regression problem is solved using a max-margin framework [25]. The function f is restricted to take the form of $f(x, y) = w^T \psi(x, y)$, where $\psi(x, y)$ is a feature vector for the input image x and a given layout y (see [10] for details). We explore three types of input features $\psi(x, y)$.

Informative Edge (IE) Features. The first type of feature is based on how well a proposed layout aligns with the generated informative edge maps. Given a proposed layout y for an image x, we generate binary masks of each of the three types of edges (wall/wall, wall/ceiling, wall/floor) in the layout and then compute the dot products of these masks with the corresponding informative edge maps generated for image x. In all of our experiments, we use binary masks with an edge thickness of 10 pixels.

Line Membership (LM) and Geometric Context (GC) Features. These two types of features are borrowed from the prior work of Hedau *et al.* [10]. We use the unweighted line membership features derived from the straight lines detected in the image during vanishing point estimation. These features consist of a scalar value computed for each face of a layout, along with the percentage area of a face. We also experimented with geometric context (GC) features as described in [10]. These are line membership features weighted by average label confidences within each type of face as given by the geometric context labels, along with average label confidences within each face. However, as we will show in Section 5, once the IE features are introduced, we do not need the GC features to get state-of-the-art performance, unlike most of the previous works.

The dimensionalities of the IE, LM, and GC features are 3, 10, and 45 respectively. Section 5 presents detailed information about the performance of these different features at predicting the room layout.

5. Experimental Results

Datasets. The standard dataset for indoor layout prediction is the dataset of Hedau et al. [10] (referred to as the Hedau dataset in this work), which consists of 209 training images and 105 testing images, for evaluation purposes. We use the same train/test split as other existing work.

Dataset	# Train	# Val.	# Test
Hedau	209	–	105
Hedau+	284	53	–
SUNbox	543	53	–
LSUN	–	–	1000

Table 1. Statistics of datasets used in this work. The Hedau dataset is our primary dataset for training the structured regressor and for layout prediction evaluation. Hedau+ and SUNbox are larger datasets we use to train detectors for informative edges. Finally, we use the LSUN test set for additional evaluation of layout prediction. See text for details.

Like in previous work [10], the structured regressor is trained on the Hedau train set. As the Hedau train set is rather small for training local edge predictors, we created a larger dataset, referred to as Hedau+, by augmenting the Hedau dataset with 128 extra labeled examples from the Bedroom dataset used in [11, 12]. The Hedau+ dataset consists of 284 train (209 from the Hedau dataset, along with 75 new images) and 53 validation examples. Hedau+ worked well for training the FCN model. However, it caused significant overfitting for training the Structured Forests, as the structured regressor was also trained on a large subset of the same images. As a result, it became necessary to introduce another training set disjoint from Hedau. We created this dataset, referred to as SUNbox, by collecting and manually annotating 543 images of indoor scenes from the SUN2012 dataset [27]. The images of the SUNbox dataset are generally more cluttered and have a larger variety in the scenes depicted than the Hedau dataset. As validation set for SUNbox, we use the same 53 images as in Hedau+. For training the FCN, we augmented the training set by 16 times using standard transformations such as cropping, mild rotation and scaling. Augmentation was not found to help the structured forests.

Finally, to test how our method generalizes to a larger, more complex dataset, we tested our models on the recently introduced LSUN dataset [1]. This dataset has 4000 training and 1000 test images. Due to constraints on time and computational resources (the dataset was released after the ICCV submission deadline), we did not use its train/validation set, but directly tested our FCN trained on Hedau+ on the LSUN test set. As we will show in Section 5, our model generalizes well despite not being retrained.

Dataset statistics are summarized in Table 1.

Informative Edge Prediction. We use two setups for training informative edge predictors: in the three-class setup, we have three positive classes, namely *wall/wall*, *wall/ceiling*, and *wall/floor*; in the one-class setup, we consider all informative edges to belong to a single class.

Setting	Forest			FCN		
	ODS	OIS	AP	ODS	OIS	AP
BSDS [7]	0.159	0.165	0.052	–	–	–
3 Class - Hedau+	**0.178**	0.176	**0.104**	0.235	0.237	0.084
3 Class - SUNbox	0.177	0.169	0.103	0.227	0.232	0.086
1 Class - Hedau+	0.174	**0.177**	0.094	0.226	0.227	0.080
1 Class - Hedau+ (Joint)	–	–	–	**0.255**	**0.263**	**0.130**
1 Class - SUNbox	**0.178**	0.172	0.103	0.179	0.180	0.056
1 Class - SUNbox (Joint)	–	–	–	0.206	0.209	0.069

Table 2. Informative edge prediction accuracy of different methods on the Hedau test set. The methods vary in number of positive classes for the edge maps (3 vs. 1), training dataset (Hedau+ or SUNbox) and training setup (Joint denotes joint training for edge maps and geometric context as discussed in Section 3). Performance is evaluated using three standard measures [2] of fixed contour threshold (ODS), per-image best threshold (OIS), and average precision (AP). For all metrics, higher values are better.

Setting	Forest		FCN	
	Uniform Layout Error (%)	Adaptive Layout Error (%)	Uniform Layout Error (%)	Adaptive Layout Error (%)
3 Class - Hedau+	23.66	20.59	26.19	16.05
3 Class - SUNbox	**19.97**	**16.89**	20.90	16.20
1 Class - Hedau+	23.15	21.71	20.62	13.89
1 Class - Hedau+ (Joint)	–	–	**18.30**	**12.83**
1 Class - SUNbox	20.81	18.03	23.64	18.43
1 Class - SUNbox (Joint)	–	–	18.95	15.09

Table 3. Layout Estimation errors obtained on the Hedau test set by using different types of Informative Edge Features. The error is the percentage of pixels whose face identity disagrees with the ground truth. The structured regressor for layout prediction is trained using Line Membership and Informative Edge features. Uniform Layout corresponds to using 10 uniformly sampled rays per vanishing point and Adaptive Layout uses parameters $K = 2$, $N = 3$ (refer to Sec. 4 for definition).

We report quantitative results of informative edge prediction in Table 2 using the evaluation framework of [2]. As is the common practice, we apply non-maximal suppression (NMS) on the edge maps to obtain thinned edges for evaluation. As a baseline, we compare our methods against the Structured Forests of [7] trained on the Berkeley Segmentation Dataset (BSDS) for generic contour detection.[3] The performance is significantly lower than that of all our methods, confirming that task-specific training is required for finding informative edges for layout prediction. The best FCN beats the best forest on all metrics. Moreover, it is evident that joint loss optimization improves the quality of the edge maps obtained.

For the jointly trained FCN, the test errors for the geometric context prediction stream are around 28-31% – higher than the error of 26.9% obtained by the original method of Hoiem *et al.* [13]. This is most likely due to Hoiem's system using mid-level features custom-tailored for this problem. However, since geometric context prediction was not a central focus of this work, we did not investigate this issue further. For our purposes, the important finding is that learning to jointly predict geometric context and informative edge maps significantly improves the accuracy of the latter.

The numbers in Table 2, especially the low AP, suggest that our informative edge prediction suffers from poor fine-grained localization. Qualitatively, we observe high edge probabilities in the correct areas, however, on thinning using NMS, we obtain poor precision. Nevertheless, these edge maps work very well for the end goal of estimating the spatial layout of indoor scenes, as discussed next.

Layout Estimation Performance. In Table 3, we report the pixel classification errors obtained with different strategies for informative edge prediction and layout sampling. Uniform layout sampling uses 10 horizontal and 10 vertical rays as in [10], while adaptive sampling uses parameter settings of $K = 2$ and $N = 3$ as described in Section 4.2. This gives only a small increase in the number of rays per vanishing point (12 vs. 10), but helps to reduce errors across the board, for all the feature settings.

Two other interesting observations can be made from Table 3. First, while all the random forest configurations perform almost the same on edge prediction (Table 2), their performance on the final spatial layout estimation task is

[3] We also benchmarked Canny Edges on this task, but their performance is very poor for all measures.

very different. In particular, as mentioned earlier in this section, the forests trained on the Hedau+ dataset suffer from significant overfitting when an overlapping subset of data is also used to train the structured regressor. Second, for the FCNs, we get the best results with single-class edge maps, while for the forests, we get the best results with three-class edge maps. We conjecture that FCNs can better handle intra-class variation, making the availability of more training data a bigger factor. Forests, on the other hand, use weaker context in making predictions and probably do better with less intra-class variation in the training data. In the end, the best FCN beats the best forest by over 4%, making it the decisive winner.

Setting	Error (%)
Line Membership (LM) Feature Only	26.42
Forest - IE Only	23.62
Forest - LM + IE	16.89
Forest - LM + IE + GC	16.85
FCN - IE Only	14.74
FCN - LM + IE	**12.83**
FCN - LM + IE + GC	13.83

Table 4. Pixelwise layout misclassification errors obtained on the Hedau dataset by using different combinations of features. Forest and FCN stand for the best performing forest (*Forest - 3 Class - SUNbox*) and FCN (*FCN - 1 Class - Hedau+ (Joint)*) respectively. The results show that the combination of Line Membership (LM) and FCN-based Informative Edge (IE) features works the best.

Analysis of Features for the Structured Regressor. The results in Table 3 were obtained using the combination of line membership (LM) and informative edge (IE) features. Next, we perform a more in-depth analysis of the contributions of different features specified in Section 4.3. The results are presented in Table 4. By using just the LM features, we obtain our highest error of 26.42%. Using just the IE features derived from the best performing forest and FCN, we obtain lower errors of 23.62% and 14.74% respectively. These numbers hint at the true potential of the FCN-based features as by using them alone, we already come within 1.5% of the previous reported state of the art [20]. By augmenting IE features with LM, we obtain our best performance. This suggests that our informative edge maps suffer from the inability to precisely localize edges, as the LM features are computed from lines that are finely localized and just a pixel wide. Our FCN is derived from VGG-16, which has five layers of 2×2 pooling with a stride of 2. This means that 32 pixels of the input get mapped to a single top-layer neuron, which makes it difficult to produce high-resolution edge maps. As for the forest, it is unable to handle image patches containing clutter very well since it uses only local context.

Rows 4 and 7 of Table 4 report performance with geometric context (GC) features, similar to prior work [10, 22, 21, 20, 16]. With the best-performing forest, we use the geometric context labels output by the method of Hedau *et al.* [10], while for the jointly trained FCN, we use the labels output by the FCN itself. Surprisingly, adding the GC features does not reduce our layout prediction error, and in the case of the FCN, the error actually goes up. After qualitatively analyzing the layout predictions and the geometric context labels, we believe that this is due to the lack of smoothness and consistency constraints over the GC labels, which some of the newer works try to address [30, 23]. These results also seem to suggest that given good informative edge maps, not a lot of information is added by the geometric context labels.

Method	Error (%)
Hedau Test Dataset	
Hedau *et al.* [10]	21.20
Del Pero *et al.* [4]	16.30
Gupta *et al.* [8]	16.20
Zhao *et al.* [29]	14.50
Schwing *et al.* [22]	13.59
Ramalingam *et al.* [20]	13.34
Ours (Forest - 3 Class - SUNbox)	16.89
Ours (FCN - 1 Class - Joint - Hedau+)	**12.83**
LSUN Test Dataset	
Hedau *et al.* [10]	24.23
Ours (FCN - 1 Class - Joint - Hedau+)	**16.71**

Table 5. Comparison of the best reported pixel misclassification errors of different methods. The FCN based features combined with adaptive ray sampling obtain state-of-the-art error, beating those that use exact ray inference and features including Geometric Context, Orientation Map, and Junction information.

Comparison with the state of the art. Table 5 compares the best results of previous methods on the layout estimation task to those of our best model. Our error of 12.83% beats the previous best error of 13.34% of Ramalingam et al. [20]. It is interesting that even though we do not perform exact inference like [22], or sample as many rays as [21], we are able to obtain very good results using the combination of our informative edge based features and adaptive sampling. Furthermore, unlike all the previous methods, which used an array of features including Geometric Context, Orientation Maps, Junction Information, and their combinations, we only rely on edge-based features. We also do not use any information about 3D clutter shapes unlike [4, 29]. Fig. 5 displays some of our best and worst predicted layouts on the Hedau dataset.

Finally, as mentioned in the beginning of this section, we evaluated our model on the much larger LSUN test dataset without any retraining or fine-tuning. As can be seen from the last two lines of Table 5, the prediction error on this test set is about 4% higher than on the Hedau test set, but this is

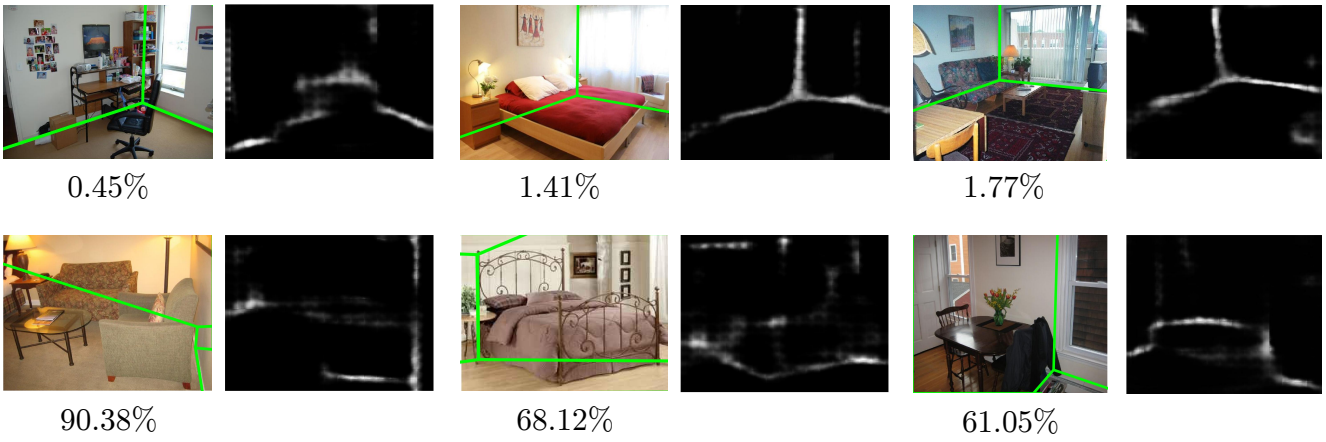

Figure 5. Examples of best and worst results on the Hedau test set by our best performing method, *FCN - 1 Class - Hedau+ (Joint)*. The top row shows pairs of image and predicted informative edge map of some of the best results obtained, while the bottom row shows some of the worst. Below each image is the pixel misclassification error percentage.

at least partially due to the more diverse nature of this test set. The only currently available baseline for LSUN is for the method of [10], and we beat it handily.

6. Conclusion

In this paper, we have presented two methods for the prediction of informative edge maps and achieved state-of-the-art results on indoor scene layout prediction by using just edge-based features, unlike previous works that rely on mid-level area-based features like geometric context [10] and orientation maps [16]. Among the learned informative edge prediction methods, the FCN clearly outperforms the Structured Forests on all accounts. Furthermore, we show that our FCN trained on the Hedau dataset generalizes well by achieving state-of-the-art results on the LSUN dataset. We also show that geometric context [13] helps in training the network, but once we use the network to obtain good features that focus on boundaries, GC features do not add much value, if any, to determining the layout of an indoor scene.

Our final prediction pipeline is fairly straightforward: Given an image, extract the informative edge map by running the trained FCN directly on top of image pixels, detect vanishing points, sample candidate layouts, and finally rank the layouts using simple features computed from the edge map and the lines used to detect vanishing points. This is in contrast to prior state of the art, which requires cumbersome, multi-stage processes such as extracting superpixels, computing their statistics, and applying classifiers, just for generating the intermediate geometric context features, followed by complex inference procedures.

For future work, we believe it is possible to further improve the resolution of the edge maps, and hence reduce the layout error by using an FCN with a fewer pooling layers or a smaller deconvolutional stride. Training FCNs on the newly available LSUN data is also likely to prove helpful.

Acknowledgments. We would like to thank Mariyam Khalid for collecting the SUNbox dataset and annotations. This work was partially supported by NSF grants IIS-1228082 and CIF-1302438, Xerox UAC, and the Sloan Foundation.

References

[1] LSUN Room Layout Estimation Dataset. http://lsun.cs.princeton.edu/. Accessed: 15 September 2015. 2, 5

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011. 1, 2, 6

[3] J. Canny. A computational approach to edge detection. *TPAMI*, 8(6):679–698, Nov 1986. 4

[4] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *CVPR*, 2012. 2, 4, 7

[5] 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. 3

[6] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *CVPR*, 2006. 1

[7] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013. 1, 2, 6

[8] A. Gupta, M. Hebert, T. Kanade, and D. M. Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010. 7

[9] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013. 3

[10] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *CVPR*, 2009. 1, 2, 3, 4, 5, 6, 7, 8

[11] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*. 2010. 5

[12] V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. In *CVPR*, 2012. 5

[13] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1):151–172, 2007. 2, 3, 6, 8

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 3

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3

[16] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 1, 2, 7, 8

[17] J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013. 1

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR (to appear)*, 2015. 1, 2, 3

[19] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 3

[20] S. Ramalingam, J. K. Pillai, A. Jain, and Y. Taguchi. Manhattan junction catalogue for spatial reasoning of indoor scenes. In *CVPR*, 2013. 1, 2, 4, 7

[21] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *CVPR*, 2012. 1, 2, 4, 7

[22] A. G. Schwing and R. Urtasun. Efficient exact inference for 3d indoor scene understanding. In *ECCV*. 2012. 1, 2, 4, 7

[23] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015. 7

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3

[25] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, pages 1453–1484, 2005. 5

[26] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*. 2010. 1, 2

[27] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 5

[28] J. Zhang, C. Kan, A. G. Schwing, and R. Urtasun. Estimating the 3d layout of indoor scenes and its clutter from depth sensors. In *ICCV*, 2013. 1

[29] Y. Zhao and S.-C. Zhu. Scene parsing by integrating function, geometry and appearance models. In *CVPR*. IEEE, 2013. 2, 7

[30] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. *arXiv preprint arXiv:1502.03240*, 2015. 7

[31] S. Zheng, A. Yuille, and Z. Tu. Detecting object boundaries using low-, mid-, and high-level information. *Computer Vision and Image Understanding*, 114(10):1055–1067, 2010. 1