Automatic Annotation for Semantic Segmentation in Indoor Scenes

Md Alimoor Reza[1], Akshay U. Naik[1], Kai Chen[2], and David J. Crandall[1]

***Abstract*— Domestic robots could eventually transform our lives, but safely operating in home environments requires a rich understanding of indoor scenes. Learning-based techniques for scene segmentation require large-scale, pixel-level annotations, which are laborious and expensive to collect. We propose an automatic method for pixel-wise semantic annotation of video sequences, that gathers cues from object detectors and indoor 3D room-layout estimation and then annotates all the image pixels in an energy minimization framework. Extensive experiments on a publicly available video dataset (SUN3D) evaluate the approach and demonstrate its effectiveness.**

I. Introduction

Service robots may soon be ubiquitous: according to a report from the International Federation of Robotics (IFR), there could be an estimated 31 million domestic robots deployed around the world by the year 2019, with functionality ranging from helping in household chores, to providing entertainment, to assisting people with physical disabilities [1], [2]. Most of these service robots will stay in a relatively closed, indoor setting, without needing to go outside [3]. Understanding this indoor setting will therefore play a critical role in the reliable functionality of these robots.

An important ingredient for understanding indoor settings is semantic segmentation — simultaneously identifying meaningful pixel regions in an image and assigning object or material labels to each of them. Existing solutions for semantic segmentation are mostly dominated by Deep Neural Network (DNN) based approaches [4]–[6]. For better or worse, large quantities of training labels are required to learn the millions of parameters of a DNN, which raises the question of how to obtain those ground truth labels in the first place. Typically, these images must be manually annotated by humans, which is time-consuming and expensive.

In this paper, we explore the possibility of densely annotating the images in an indoor video without the need for human annotations at all. This work builds on previous approaches that have asked humans to label a few key frames, and then propagate these annotations across the entire video [7]–[9]. Instead of requiring human annotation, we rely on signals from an object detector [10] applied to various object categories (e.g., bed, tv, etc.). To account for the remaining regions that are not explained by the object detectors, we automatically estimate the 3D layout of the scene, which helps to identify background regions. We then introduce a novel energy minimization-based formulation for solving for dense pixel-level annotations over an entire video.

Fig. 1: We automatically annotate indoor scenes for training semantic segmentation models. Images (left) are automatically annotated (right) based on off-the-shelf object detectors and a 3D room layout estimator.

We make the following contributions:

- First, we propose a novel method to densely annotate pixels of an indoor scene for semantic segmentation. Our method combines masks from pre-trained object detectors with the estimated indoor scene layout to explain all the pixels in an image including the background. We formulate the pixel-level annotation in a Conditional Random Field (CRF) energy minimization framework, to use the regularities between successive video frames to produce a consistent annotation over the entire video.
- Second, our method serves as an alternative source for generating large quantities of automatically annotated labels for an entire video.
- Finally, we demonstrate that our automatic annotations can be used to train a data-hungry Deep Neural Network for semantic segmentation on SUN3D.

II. Related Work

Semantic Segmentation: Before the advent of Deep Convolutional Neural Networks (DCNNs), semantic segmentation was usually performed bottom-up using hand-engineered features [11]. Deep neural networks have since surpassed these earlier approaches in accuracy.One successful application of an end-to-end trainable convolutional network for semantic segmentation is the Fully Convolution Network (FCN) of Long *et al.* [4]. This idea was further refined by SegNet [5]. To mitigate the cost of pixel-level annotation, Dong *et al.* [12] recently proposed a few-shot semantic segmentation approach that learns a DCNN model from very few annotated images.

Annotation: Castrejon et al. [13] learned a Recurrent Neural Network (RNN) model that could predict the polygonal vertices encompassing an object inside a cropped RGB

[1]School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA. Email: {mdreza, aunaik, djcran}@iu.edu

[2]School of Computer Science, Fudan University, Shanghai, China. Email: kchen16@fudan.edu.cn

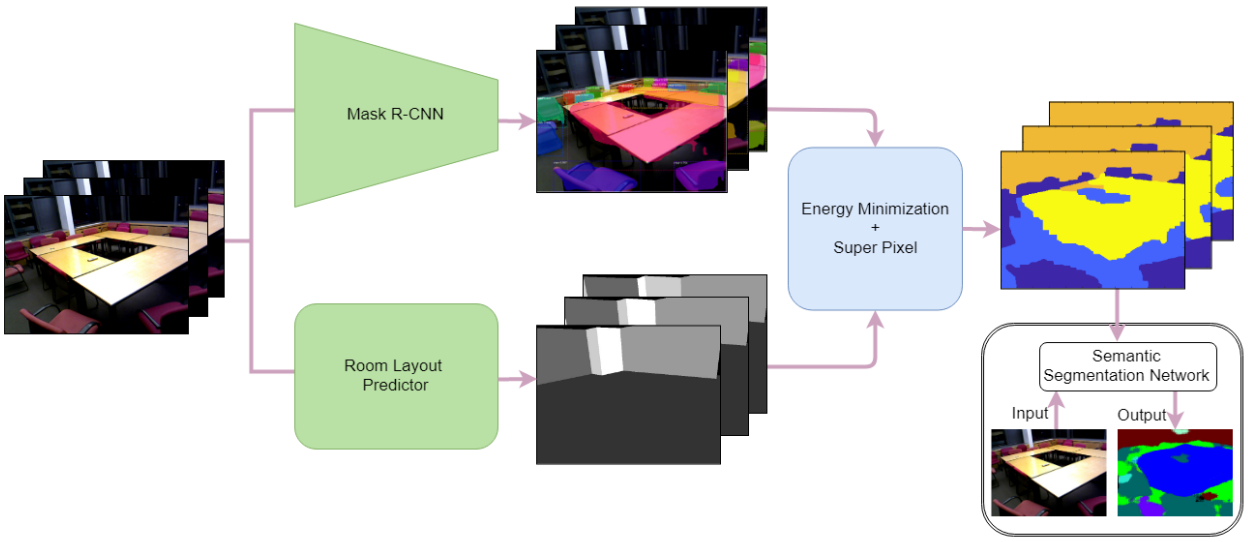

Fig. 2: For each video frame, we identify candidate object masks using pre-trained object detectors (top branch). The pixels not explained by the detector are estimated from 3d scene layout (bottom branch). This evidence is combined in an energy minimization framework to estimate our final annotation.

image. This method includes an interactive tool to correct prediction errors. EasyLabel [14] is a semi-automatic method for annotating objects on the RGB-D table-top setting. LabelFusion [15] is another semi-automatic method for generating large quantities of semantic labels from RGB-D videos. This method receives user annotation on the 3D reconstruction of the environment, which is then leveraged to propagate the labels across the frames in the RGB-D video. Unlike these methods, we propose a fully automatic method for labeling all pixels – covering a range of categories from small objects to large furniture and background – for all the images in an RGB-D video.

Video Label Propagation: Most semantic label propagation methods in a video start with manually annotations of a few keyframes, and then propagate those annotations across the remaining frames using cues such as spatial proximity, optical flow, or 3D reconstruction [7]–[9], [16], [17].

Object Detection: Detection tackles the problem of identifying objects in an image along with their locations (typically in the form of bounding boxes). SSD [19], YOLO [20], and Mask R-CNN [10] are popular choices. For example, Mask R-CNN [10] detects objects by first creating regions of interest, performing classification on each region, and then using per-class non-maximal suppression to avoid duplicate bounding boxes. Here we use Mask R-CNN, since it also provides segmentation masks for detected objects.

Indoor Scene 3D Layout Estimation: Mallya *et al.* [21] introduced the use of edge-based features to recover 3D layout, while Dasgupta *et al.* [22] use a Fully Convolutional Network along with a novel optimization framework. RoomNet *et al.* [23] proposed a faster DNN-based solution by estimating corner points of a layout along with classifying its layout-type from a fixed set of categories. These RGB only learning-based approaches rely heavily on the training data, and experimentally we found they perform poorly on images from a novel environment. This weakness could be addressed with the effective exploitation of the *depth*-channel in an RGB-D setting. Taylor *et al.* [18] proposed a layout estimation problem from an indoor RGB-D image. Their approach finds planar candidates directly by utilizing the depth data, then formulates layout estimation as a dynamic programming problem to find Manhattan structure.

Synthetic Data: A strategy for dealing with limited training data is to generate synthetic data [24], [25], but a caveat is that deep neural networks trained with synthetic data may not perform well when applied on real world images. A more effective approach may be to generate annotated data directly from natural images. In this paper, we address this problem and propose an automatic method for generating annotation from frames of video sequences.

III. Approach

We address the problem of automatically annotating all the pixels in an indoor image from a video sequence without any human annotation. Most pixels in an indoor scene belong to one of two broad categories: *object* or *background*. To automatically annotate all the pixels in an image, we need to find labels for these two different categories. Object detectors allow us to incorporate annotation information for the various specific *object* categories, such as "bed," "chair," "tv," etc. But a large fraction of the pixels in an indoor scene consist of *background* categories such as "wall," "ceiling," "floor," "window," etc. In order to annotate the pixels for these *background* categories not explained by an object detector, we resort to 3D layout estimation of the scene. Information from these two complementary sources is fused together by solving an energy minimization problem in a Conditional Random Field (CRF) framework. Figure 2 shows the pipeline

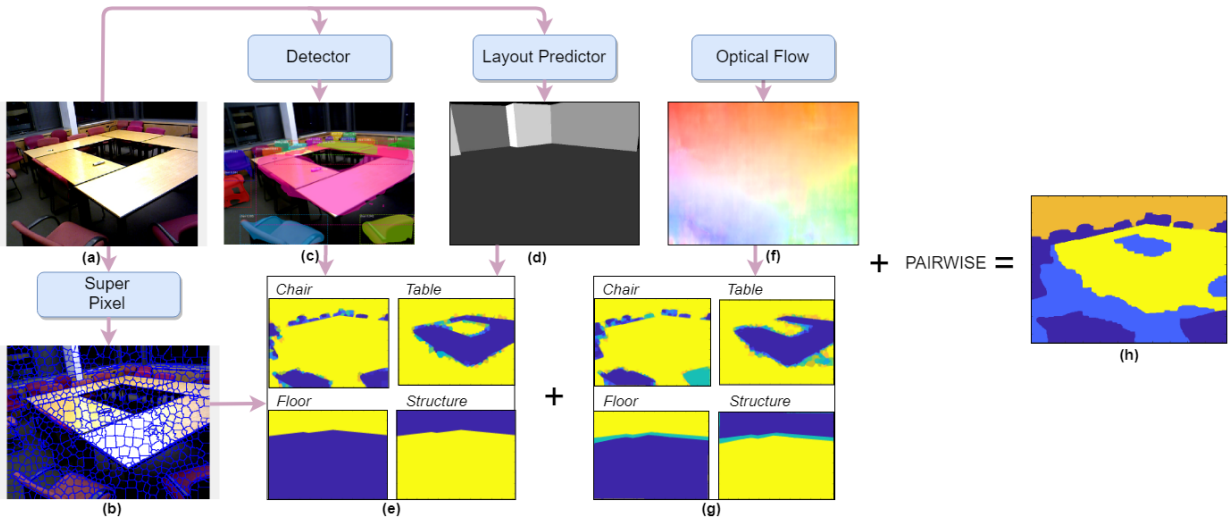

Fig. 3: Visualization of our energy minimization formulation. (a) For each frame, we (b) identify candidate object segmentation masks from pre-trained object detectors [10]. (d) The remaining pixels are estimated from the layout of the scene [18]. These are combined via energy minimization to estimate our final annotation (h). In addition to a unary term (e) from the current frame, we incorporate a second unary (g) that encodes evidence from previous frames, using optical flow as shown in (f).

of our methodology. We now describe these components in detail.

A. *Object Detection*

Object detection [19], [20] identifies the *objects* present in an image along with their locations in the form of rectangular bounding boxes. To find a coarse segmentation mask of each detected object, we use the object segmentation method of Mask-RCNN [10]. Figure 4 (top row) shows detection results on images from two different scenes in our experiments. Notice that while the object identifications and boundaries are generally accurate, a large fraction of pixels that are in the *background* are not labeled. We find the annotation information for these image pixels by estimating the structural layout of the scene.

B. *3D Scene Layout Estimation*

The approximate structure of a typical indoor scene consists of a set of 3D planes intersecting with each other. Individual components of these planar structures can typically be labeled as "wall," "floor," "ceiling," etc. Finding and identifying these planes is relatively straightforward if there are not many objects in the scene, but is much more difficult if the planes are occluded with multiple objects. Various methods have been proposed to solve the scene layout estimation problem [18], [21], [26]. After experimenting with various of these, we settled on the approach of Taylor *et al.* [18], which estimates the structure of the scene by first finding 3D planes utilizing the depth channel from an RGB-D image, and then assigns labels to each plane based on its estimated normal. The plane aligned to the gravity direction is labeled as "floor," the plane orthogonal to the "floor" is labeled "wall," and the remaining portion of the layout is labeled as "ceiling." Figure 4 shows the estimated scene layout components for two sample images.

Fig. 4: Sample detection and 3D room layout results from two different scenes: *Studyroom* (Left) and *MIT-32* (Right) from SUN3D [17]. Detector outputs (top) from Mask RCNN [10] provide an initial coarse segmentation around detected objects, while 3D layout estimation (below) explains background categories including "wall," "floor," and "ceiling."

C. *Superpixels*

An image superpixel is a set of contiguous pixels that share homogeneity in appearance, texture, etc. [27]–[29]. A superpixel generation algorithm partitions the image into a reduced number of segments, thereby speeding up the

work of subsequent processing which can process partitions instead of individual pixels. Reza *et al.* [7] generated high-quality superpixels, but relied on an expensive image-contour generation process that can take several minutes per image. In contrast, we follow a simpler and more efficient alternative, SLIC (Simple Linear Iterative Clustering) [28], which can generate superpixels in less than a second. Figure 3(b) shows superpixel boundaries overlaid on an image from our experiments. We use our superpixels as atomic units to incorporate annotation information from our two complementary sources of evidence, object detection and 3d scene layout estimation.

D. Pixelwise Annotation

We assume that we are given a video sequence consisting of frames $\{I_1, I_2, ..., I_N\}$. For a given unannotated frame I_k, we would like to minimize,

$$\begin{aligned} E(X_k|I_k, I_{k-1}, I_{k-2}, I_{k-3}) = &\sum_{i \in V} \theta_i(x_i; I_k) \\ &+ \sum_{i \in V} \phi_i(x_i; I_{k-1}, I_{k-2}, I_{k-3}) \\ &+ \sum_{(i,j) \in \zeta} \psi_{ij}(x_i, x_j; I_k), \end{aligned} \tag{1}$$

where $\theta_i(.)$ and $\phi_i(.)$ are the *unary* energy functions and $\psi_{ij}(.)$ is the *pairwise* function. The CRF graph $G = (V, \zeta)$ is defined over the pixels in the image I_k and 4-connected neighbors. We use the 3 frames immediately preceding I_k, namely I_{k-1}, I_{k-2}, and I_{k-3}, and exploit their unaries computed earlier by transferring them into the current frame using optical flow. This ensures temporal smoothness in finding the annotation for the current frame.

Unary Terms: From the detector output, we obtain a set of detected *object* masks along with their labels. For the *background* category, the predicted layout mask intersects with almost the entire image. We assign a fixed score to all the pixels that overlap with our various *background* categories (such as "wall," "floor," "ceiling," etc.). Figure 4 shows detection masks in different colors along with their label on the top-left corner of each bounding box. We find the intersection of a mask with a superpixel, and within each superpixel distribute the same score to all the pixels.

More specifically, we compute our first unary term,

$$\theta_i(x_i; I_k) = -f(x_i; I_k), \tag{2}$$

where $f(.)$ is a score for the pixel i computed by the superpixel that engulfs it. For each superpixel, we count the fraction of pixels that overlap with the detection mask of object a_j. As an example, if a detection mask from the "chair" category completely overlaps with a superpixel, then $f(.)$ assigns a score of 1.0 for "chair" category. Figure 3(b) shows an example of our unary energy term for different annotation categories.

Fig. 5: Detection masks on three successive frames in a video sequence. Notice that the detector fires inconsistently on the same instance of "chair" object category. Our formulation can handle this noise with a unary term $\phi(.)$ that encourages temporal consistency across frames.

Our second unary term is,

$$\phi_i(x_i; I_{k-1}, I_{k-2}, I_{k-3}) = -g(x_i; I_{k-1}, I_{k-2}, I_{k-3}), \tag{3}$$

where $g(.)$ is another scoring function based on the unary energy terms for the three frames immediately preceding frame I_k, in particular taking the average of the unary energy terms from the frames I_{k-1}, I_{k-2}, and I_{k-3} by transferring them into frame I_k using optical flow. Figure 5 shows a situation that demands this temporal consistency for finding the correct annotation.

Pairwise Term: To encourage smoothness, we adopted a simple Potts model for our pairwise energy function, which penalizes adjacent pixels having different annotations by a constant factor,

$$\psi_{ij}(x_i, x_j; I_k) = \begin{cases} 0, & x_i = x_j \\ b, & x_i \neq x_j \end{cases} \tag{4}$$

where b was empirically set to 0.5 for all our experiments.

Equation (1) is minimized using Graph Cuts [30] inference. A summary of the steps for finding the automatic annotation for an image is shown in Figure 3.

IV. Experiments

We experimented on the eight RGB-D video sequences from SUN3D [17] to validate our automatic annotation approach. Table II shows statistics for the eight video sequences. Each video consists of thousands of frames captured across various indoor locations in university campuses and dormitories. Only a fraction of these frames have been manually annotated. We validate the automatically generated annotations from our approach on these frames.

We used 10 categories, including both fine-grained (*Bed, Chair, Table, TV, Floor, Ceiling*) and generic categories (*Props, Furniture, Structure*). We conform to this selection based on the labeling criteria laid out by the popular indoor scene understanding dataset NYUD-V2 [31].

We used an open-source implementation of Mask-RCNN [32] pretrained on MS COCO [33] as our object detector. MS COCO consists of 80 categories commonly found in both indoor and outdoor scenes; we selected only the indoor object categories. We mapped categories of MS COCO to categories used in our experiments, as shown in Table IV.

For 3D scene layout estimation, we used the implementation by the author of [18]. The estimated layout provides

Video	Bed	Ceiling	Chair	Floor	Furniture	Props	Structure	Table	TV	Mean across category
hotel-umd	81.9 / 60.0	60.6 / 33.6	51.3 / 39.0	56.7 / 37.3	12.9 / 05.6	21.9 / 10.6	66.6 / 59.1	—	54.4 / 52.7	50.8 / 37.2
hv-c5	—	0 / 0	77.6 / 66.9	83.8 / 49.9	0 / 0	64.0 / 05.7	81.7 / 76.4	84.8 / 80.2	—	56.0 / 39.9
studyroom	—	0 / 0	74.8 / 64.8	74.2 / 59.6	36.0 / 31.0	23.9 / 07.5	87.5 / 70.9	48.0 / 45.4	—	49.2 / 39.9
mit-32	—	—	72.4 / 66.0	91.5 / 73.5	—	35.6 / 09.6	69.9 / 62.1	59.6 / 55.4	—	65.9 / 53.3
hv-c6	—	—	77.5 / 68.0	70.2 / 39.7	0 / 0	23.9 / 04.3	87.5/ 84.2	84.4 / 76.8	—	57.2 / 45.5
hv-c8	—	18.5 / 10.9	79.5 / 10.9	95.7 / 68.6	0 / 0	70.5 / 07.1	74.9 / 73.4	77.0 / 74.4	—	59.4 / 44.3
dorm	85.7 / 84.0	49.9 / 42.6	96.8 / 73.4	89.7 / 06.9	14.9 / 08.6	47.1 / 35.5	69.3 / 58.5	47.7 / 44.5	—	62.6 / 44.3
mit-lab	—	0 / 0	99.2 / 75.3	78.2 / 68.9	99.8 / 54.4	18.1 / 15.9	80.5 / 77.0	88.9 / 38.5	—	66.4 / 47.1
Mean across video sequence	**83.8 / 72**	21.5 / 14.5	**78.6 / 58.0**	**80 / 50.6**	23.4 / 14.2	38.1 / 12.0	**77.2 / 70.2**	**70.1 / 59.3**	54.4 / 52.7	—

TABLE I: Quantitative evaluation of our proposed automatic annotation method. First and second items in each entry denote evaluation metrics *average per-class* and *average IoU* respectively. The last column reports mean across the categories in each video (row wise). The bottom row shows the mean across video sequences for each category (column wise).

Scene	# Frames	# of Human-annotated Frames
hotel-umd	1869	82
hv-c5	2063	24
studyroom	3322	49
mit-32	5444	109
hv-c6	961	36
hv-c8	1003	23
dorm	2675	58
mit-lab	1906	14

TABLE II: Statistics of 8 video sequences in SUN3D [17].

a single mask for *floor* and *ceiling* categories, and the remaining layout is represented as series of other masks such as *Wall, Office-partition, Door,* etc. We map these categories to a generic *Structure* category as in NYUD-V2 [31].

To measure the performance of our automatic annotation, we used two metrics: *per-class accuracy:* for each class, find the proportion of correctly-labeled pixels, and *per-class IoU:* for each class, compute the ratio of the size of the intersection of ground truth label and estimated label regions, and the size of the union between the ground truth and estimated label.

A. Automatic Annotation Results

We validated the annotations generated automatically by our method against the ground truth labels manually prepared by a human in each video sequence. As the manual annotation is laborious and expensive, each video sequence has only a small fraction of the frames manually labeled (as shown in Table II). This is exactly the motivation for our work: we can generate automatic annotations for all the frames in a video sequence, allowing a larger quantity of annotations with minimal human effort.

The results of our evaluation using the two metrics defined above are shown in Table I. The table evaluates for each individual category as well as the average across categories (last column). To evaluate the category specific performance across all the videos, we also report an aggregated mean in the last row. Each entry in the table lists two numbers: i) *per-class accuracy* and ii) *per-class IoU* respectively. A missing entry signifies that that object is not present in that video (e.g., *TV* is present only in *hotel-umd*).

Our automatic annotation method performs well on object categories such as *Chair, Table,* and *Bed*, presumably because Mask-RCNN trained on MS COCO [33] has modeled these categories well. Some qualitative visualizations are shown in Figure 6. As we notice, our method can reliably annotate *chair, table, bed* categories in most cases. Our method had weaker performance on the generic object categories such as *Props* and *Furniture*. Our method solely relies on the signals from our object detectors to capture the annotation information; when a detector consistently fails to detect an object across a video sequence, our method fails to annotate that object. Our method also captures the annotations for *Floor* and Structure categories since our layout estimation can retrieve the structure of almost all of the scenes from the RGB-D images.

B. Semantic Segmentation with Automatic Annotations

Since our goal is to generate automatic annotations that would be useful for training deep semantic segmentation models, we evaluated our technique as a means of generating ground truth labels for FCN [4] for the 10 object categories mentioned above. We partitioned the 8 videos of SUN3D into 4 for training and 4 for testing. The training video sequences include *hotel-umd, hv-c5, studyroom, mit-32* which have a total of 264 human-annotated keyframes. Our test partition, *hv-c6, hv-c8, dorm, mit-lab*, has 131 human-annotated frames in total. We use all the 264 training frames along with their ground truth labels to train a FCN model, which we refer to as *GT*. We then automatically generated annotations for these frames using our method, and used them to train another FCN model, which we call *Auto*. Both models were trained for 60,000 iterations with learning rate $1e^{-}5$ and cross-entropy loss.

Quantitative results are shown in Table III (excluding *TV* which is absent in the test partition), with the first value in each entry indicating per-class accuracy and the second indicating IoU accuracy. Qualitative results are shown in Figure 7. The average per-class accuracy for *GT* is 56.4% and average IoU accuracy is 42.0%. The average per-class accuracy of the model *Auto* is 35.1% and average IoU accuracy is 20.1%. Of course, this is to be expected: *GT* was trained on laboriously hand-labeled training data, whereas *Auto* required

Train Set	Bed	Ceiling	Chair	Floor	Furniture	Props	Structure	Table	Average value
GT	39.3 / 33.6	68.5 / 54.7	73.3 / 40.2	66.3 / 39.4	16.0 / 10.5	15.4 / 12.0	84.2 / 71.8	88.3 / 74.2	56.4/42.0
Auto	1.2 / 1.0	11.3 / 9.5	53.1 / 21.8	72.0 / 15.7	0.9 / 0.7	4.0 / 2.9	70.8 / 53.9	67.6 / 55.3	35.1/20.1
GT + Auto-sample	15.1 / 13.4	33.8 / 30.7	**81.0** / 38.1	**73.3** / 22.9	10.5 / 7.0	6.4 / 5.4	78.3 / 61.9	**89.6** / 60.2	48.5/30.0

TABLE III: Semantic segmentation performance comparison. First and second items in each table entry denote metrics *average per-class* and *average IoU* respectively. Last column shows the aggregated performance across all 8 classes.

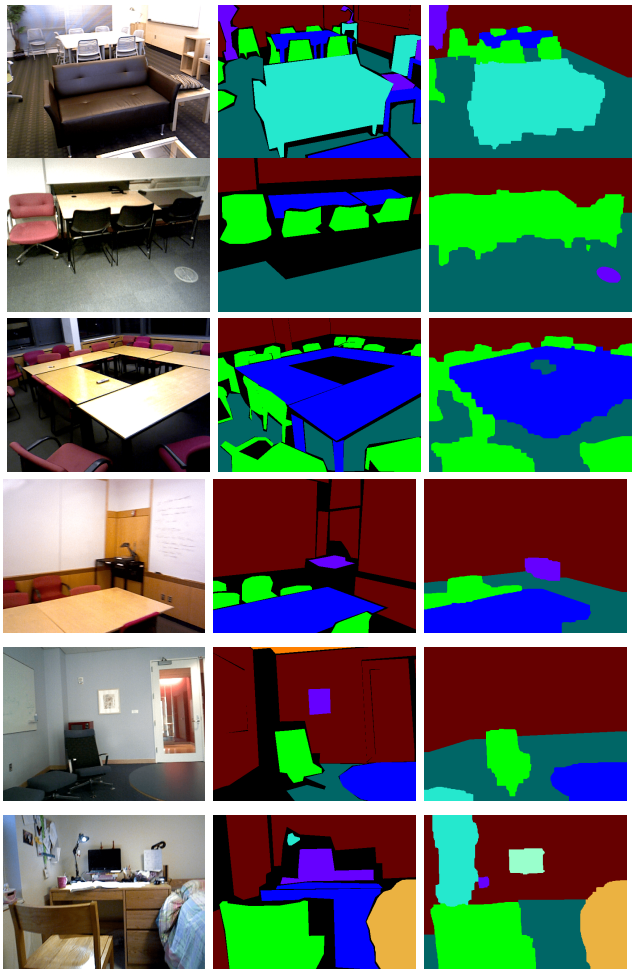

Fig. 6: Qualitative results for automatic annotation experiment on different video sequences from SUN3D [17]. From left to right we show the RGB image, the ground truth, and the automatic annotations from our method.

no human annotation whatsoever. *Auto* performs well on categories such as *chair, floor, structure* and *table*, although not as well as the *GT* model. Additionally, we observe that both models do not perform well on categories such as *furniture*, and *props*, as SUN3D has very few instances of these categories, making it difficult for the segmentation network to learn a reasonable model even with perfect ground truth annotations.

To further understand the effectiveness of our automatically generated annotation, we trained another model *GT+Auto-sample* (last row in Table III) by adding more samples of automatic annotated frames to the existing 264

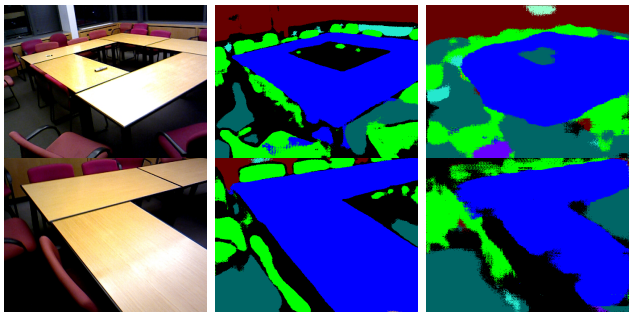

Fig. 7: Qualitative comparison for semantic segmentation on the images on test set (left) when trained on human annotated (middle) vs automatic annotated (right).

training human-annotated frames. More specifically, *Auto-sample* was prepared by sampling the automatic annotation of every 15-*th* image in each training video, resulting in a total 838 automatic annotated frames. Although the overall performance of *GT+Auto-sample* is inferior compared to *GT* (average per-class and IoU are 48.5% and 30.0% respectively), we observe performance improvements for some categories such as *Chair, Table*, and *Floor*. These three belong to the classes for which our automatic annotation method performed well (as reported in Table II and also discussed in Section IV-A).

V. CONCLUSION

In this work, we presented a method for generating annotations for indoor images from a video sequence without any human intervention. In order to produce the annotations automatically, our method relies on two complementary sources: object detectors and scene layout estimators. Our method offers an alternative source for generating a large quantity of dense pixel-level annotations. These annotations are effective for training a deep neural network for the semantic segmentation task. In the future, we plan to augment the method to generate annotations for large number of fine-grained indoor object categories.

VI. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (CAREER IIS-1253549). Kai Chen participated in this project while visiting Indiana University through the IU Global Talent Attraction Program (GTAP) program. All work was conducted while the authors were at Indiana University.

MS COCO	bed	dining table	chair	tv	oven	refrigerator	toilet	couch	bench	microwave	sink	book	remote	clock	bowl	wine glass	hair drier	fork	toothbrush	spoon	knife	tie	suitcase	laptop	mouse	keyboard	toaster	vase
Our Approach	bed	table	chair	tv	structure	furniture	furniture	furniture	furniture	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop
MS COCO	vase	cell phone	cup	handbag	bottle	cake	potted plant	pizza	person	scissors	sports ball	frisbee	umbrella	banana	apple	teddy bear	donut	skis	snowboard	kite	baseball bat	baseball glove	skateboard	tennis racket	sandwich	orange	broccoli	hot dog	
Our Approach	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	prop	

TABLE IV: Mapping of different MS COCO [33] categories to different indoor scene categories for our automatic annotation approach.

REFERENCES

[1] B. Gates, "A robot in every home," *Scientific American*, 2007.

[2] (2016). [Online]. Available: https://ifr.org/ifr-press-releases/news/31-million-robots-helping-in-households-worldwide-by-2019

[3] S. Song, L. Zhang, and J. Xiao, "Robot in a room: Toward perfect object recognition in closed environments," in *arXiv*, 2015.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encloder-decoder architecture for scene segmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[6] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[7] M. Reza, H. Zheng, G. Georgakis, and J. Kosecka, "Label propagation in RGB-D video," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[8] V. Badrinarayanan, F. Galasso, and R. Cipolla, "Label propagation in video sequences," in *CVPR*, 2010.

[9] S. Mustikovela, M. Yang, and C. Rother, "Can ground truth label propagation from video help semantic segmentation?" in *ECCV Workshop on Video Segmentation*, 2016.

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision*, 2017.

[11] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[12] N. Dong and E. Xing, "Few-shot semantic segmentation with prototype learning," in *British Machine Vision Conference*, 2018.

[13] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[14] M. Suchi, D. F. T. Patten, and M. Vincze, "Easylabel: A semi-automatic pixel-wise object annotation toolfor creating robotic rgb-d datasets," in *International Conference on Robotics and Automation (ICRA)*, 2019.

[15] P. Marion, P. Florence, L. Manuelli, and R. Tedrake, "Labelfusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes," in *International Conference on Robotics and Automation (ICRA)*, 2019.

[16] O. Miksik, D. Munoz, J. Bagnell, and M. Hebert, "Efficient temporal consistency for streaming video scene analysis," in *IEEE International Conference on Robotics and Automation*, 2013.

[17] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SfM and object labels," in *IEEE International Conference on Computer Vision*, 2013.

[18] C. Taylor and A. Cowley, "Parsing indoor scenes using RGB-D imagery," in *Robotics Science and Systems (RSS)*, 2012.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*, 2016.

[20] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv*, 2018.

[21] A. Mallya and S. Lazebnik, "Learning informative edge maps for indoor scene layout prediction," in *IEEE International Conference on Computer Vision*, 2015.

[22] S. Dasgupta, K. Fang, K. Chen, and S. Savarese, "DeLay: Robust spatial layout estimation for cluttered indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[23] C. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich, "Roomnet: End-to-end room layout estimation," *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[24] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[25] S. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European Conference on Computer Vision*, 2016.

[26] H. Su, C. Qi, Y. Li, and L. Guibas, "Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views," in *IEEE International Conference on Computer Vision*, 2015.

[27] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal on Computer Vision*, 2004.

[28] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and Sabine, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[29] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[30] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via Graph Cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[31] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *European Conference on Computer Vision*, 2012.

[32] W. Abdulla, "Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow," 2017.

[33] T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick, "Microsoft COCO: common objects in context," *arXiv*, 2014.