Learning to Synthesize 3D Indoor Scenes from Monocular Images

ABSTRACT

Depth images have always been playing critical roles for indoor scene understanding problems, and are particularly important for tasks in which 3D inferences are involved. However, since depth images are not universally available, abandoning them from the testing stage can significantly improve the generality of a method. In this work, we consider the scenarios where depth images are not available in the testing data, and propose to learn a convolutional long short-term memory (Conv LSTM) network and a regression convolutional neural network (regression ConvNet) using only monocular RGB images. The proposed networks benefit from 2D segmentations, object-level spatial context, object-scene dependencies and objects’ geometric information, where optimization is governed by the semantic label loss, which measures the label consistencies of both objects and scenes, and the 3D geometrical loss, which measures the correctness of objects’ 6Dof estimation. Conv LSTM and regression ConvNet are applied to scene/object classification, object detection and 6Dof estimation tasks respectively, where we utilize the joint inference from both networks and further provide the perspective of synthesizing fully rigged 3D scenes according to objects’ arrangements in monocular images. Both quantitative and qualitative experimental results are provided on the NYU-v2 dataset, and we demonstrate that the proposed Conv LSTM can achieve state-of-the-art performance without requiring the depth information.

CCS CONCEPTS
- Deep Learning for Multimedia; - Multimodal Analysis and Description;

KEYWORDS
3D scene generation, long short-term memory, CNN

ACM Reference Format:

1 INTRODUCTION

Recent resurgence of virtual reality (VR) and augmented reality (AR) are resulting in an increasing demand of 3D indoor scene data. It is therefore of great interest in the development of generation of 3D indoor scenes from large 2D image dataset. However, due to the significant loss in visual information by projection of 3D scene to 2D image, a computer program often faces great challenges in accurately generating 3D scene from 2D images.

Fundamental challenges of scene understanding are how objects that present in the scene can be robustly detected and categorized. A large number of literatures have been investigating approaches for scene classification [9, 25, 29, 38, 41, 44] and object detection [10, 30, 39, 46]. On the top of regular object detection approaches which return 2D bounding boxes of detection results, some more advanced approaches extend the widely popularized deformable part model [10] to perform 3D object detection on monocular imagery by producing detection results as 3D cuboids [11, 20]. More recently, inspired by the success of depth cameras, an increasing number of approaches [15, 24, 37] incorporate depth information with appearance information for 3D object detection, and have demonstrated improved performance.

While ConvNet has, to a significant extend, improved the performance of various scene understanding tasks through building an end-to-end learning framework that learns from image pixels to ground-truth labels, it is incapable of capturing object-level spatial structure within scene images. Such spatial structure information is in fact very important when analyzing a scene. For example, given an image of a bedroom, when a bed is detected by an object detector, there are higher probabilities that a nightstand being detected at neighboring locations of the bed. In this work, we mainly focus on addressing two questions: 1) how can a scene understanding system fully exploit the objects’ spatial dependencies and indoor scene geometrical priors while robustly addressing occlusion, clutter as well as objects appearance variations and scene structural variations? and 2) how can a scene understanding system effectively extend the knowledge learned from the 2D space to the 3D space, and automatically design a virtual 3D scene that agrees with the layout of the monocular imagery? Correspondingly, the main efforts align with above questions, and include: 1) the development of a Conv LSTM network that extracts representative object and scene features and captures the spatial dependencies between objects as well as the relationship between objects and the scene, and 2) the development of a 3D scene synthesis mechanism that fully utilizes knowledge inferred by two convolutional neural networks as priors, and automatically designs virtual 3D scenes which agree with the scene layout and object arrangement of the input monocular RGB imagery. To achieve above goals, we propose to learn a convolutional long short-term memory (Conv LSTM) network from purely monocular RGB scene images for estimating 2D object locations and dimensions, and we further build a regression ConvNet for estimating objects’ degrees of freedom (Dof) information. Conv LSTM and regression ConvNet are trained separately following an end-to-end learning strategy, where the former maps pixels of object segments within scene images to both semantic object labels and scene labels, while capturing the inter-object and object-scene
relations using the recurrent unit, and the latter maps pixels of object segments to parametrized object poses, position and dimension information so as to provide 3D inferences in continuous forms. By enforcing some reasonable geometric constrained factors on objects’ DoF, estimations obtained from Conv LSTM and regression ConvNet can jointly provide 3D inferences to an object given only monocular RGB queries. We conclude the main contributions of this work as:

- We propose a framework that learns to provide indoor scene predictions on various tasks, and eventually synthesize 3D scenes from monocular indoor scene images.
- We exploit the benefit of image segmentation, and propose an end-to-end learning strategy that maps pixels of object segment regions to inter-object relations, object-scene relations and semantic labels of both object and scenes by adding recurrent units to ConvNets. We demonstrate that the proposed Conv LSTM network outperforms CNN on various tasks.
- We propose a regression ConvNet that learns from object segment regions and 3D pose and position parameters so as to provide continuous parametrized estimations in 3D space. Also, the regression ConvNet can incorporate with Conv LSTM to jointly infer a 3D space that describes a monocular RGB image.
- We provide extensive evaluations on the NYU-v2 indoor scene dataset, and we achieve state-of-the-art performance on various indoor scene understanding tasks while without requiring any depth information in test images.

2 RELATED WORKS

2.1 Indoor scene understanding

Indoor scene understanding essentially deals with various visual categorization tasks within a more constrained environment (e.g., a limited number of object categories and restricted space dimensions). Among the large body of works that have been proposed to address indoor scene understanding problems, bottom-up approaches [18, 28, 31, 34, 45] that utilize over-segmented image regions for the inferences of semantic object and scene labels have been widely popularized. In order to effectively exploit the contextual information within scene images, many approaches build graphical models such as Markov Random Field (MRF) [31] and Conditional Random Field (CRF) [28] on superpixels or image segments to perform inferences. With the availability of depth data in indoor scene datasets [34, 35], the majority of recent approaches encode the depth channel with horizontal disparity, height above ground and the angle of the pixel’s local surface normal (HHA), and demonstrate significantly improved performance [19] when combine HHA and RGB channels than using RGB images only. In the last a few years, there have been some attempts of extending the 2D object detection task into a 3D object detection task in indoor scenes. Song and Xiao [36] train exemplar SVMs and use the 3D cuboid sliding approach to search for positive 3D detections. Chen et al. [7] attempt to generate 3D bounding boxes for vehicle detection using monocular image.

2.2 Deep neural networks

Deep learning techniques have been extensively applied to many tasks in various research fields, e.g., computer vision and natural language processing. Except for the widely known fact that CNN [2] is extraordinarily effective when dealing with image classification or similar tasks, recurrent neural networks (RNN) are becoming increasingly popular when dealing with visual categorization tasks. Unlike the feed-forward architectures that are used in neural networks such as CNN, RNN allows cycled signal flows within the network by building internal states. While RNN is originally specialized in learning sequential data in tasks such as speech classification [16] and caption generation [21], some recent works expand applications of RNN to scene understanding tasks. Zuo et al. [47] propose the convolutional recurrent neural networks for learning contextual dependencies between different image regions. Intuitively, our work shares similar motivations with [47], however, we differ from [47] in various aspects: 1) instead of converting images into 3D pixel sequences, our approach learns from object segment proposals that contain more informative object-level knowledge; and 2) we utilize a LSTM network architecture that has better capability of capturing the mutual relationship between all objects within a scene than regular CNN architectures. Another recent RNN-based scene labeling method [33] attempts to build directed acyclic graph (DAG) RNN to capture the low-level spatial relationship between neighboring scene image units, where DAG is utilized to guarantee spatial neighbor image units stay close to each other in the chain data structure of RNN. While regular RNN is only capable for capturing short-term relations between inputs at consecutive entries, the LSTM network can capture both long-term and short-term dependencies, where the latter has recently been applied to action recognition [8] and scene labelling tasks [3]. Zhen et al. [27] address the indoor scene labeling problem following the idea of ReNet [40], which replaces the canonical convolution+pooling layer in CNN with RNN that sweeps across the image in both horizontal and vertical directions. Zhen et al. [27] improves over ReNet and utilizes LSTM to perform such an operation, and further designs a memorized fusion layer to unify information in both RGB and depth channels. Since the LSTM network in our work is built on CNN with end-to-end optimizable learning structure, it naturally inherits better representative and discriminative power than [3] when dealing with scene understanding tasks.

2.3 3D scene synthesis

To the best of our knowledge, the existing number of work with the effort on synthesizing virtual 3D spaces is limited, where a notable approach is [22], which employs a progressive strategy that inserts a single or multiple objects at each time while accounting both co-occurrence and arrangement models to provide controllability of the synthesized 3D scene. While Kermani et al. [22] utilizes manually-engineered representations to construct scene graphs that describe objects’ spatial dependencies and object-scene relationship, our approach instead employs a deep convolutional recurrent neural network to automatically capture such context information between objects and scenes within monocular images. An earlier best know work on synthesizing 3D indoor scenes is [12], which takes user-supplied examples as inputs and generates new

We provide extensive evaluations on the NYU-v2 indoor scene dataset, and we achieve state-of-the-art performance on various indoor scene understanding tasks while without requiring any depth information in test images.
Learning to Synthesize 3D Indoor Scenes from Monocular Images

Figure 1: Overview of our approach: Annotations on ground-truth object regions and scene images are used for training the Conv LSTM network and regression ConvNet. The top learning stream demonstrates the Conv LSTM framework, which seamlessly connects CNN with two LSTM modules, where we compute the Softmax output of each LSTM hidden layer values to obtain the semantic scene label loss and the semantic object label loss respectively. The regression ConvNet (shown in the bottom stream) is trained using the geometric loss, which measures the correctness between ground-truth Dof values and regression values obtained by the network. Conv LSTM and regression ConvNet jointly provide inferences for objects’ poses, positions and dimensions in the 3D space, and can eventually generate a 3D scene that agrees with the floor plan of the query monocular image.

scenes with diversified object occurrences and layout arrangements using a larger 3D scene database.

Another relevant work can be found in [17], where attempts are made to align 3D models to RGB-D images through training a classification ConvNet that matches surface normal images to rendered poses of 3D models. However, in this work, we are dealing a more challenging scenario since the depth channel (and also the surface normals) is not available in the testing scene images. Moreover, instead of using a classification ConvNet to generate discrete predictions, we train a regression ConvNet that gives continuous object pose inferences.

### 3 ConvNets Learning

We propose to train two ConvNets, the Conv LSTM networks and the regression ConvNets. Our goal is to employ the trained ConvNets to provide inferences on multiple indoor scene understanding tasks, including scene/object classification, object detection and object Dof estimation. Benefiting from ground-truth object segment annotations, the Conv LSTM network seamlessly integrates CNN to a recurrent neural network with a LSTM structure, where CNN takes pixels of image regions as inputs at a low level, and passes high-dimensional vectors that represent either an object within a scene image or a holistic scene image to the LSTM module. As an improvement over a pure CNN model, the inter-object spatial context and object-scene dependencies can be captured with the existence of the memory unit of LSTM. Both semantic object label loss and semantic scene label loss that measure the label consistencies of both scenes and objects are utilized for governing the optimization of Conv LSTM. On the other hand, the 3D geometric loss is utilized for training the regression ConvNet, which maps pixels from ground-truth object regions to continuous values that describe an object’s pose, position and dimension within the 3D space. The overview of both ConvNets are shown in Figure 1.

#### 3.1 Conv LSTM

Let \( G = \{G_1, G_2, \cdots, G_K\} \) be the \( K \) ground-truth objects within a scene image \( I \). we follow the typical region-based CNN (R-CNN) [14] manner and feed both the ground-truth segments \( G \) and the holistic scene image \( I \) into the CNN module, where each holistic scene image is simply considered as an object region. We utilize the CNN architecture which follows AlexNet [23], and the pre-trained (on ILSVRC-2012 [32] dataset) network weights up to the 7-th fully connected layer are used as initial weights for training our model. When directly fine-tuning CNN with ground-truth objects and scenes, the network is capable of providing segment-level inferences to query object proposals within a scene image, however, it is incapable of capturing the intra-object relation or object-scene relation. Thus, we build a LSTM model on the top of ground-truth object segments so that the objects’ spatial context along with object-scene dependencies can be learned.

In a general form of R-CNN, we denote features of a set of object regions as \( X = \{x_1, x_2, \cdots, x_f\} \), where \( x_i = f_{\text{CNN}}(G_i) \) represents
As illustrated in Figure 1, objects within the same scene image are fed into the same unit of LSTM at different “time-step” entries. Benefiting from the internal state module of LSTM, spatial dependencies between objects as well as dependencies between objects and the scene category can be fully explored when optimizing the network.

The data that are utilized to train RNN (including LSTM) are generally presented in sequential forms, however, objects’ spatial dependencies do not naturally possess the sequential knowledge. For example, as shown in Figure 1, when a “night stand” and a “bed” both exist in a scene image, we are expecting equally mutual dependencies between the “night stand” and the “bed”, since the existence of either object can increase the probability of the counter part’s existence within the same scene. Thus, simply formulating spatial object regions in a “sequential” form can result in unbalanced dependencies between objects. Instead, when learning Conv LSTM we formulate object segments in multiple sequential orders for each scene image by feeding objects to Conv LSTM in random input orders. As a result, we can expect multiple training “sequences” that represent an identical scene image, where objects contained in each “sequence” are placed with random permutation in the training set.

In the testing stage, a set of segment proposals are generated following the same strategy as in the training stage. These segment proposals along with the scene image are fed into the Conv LSTM network. The scene category can be determined based on the scene image I by computing the probability \( P(\hat{y} = c(I)) \) that the scene belongs to category \( c \) through the output of the Softmax layer. Similarly, object detection can be achieved by simply ranking segment proposals’ output scores.

### 3.2 Regression ConvNet

Six degrees of freedom (6DoF) is commonly used to indicate the movement of a rigid object in 3-dimensional space. When dealing with the problem of placing an object in the 3D space, we need to consider the dimensions of an object in addition to its 6DoF. Thus, we parametrize the degrees of freedom for each object as \((p_x, p_y, p_z, d_x, d_y, d_z, roll, pitch, yaw)\), where the first three variables indicate object’s 3D translations from the zero point, the middle three variables are object’s 3D dimensions, and the last three variables indicate objects’ rotations against each axis. We apply some reasonable constraints to simplify the problem. Specifically, we restrict the candidate object to be picked from a reduced number of object categories, such as bed and table, which can only be placed on the floor \( p_z = 0 \). Intuitively, objects from these categories are unlikely to rotate against certain axis, so that we only allow the object to rotate against z-axis (yaw), and correspondingly ignore the roll and pitch values. As a result, the remaining parameters that need to be estimated for determining an objects’ placement into a 3D space form a 6-dimensional vector \((p_x, p_y, d_x, d_y, d_z, yaw)\), among which \( p_x, d_x \) and \( d_y \) can be inferred from 2D object detection results if we assume the image plane is parallel to the x-z plane (as shown in Figure 1). Thus, we only need to estimate \((p_y, d_z, yaw)\) with the regression ConvNet.

We train the regression ConvNet using annotations on both ground-truth object segments as well as segment proposals obtained from the unsupervised object segmentation method constrained.
parametric min-cut (CPMC) [5], where for the latter, corresponding depth channels are used to compute the annotations in the training stage [28]. In the testing stage, only RGB channels of object segment proposals are required as inputs to the regression ConvNet for pose and position estimation. Empirically, the commonly used least square loss can be highly sensitive to outliers, and in our case outliers can be easily observed, so we instead choose to the robust square loss for training the regression ConvNet:

$$
\mathcal{L}_{\text{pose}}(q_i, \hat{q}_i) = \begin{cases} 
  e & \text{if } e \leq 1 \\
  1 + \log e & \text{if } e > 1,
\end{cases}
$$

(6)

where the error $e$ is the $L_2$-distance $\|q_i - \hat{q}_i\|_2$ between the ground-truth annotations $q_i$ and the estimated pose and position variables $\hat{q}_i$ of object $i$. Since the relations between objects’ poses and positions are not directly helpful for estimating objects poses, we do not consider the recurrent unit when building the regression ConvNet. We build the regression ConvNet by applying some modifications to the AlexNet architecture, and the resulting ConvNet follows the stream: $C(11, 96, 4) \rightarrow \text{ReLU} \rightarrow P(2, 2) \rightarrow C(5, 256, 1) \rightarrow \text{ReLU} \rightarrow C(3, 384, 1) \rightarrow P(2, 2) \rightarrow F(512) \rightarrow \text{ReLU} \rightarrow F(128) \rightarrow \text{ReLU}$. Experimental results suggest fine-tuning a pre-trained ConvNet usually leads to a higher loss when training the regression ConvNet, so that we choose to train the ConvNet from scratch using random initial weights. We start with an initial learning rate of 0.01 and decay the learning rate by 0.96 after every 2K iterations. We stop training at 20K iterations.

4 INFERENCE

The inference stage of our approach starts with obtaining a set of figure-ground object segments from each scene image. We employ the CPMC [5] to test monocular images for obtaining independent figure-ground overlapping image partitions by solving a sequence of constrained parametric min-cut problems, while not requiring any prior information on objects’ ground-truth labels or locations. We define $\mathcal{S} = \{S_1, S_2, \ldots, S_T\}$ as $T$ object segment proposals that are generated from an image. The candidate image segment proposals can easily fit the R-CNN framework, and use the network’s fully-connected layer output for training object detectors. On the other hand, the problem becomes more complicated when utilizing the recurrent unit that relies on multiple object regions. In our implementation, we define that Conv LSTM handles a fixed number of $K$ object segments in both training and testing stages. Ideally, when extracting the feature of a segment $S_i$ from the Conv LSTM, we expect the remaining $K-1$ segment proposals that jointly affect the representation of $S_i$ to be contextually meaningful.

With efforts toward this end, we apply a greedy approach (similar as the segment aggregation method presented in [4]) that iteratively selects $K-1$ most “salient” segments from each testing scene image in an unsupervised manner. Given the set of segments in each test scene image, by filtering out small segments and sampling from the remaining, we allow up to 200 figure-ground segments in each input image. Based on all these segments, a graph $G = (\mathcal{V}, E)$ that models pairwise relations between the vertices (segments) $v \in \mathcal{V}$ with edges $e \in E$, where the weight $w_{ij}$ assigned to each edge $e_{ij}$ is measured by the chi-squared distance $\exp(-\gamma \chi^2(q_i, q_j))$.

We model the problem of selecting the representative and compact segment set among all the segments in an image as the facility location problem [13, 26]. It can be considered as the set of facilities for opening facilities. With the constraint $K - 1$, the combinatorial formulation of the facility location problem can be applied:

$$
\max_{\mathcal{P} \subseteq \mathcal{V}} \mathcal{H}(\mathcal{P}) = \sum_{i\in \mathcal{V}} \max_{j\in \mathcal{P}} w_{ij} - \sum_{j\in \mathcal{P}} \phi_j \\
\text{s.t. } \mathcal{P} \subseteq \mathcal{S} \subseteq \mathcal{V}, N_{\mathcal{P}} \leq K - 1
$$

(7)

where $w_{ij}$ denotes the chi-squared distance between a group element $v_i$ (considered as clients) and a potential group center vertex $v_j$ (considered as facilities), and the cost $\phi_j$ of opening a facility is fixed to $\delta$. Submodularity of the overall profit $\mathcal{H}$ has been proved in [13, 26].

When dealing with the object detection task, the representation of each candidate object segment depends on the $K-1$ selected object regions. For extracting the representation for each object segment, we place the query segment at the last sequential order when feeding the set of segments into the Conv LSTM network. For segments that are selected as one of the $K-1$ segments, we simply ignore their existence in earlier entries of the sequential data, and follow the same procedure for extraction their representations.

5 EXPERIMENTS

5.1 Dataset and implementation details

We conduct both quantitative and qualitative experiments on various tasks using the NYU Depth V2 indoor dataset [34], which contains 1449 RGB-D images. We follow the settings in [28, 34] and utilize 795 scenes as the training set, and 645 scenes as the testing set. For training Conv LSTM, we use pixel-wise object labels and scene image labels as supervision, where all background regions are zero padded when inputting object regions to the network. In order to train the regression ConvNet, we use the parametrized object pose annotations [28], which contains three dimensions of objects’ 3D bounding boxes, 3D coordinates of bounding box centers and yaw values that indicate objects’ rotation angles against z-axis.

The training of both Conv LSTM and regression ConvNet is performed on a NVIDIA Tesla K80 GPU with dual 12-core Intel Xeon E5-2603 CPUs, and all deep ConvNets in our experiment are built using the TensorFlow library [1]. Fine-tuning Conv LSTM with pre-trained weights using a fixed learning rate of 0.001 and the batch size of 32² converges at 5K iterations, and takes 2 hours, and training regression ConvNets from scratch with an adaptive learning rate up to 20K iterations takes roughly 5 hours.

5.2 Scene and object classification

For the purpose of examining the effectiveness of the Conv LSTM network, we first apply the proposed method on both scene and object classification tasks. Our evaluations follow the “cleaned up” object and scene labels, which each has 21 and 27 categories respectively. Similarly as [28], we avoid the errors that can be potentially introduced from object detection results when dealing with scene
and object classification tasks, and directly extract features from ground-truth object regions. Note that for scene and object classification tasks, since we utilize the ground-truth object segment in both training and testing stages, the greedy segment selection procedure is not required. When computing scene representations using Conv LSTM, we use both object regions and the scene image, where the scene image is considered an additional object region and placed at the last "time-step" of the object-scene "sequence" for inputting to the network. We use a fixed number of K = 7 objects (including the scene image) for each scene, and only use the Softmax output that corresponds to the scene image as the category prediction, which is measured using accuracy. In the cases that a scene contains less than K − 1 = 6 objects, we randomly duplicate some objects; on the other hand, we randomly abandon some objects if the object number exceeds 7. A similar strategy is adopted in the object detection task, where each respective "time-step" Softmax output is considered as the object prediction. In addition to utilizing ground-truth object regions for the scene classification task, we also evaluate our approach with object segment proposals that are selected using the greedy approach.

The comparisons on scene classification and object recognition tasks between our approach and [28], the pre-trained CNN and fine-tuned CNN are shown in Table 1 and Table 2. While both RGB and depth images are used in [28], and only RGB images are used in our approach, performance improvement of 0.91% and 2.90% on scene and object classification tasks can still be observed. Results of CNN are obtained by directly employing the pre-trained weights on image regions or scene images for each task respectively, and the pre-trained network is further fine-tuned by object regions or scene images for each task separately. The effectiveness of Conv LSTM can be highlighted since in both scenarios the proposed Conv LSTM can output fine-tuned CNN.

Table 1: Scene classification results on NYU depth V2 dataset.

<table>
<thead>
<tr>
<th>Methods/Configuration</th>
<th>scene classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin et al. [28]</td>
<td>58.72</td>
</tr>
<tr>
<td>CNN [25]</td>
<td>53.87</td>
</tr>
<tr>
<td>CNN fine-tuned</td>
<td>56.15</td>
</tr>
<tr>
<td>Conv LSTM (salient segments)</td>
<td>57.87</td>
</tr>
<tr>
<td>Conv LSTM (ground-truth segments)</td>
<td>59.64</td>
</tr>
</tbody>
</table>

Table 2: Object classification results on NYU depth V2 dataset.

<table>
<thead>
<tr>
<th>Methods/Configuration</th>
<th>object classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin et al. [28]</td>
<td>60.49</td>
</tr>
<tr>
<td>CNN [25]</td>
<td>49.22</td>
</tr>
<tr>
<td>CNN fine-tuned</td>
<td>59.63</td>
</tr>
<tr>
<td>Conv LSTM (ground-truth segments)</td>
<td>62.53</td>
</tr>
</tbody>
</table>

5.3 Object detection

Detecting objects from cluttered scenes with occlusions is a long-term challenging problem. Our aim is to determine whether each candidate image segment contains the object of interest among all image segment proposals. Specifically, for each scene image we allow the CPMC [5] algorithm generates up to 200 object segment proposals, where each segment proposal is a binary mask in irregular shapes instead of rectangulars. We also follow the commonly used criterion and consider a object segment $S_i$ as recalled if its intersection-over-union (IoU) score $O(S_i, S_{gt})$ between $S_i$ and the ground-truth object region $S_{gt}$ is above 0.5, where $O(S_i, S_{gt})$ can be computed as:

$$O(S_i, S_{gt}) = \frac{|S_i \cap S_{gt}|}{|S_i \cup S_{gt}|}. \quad (8)$$

To be consistent with our training procedure, we first zero-pad each object image region with the binary mask before feeding these image regions to Conv LSTM. Instead of directly employing the Softmax layer output for determining the objection detection results, we extract the hidden layer values $h$ from the LSTM unit for all candidate object regions, and train a binary lib-svm [6] classifier for each object category using the hidden layer features. The hidden layer dimension $h$ of both LSTM modules are empirically set as 256. For obtaining object representations of testing samples, experimental results suggest that using multiple "sequences" of objects and extracting the last "time-step" LSTM hidden layer from each object "sequence" as the representation of a candidate object segment can lead to improved performance than using a single "sequence" of objects and extracting each corresponding "time-step" outputs as an object’s representation. Thus, for a scene image that contains $T$ segment proposals, we generate $T$ "sequences" of objects, where each "sequence" contains $K = 1$ segments and the segment of interest is placed at the last "time-step". In order to examine the performance of our method, we train 21 binary classifiers for all object categories, and show the detection results of each category in Table 5.3. We use the F1-score as measurement, and for a fair comparison with [28], we compute the F1-score when considering 15 highly ranked object segments within each scene image, where non-maximum suppression is performed on the pool of segment proposals to suppress overlapped proposals.
we define \{-x\} values for each parameter within the training samples, and we value of the 3D bounding box along \(z\).

## 5.4 Objects’ Dof estimation

For simplicity, we only consider 9 object categories with floor ground support, including *counter, toilet, bathtub, bed, table, sofa, chair, chest and fridge*. Since objects’ poses, positions and dimensions can be parametrized using 6 parameters under some constrained defined in Section 3.2, and 3 out of 6 parameters can be estimated through the 2D object detection task in Section 5.3 using only RGB images, we train a ConvNet for jointly estimating each of the 3 remaining pose parameters, including the dimension of the objects’ 3D bounding box along \(y\)-axis, the central point coordinate value of the 3D bounding box along \(y\)-axis and the \(yaw\) value. Here, we define \(y\)-axis to be perpendicular to the image plane, and the \(x-z\) plane to be parallel to the image plane (as shown in Figure1).

As a pre-processing step, we first compute the mean value of each regression parameter within the training samples, and let each parameter subtracts the mean value when training the regression ConvNets. In addition, we compute the minimum and maximum values for each parameter within the training samples, and apply these values as lower and upper bounds to estimated pose parameters so as to avoid estimations that have arbitrarily large absolute values. We illustrate some qualitative results in Table 3 and also show quantitative comparison between our regression ConvNet and directly fine-tuning using the AlexNet architecture in Table 5. The rotation range is defined between \(-180^\circ, 180^\circ\). The regression ConvNet shows consistent improvement over directly fine-tuning a CNN with the AlexNet architecture on all dimension, position and rotation variables.

### Table 5: Mean errors on objects’ Dof parameters.

<table>
<thead>
<tr>
<th>Methods/Parameter</th>
<th>dimension err</th>
<th>position err</th>
<th>rotation err</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN fine-tuned</td>
<td>0.78</td>
<td>1.06</td>
<td>94.1°</td>
</tr>
<tr>
<td>Regression ConvNet</td>
<td>0.46</td>
<td>0.77</td>
<td>67.3°</td>
</tr>
</tbody>
</table>

## 5.5 3D scene synthesis

In order to generate synthesize 3D scenes based on the object location and pose parameters inferred by Conv LSTM and regression ConvNet, we utilize 3D models from a subset of the Model dataset [43], which contains semantically identical object categories including *toilet, bathtub, bed, table, sofa, chair, chest and fridge*.

We provide qualitative illustrations of synthesized 3D scenes in Figure 2, where these 3D scenes are built using the Three.js JavaScript library based on WebGL. The top row contains some indoor scenes with both ground-truth and estimated bounding boxes, and sub-figures in the bottom row are synthesized 3D scenes based on inferred 6Dof information.

## 6 CONCLUSION

In this work, we proposed to train a Conv LSTM network and a regression ConvNet to deal with various indoor scene understanding problems. Benefiting from object segmentation, the Conv LSTM learns inter-object spatial context and object-scene dependencies with the recurrent unit using both semantic object label loss and semantic scene label loss, and the regression ConvNet is learned by mapping local object patches within scene images to parametrized object pose, position and dimension variables so as to provide continuous-form 3D inferences. Experiments on NYU-v2 dataset demonstrate the effectiveness of introducing the LSTM recurrent unit into a pure ConvNet framework by showing consistent improvements over directly fine-tuned CNN. Also, we demonstrate that training the regression ConvNet from scratch can achieve significantly less error rate than the fine-tuned CNN approach. In addition to achieving state-of-the-art performance on object/scene classification, object detection and object Dof estimation tasks while without requiring any depth information in the test stage, we further demonstrate qualitative results of synthesized 3D scenes that agree with the inferred room floor plan based on monocular indoor scene image.
REFERENCES


