# SL3D - Single Look 3D Object Detection based on RGB-D Images

Gopi Krishna Erabati and Helder Araujo
Institute of Systems and Robotics, University of Coimbra, Portugal
gopi.erabati@uc.pt, helder@isr.uc.pt

***Abstract*—We present SL3D, Single Look 3D object detection approach to detect the 3D objects from the RGB-D image pair. The approach is a proposal free, single-stage 3D object detection method from RGB-D images by leveraging multi-scale feature fusion of RGB and depth feature maps, and multi-layer predictions. The method takes pair of RGB and depth images as an input and outputs predicted 3D bounding boxes. The neural network SL3D, comprises of two modules: multi-scale feature fusion and multi-layer prediction. The multi-scale feature fusion module fuses the multi-scale features from RGB and depth feature maps, which are later used by the multi-layer prediction module for 3D object detection. Each location of prediction layer is attached with a set of predefined 3D prior boxes to account for varying shapes of 3D objects. The output of the network regresses the predicted 3D bounding boxes as an offset to the set of 3D prior boxes and duplicate 3D bounding boxes are removed by applying 3D non-maximum suppression. The network is trained end-to-end on publicly available SUN RGB-D dataset. The SL3D approach with ResNeXt50 achieves 31.77 mAP on SUN RGB-D test dataset with an inference speed of approximately 4 fps, and with MobileNetV2, it achieves approximately 15 fps with a reduction of around 2 mAP. The quantitative results show that the proposed method achieves competitive performance to state-of-the-art methods on SUN RGB-D dataset with near real-time inference speed.**

***Index Terms*—Computer vision, Object detection, RGB-D, CNN**

## I. Introduction

In computer vision, object detection is a two fold process of simultaneously classifying the category of object and localising the object. Due to its significance in autonomous systems and wide range of applications, such as robotic vision, surveillance, face recognition and autonomous driving, it has attracted lot of attention both in research and industry.

Before the advent of deep neural networks, object detection was based on hand-crafted features (like SIFT [1], Haar-like [2], HOG [3] etc.) and shallow networks for classification. In recent times with the advent of Convolutional Neural Networks (CNNs) [4], large amount of data [5] and higher computational capabilities, the task of object detection is achieved with CNNs [6] producing higher accuracy than its predecessors with hand-crafted features. This is due to the fact that CNNs are able to learn low-level and high-level complex semantic features which helps in improving classification accuracy.

Region-CNN (RCNN) [6] is one of the first methods to approach the 2D object detection task with CNNs. The 2D object detection approaches provide 2D bounding boxes of the objects in image plane but they do not provide position, physical size and orientation of the object in real world which is useful for many real-time applications like object manipulation or autonomous driving.

The 3D object detection methods predicts 3D bounding boxes, which provides position, physical size and orientation of the object in the real world. With the advent of low-cost RGB-D sensors (like Microsoft Kinect, Asus Xtion), depth data is being used in many applications which provides information about object geometry (like shape) in addition to RGB data which provides appearance and texture information. Although there are many representations [7] to use the 3D data, we can classify in two types: 1) 2.5D representation [8]–[10] which uses depth data as an additional channel to RGB data and process the data with the conventional CNNs to predict the 3D bounding boxes, 2) 3D representation [11], [12] which converts the depth data into a point cloud using camera intrinsic parameters and later use 3D CNNs on voxels to predict 3D bounding boxes. The latter representation is computationally heavy and not suitable for real-time applications. So, we propose to use the former 2.5D representation to predict 3D bounding boxes.

The state-of-the-art methods [11]–[13] that use RGB-D image pair to predict 3D bounding boxes works on two-stages: to generate object proposals, and, to classify and regress the proposed 3D bounding boxes. This two-stage 3D object detection is computationally expensive and it is not suitable for real-time applications. Moreover, the 3D object proposal generation methods in these approaches [12], [13] are independent of the later detection network, which makes difficult for the end-to-end training of the network and thereby loses performance.

In order to tackle the above issues, we develop a proposal-free single-stage 3D object detection network (SL3D) that can be trained in an end-to-end setting to better optimize the model and increase the performance of the object detection task. The proposed method SL3D takes a pair of RGB-D images, looks once at the input images to extract feature maps and outputs 3D bounding boxes of the objects in near real-time. The network has two modules: multi-scale feature fusion and multi-layer predictions. The former module takes pair of RGB and depth images and extracts semantic features from the RGB and depth images using feature extraction networks like MobileNetV2 [14] or ResNeXt50 [15]. The multi-scale feature maps of the RGB and depth images are fused together with the help of fusion blocks. The latter module of SL3D adapts

multi-layer prediction architecture of SSD [16] to provide predictions from multiple feature maps. A set of 3D prior bounding boxes are attached to each of the prediction layers to discretize the search space. The physical size of these 3D prior boxes are empirically calculated from the training set and position of prior boxes are determined from depth information. The multi-scale prediction layers are convolved with a small $3\times3$ convolutional kernel to provide the position, size and orientation of the object as an offset to 3D prior boxes along with the object category.

Our approach is evaluated on SUN RGB-D dataset [17]. The quantitative results show that our approach SL3D outperform the state-of-the-art Deep Sliding Shape [12] and COG [18] methods both in accuracy and inference time. The main contributions of our approach are:

1) A proposal-free network SL3D, that looks at the RGB-D image pair only once to predict 3D bounding boxes for the task of 3D object detection. The network can be efficiently trained in an end-to-end fashion for better model optimization to leverage performance of the 3D object detection. The proposal-free network can run at near real-time inference speed.
2) We evaluate our model using two feature extractors (MobileNetV2 [14] and ResNeXt50 [15]) on the SUN RGB-D [17] benchmark 3D object detection dataset.

The paper is structured as follows: related work of object detection methods is given in Section II, architecture and methodology of our approach is discussed in Section III, training and implementation details are given in Section IV, experimental results of our approach are presented in Section V and finally conclusions are drawn in Section VI.

## II. Related Work

We briefly review the existing works on 2D object detection and 3D object detection from RGB-D images.

*1) 2D Object Detection:* The 2D object detection approaches aims at classifying and localizing the objects in the image with rectangular 2D bounding boxes. Upon the advent of neural networks [4], the task of object detection started to have realistic solutions [6]. The framework for 2D object detection can be classified into two types: 1) Two-stage methods with region proposal generation as former stage and classification as latter stage, and 2) Single-stage methods with unified architecture to classify and localize the objects. R-CNN [6] is one of the first 2D object detection networks to achieve state-of-the-art performance in accuracy and speed which is later improved by Fast-RCNN [19] and Faster-RCNN [20]. However, these networks are two-stage methods for 2D object detection that doesn't provide real-time inference speeds. To resolve the issue of real-time application, YOLO [21] is the first single-stage 2D object detection network that provided real-time inference speeds for 2D object detection, later improved by YOLOv2 [22], YOLOv3 [23], YOLOv4 [24] and YOLOv5 [25]. SSD [16] is a single-stage 2D object detection network that incorporates multi-layer predictions to improve the performance of the network on different object sizes. We adapt the multi-layer prediction network of SSD to our approach to increase the performance of 3D object detection.

*2) 3D Object Detection:* In computer vision applications like object detection, 3D data (RGB and depth images) is a valuable asset as it provides rich geometric information about the scene. The 3D object detection approaches aims at classifying and localizing the objects in the scene with 3D bounding boxes. There are various ways of representation of 3D data [7] both in 3D Euclidean data representation (with grid structure) and 3D Non-Euclidean data representation. The 3D Euclidean data representation includes RGB-D image representation as 2.5D data, projection of point clouds (converted from depth images) and discretization of point clouds as volumetric data. The 3D Non-Euclidean data representation includes point clouds, meshes and graphs.

Eitel *et al.* [26] proposed a RGB-D object recognition approach with late fusion of RGB and depth feature maps. The learning accuracy in this method is improved by effective depth encoding scheme and data augmentation techniques for robust learning. Gupta *et al.* [9] introduced a depth encoding scheme called HHA encoding, that encodes each pixel to height above ground, horizontal disparity and pixelwise angle between surface normal and direction of gravity. They use R-CNN [6] like architecture to detect the objects using fusion of RGB and depth feature maps. In the prior work, the researchers used different methods to fuse different modalities of data. *Early fusion* [27], *late fusion* [12] and *deep fusion* [28] are some of the methods to fuse the data from different modalities. Our fusion methodology incorporates multi-scale deep fusion for interaction between RGB and depth feature maps which helps our model to automatically learn the fusion strategy rather than training a linear classifier on top of feature extraction of RGB and depth feature maps. We also incorporate depth encoding scheme proposed in [26] which colorizes the depth data without much preprocessing and thereby improves the performance of our model. Liu *et al.* [29] adapted a brute-force sliding window approach in the point cloud data to generate object proposals and the proposals are projected onto the RGB and depth images separately whose features are extracted by two RCNNs. Lahoud *et al.* [13] uses a Faster-RCNN [20] network to detect the objects in 2D RGB image space, converts the detected 2D bounding box region of depth image into 3D point cloud and uses a multi-layer perceptron to regress the position and physical size of 3D bounding box.

The methods stated above use the RGB-D image pair directly as input data to the network. Some 3D object detection methods converts the depth data into the point cloud and use 3D ConvNets for better semantic feature representation of data. However, the 3D ConvNets are computationally very expensive when compared to 2D ConvNets. Song *et al.* [11] proposed *Sliding Shapes* method of 3D object detection. This method initially converts depth data into a point cloud and as name suggests, slides a 3D window in the point cloud as a brute-force object proposal approach. Later, handcrafted 3D features are used to represent the proposed object and fed to

SVMs for each object class. However, this method is quite slow because of repeatedly computing 3D features for each sliding window and applying the features to many SVMs. Instead of brute-force sliding window object proposal scheme, *Deep Sliding Shapes* [12] incorporates 3D ConvNets to generate object proposals by using 3D Region Proposal Network (3D RPN), inspired by 2D RPN from Faster-RCNN [20]. A 3D ConvNet was used for each 3D region proposal to classify and regress the 3D bounding box position and size. The use of 3D RPN had a significant improvement as compared to brute-force sliding window approach [11] in terms of accuracy and performance. Instead of computationally heavy 3D ConvNets, our model uses 2D ConvNets to extract features from RGB and depth images which are later deep fused in multi-scale scheme, followed by multi-layer prediction with a set of 3D prior boxes to predict the 3D bounding boxes.

## III. Methodology

### A. About the SL3D Network

The SL3D neural network architecture is shown in Fig. 1, it consists of two main modules: multi-scale feature fusion module and multi-layer prediction module. The multi-scale feature fusion module takes pair of RGB and depth images and extracts features using MobileNetV2 [14] or ResNeXt50 [15] feature extraction networks. The layers of RGB and depth feature maps at multiple scales are deep fused in the fusion block shown in Fig. 1. The multi-layer prediction module includes few extra feature layers which form a set of multiple prediction layers. Every location of each prediction layer is attached with a set of 3D prior boxes with manually defined physical size along with their physical 3D positions estimated from depth image. A classifier block with a small 3×3 convolutional kernel is attached to each prediction layer to estimate classification score and also to regress the 3D prior box position, size and orientation offsets. The outputs of classifier blocks of multi-layer prediction module are all concatenated to efficiently predict the 3D objects by applying 3D non-maximum suppression.

### B. Multi-Scale Feature Fusion

In our approach as shown in Fig. 1, we employ multi-scale *deep fusion* method to fuse the RGB and depth feature maps to leverage intermediate-level and high-level semantic data.

*1) ResNeXt50 Feature Fusion:* As shown in Fig. 1, RGB and depth input (300 × 300) image pair is passed through two separate feature blocks to extract RGB and depth feature maps using ResNeXt50 [15] feature extractor (Fig. 1 (middle row)). ResNeXt50 feature extractor consists of 5 main convolutional modules with convolutional blocks repeating 1, 3, 4, 6, 3 times for each of five main convolutional modules. The output from the RGB modality convolutional layers of ResNeXt50: *c3_b4_out_rgb* (38×38×512), *c4_b6_out_rgb* (19×19×1024) and *c5_b3_out_rgb* (10×10×2048), and depth modality convolutional layers of ResNeXt50: *c3_b4_out_depth* (38×38×512), *c4_b6_out_depth* (19×19×1024) and *c5_b3_out_depth* (10×10×2048) are passed to the fusion block. The fusion block as shown in Fig. 1 (middle row) consist of three fusion sub-blocks which takes feature maps from three multi-scale feature map layers from RGB and depth modality. Essentially, the fusion sub-block takes RGB and depth feature maps, evaluates element-wise mean, passes the output through two 1×1 convolutional layers and results the fused layer by computing element-wise mean of the two 1×1 convolutional layers. Multi-scale deep feature fusion strategy enable interactions among features from different modalities.

*2) MobileNetV2 Feature Fusion:* The feature fusion of MobileNetV2 [14] feature extractor (Fig. 1 (bottom row)) is similar to ResNeXt50 feature fusion described above, with a small change in the number of fused layers. MobileNetV2 feature extractor consist of 16 blocks of depthwise separable convolutional layers along with general convolutional layer at start and at the end. The output from the RGB modality convolutional layers of MobileNetV2: *b13_exp_relu_rgb* (19×19×576) and *out_relu_rgb* (10×10×1280), and depth modality convolutional layers of MobileNetV2: *b13_exp_relu_depth* (19×19×576) and *out_relu_depth* (10×10×1280) are passed to fusion block. Instead of three fusion sub-blocks as in ResNeXt50 feature fusion, we only have two fusion sub-blocks to fuse two multi-scale feature maps in a similar way explained above.

### C. Multi-Layer Prediction Module

The outputs from the fusion block of the feature extractor forms a part of multi-layer prediction module. In addition, we incorporate three extra feature layers to the network to extract more high-level semantic information from the images. Essentially, we add three blocks of convolutional layers (ConvA, ConvB, ConvC) as shown in Fig. 1. Each block has a 1×1 convolution followed by depthwise separable convolutional layer. We have 6 prediction layers (3 from fusion block and 3 extra feature layers) for ResNeX50 feature extractor and 5 prediction layers (2 from fusion block and 3 extra feature layers) for MobileNetV2 feature extraction network. Each location of all prediction layers is attached with a set of 3D prior boxes. A classifier block with a small 3×3 convolutional layer is applied on every prediction layer to classify the objects and regress the position, size and orientation offsets. The feature maps with high receptive field will be able to capture features of large objects, on the other hand feature maps with low receptive fields will be able to capture features of small objects. Therefore, similar to [16], we adapt multi-layer prediction paradigm to increase the performance of 3D object detection. The output of multi-layer prediction module is fed to 3D non-maximum suppression to predict final 3D objects.

*1) 3D Prior Boxes:* In order to reduce the output search space of 3D object detection, we discretize the output search space into a set of 3D prior boxes. The set of 3D prior boxes are added to each location of all prediction layers to indicate potential object candidates. The physical size of 3D prior boxes are obtained from the statistics of physical sizes of objects in the training dataset and the 3D location of prior box

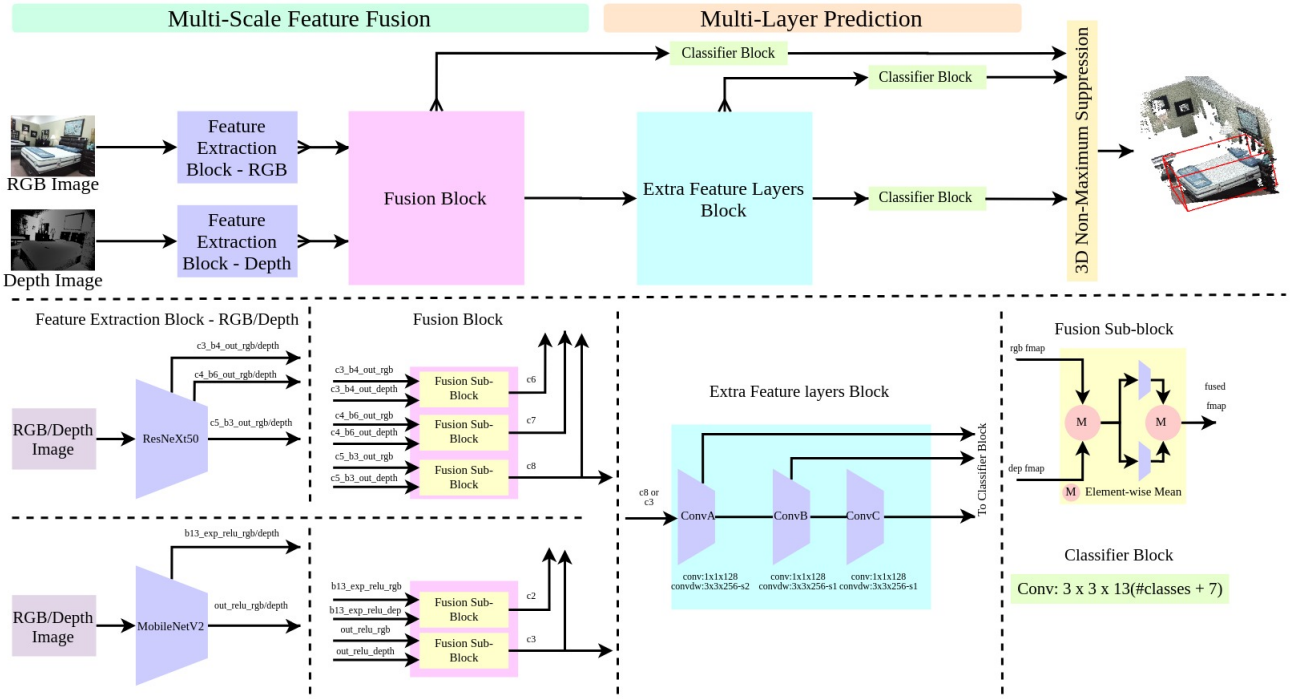


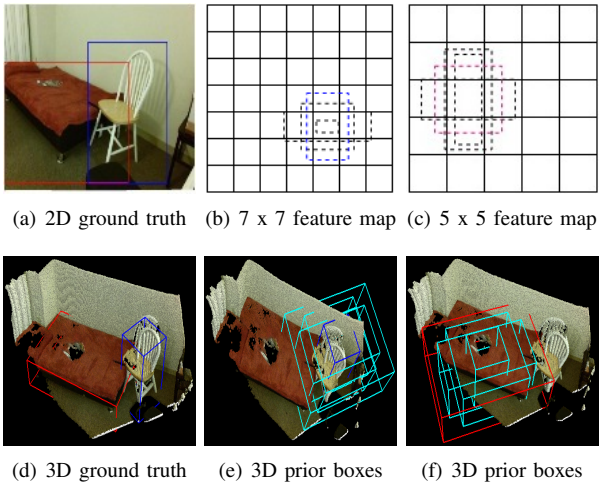

Fig. 1. SL3D Network Architecture

is evaluated from the depth image. Similar to [12], we adopt 13 different sizes of 3D anchor boxes for SUN RGB-D dataset [17]. We adopt the strategy presented in [8] to compute the position of 3D prior boxes by projecting 2D segment pixels of depth image into 3D space. Concretely, for each prediction layer of size $m \times n$, we divide the depth image into $m \times n$ grid cells. The $i^{th}$ row and $j^{th}$ column of the depth image grid cell corresponds to center $(i, j, z_{med})$ of grid cell or center $(c_x, c_y, z_{med})$ of the 3D prior box in the camera coordinate system. Since the depth images are usually noisy, we take median value $(z_{med})$ of the depth image grid cell for the sake of robustness. The 3D position of prior box $(x_{box}, y_{box}, z_{box})$ in world coordinate system can be found by projecting the center of grid cell $(c_x, c_y, z_{med})$ in the camera coordinate system to world coordinate system as given in (1).

$$\begin{aligned} x_{box} &= z_{med} * (c_x - o_x)/f \\ y_{box} &= z_{med} * (c_y - o_y)/f \\ z_{box} &= z_{med} \end{aligned} \tag{1}$$

where $f$ is the focal length and $(o_x, o_y)$ is the principal point of the camera.

A 3D box is represented by $(x_{box}, y_{box}, z_{box}, l, w, h, \theta)$, where $(x_{box}, y_{box}, z_{box})$ is the center position of 3D box, $(l, w, h)$ is the physical size of the 3D box in world coordinate system and $\theta$ is the orientation of the 3D box around its $z$ axis. The orientation angle $\theta$ for the 3D prior box is set to $0^\circ$ at the start of the experiment.



(a) 2D ground truth (b) 7 x 7 feature map (c) 5 x 5 feature map

(d) 3D ground truth (e) 3D prior boxes (f) 3D prior boxes

Fig. 2. Matching the 3D prior boxes with 3D ground truth boxes. (a) shows 2D ground truth boxes of bed (red) and chair (blue). (b) and (c) shows different scale feature maps along with 2D prior boxes matched with 2D ground truth boxes. This gives us the 2D location of possible match of 3D prior box with 3D ground truth box in feature map space. At this 2D location in the feature map, all the 3D prior boxes are aligned with 3D ground truth box and the 3D prior box with maximum IoU above a defined threshold is chosen as a positive match. (d) shows 3D ground truth boxes of bed (red) and chair (blue). (e) and (f) shows matched 3D prior boxes of chair and bed respectively.

## IV. Implementation Details

### A. Matching 3D Ground Truth Boxes

There are number of 3D prior boxes attached to the multiple prediction layers (25,220 for ResNeXt50 model and 6,448 for MobileNetV2 model). Therefore, we need to match the 3D

prior boxes to 3D ground truth boxes for positive training samples. We adopt to use 2D ground truth data of the corresponding 3D ground truth box for positive sample matching. In our approach as shown in Fig. 2, 2D prior boxes [16] are used to match with 2D ground truth box to determine the 2D location of potential 3D prior box match in the feature map space. As there is one-to-one mapping between 2D ground truth box and 3D ground truth box, we can identify matched 3D ground truth box. Once we obtain the potential 2D location of 3D prior box match in feature map space, we align the centers of all the 13 3D prior boxes to the center of 3D ground truth box because matching of prior and ground truth boxes is dependent on size similarity. We compute the 3D Intersection over Union (IoU) overlap between all 13 3D prior boxes and 3D ground truth box, and the 3D prior box with highest overlap and greater than a defined positive IoU threshold is chosen as positive match. All the remaining 3D prior boxes which are above a defined negative IoU limit and below positive IoU threshold are set as neutral because they are neither a positive match nor can be considered as background. All the 3D prior boxes which are below negative IoU limit are considered as background.

### B. Loss Function

The loss is a generalization of SSD loss [16] for 3D object detection. It is defined as a weighted sum of localization loss (loc) and classification loss (conf), given by (2).

$$L[x, k, p, g] = \frac{1}{N}[L_{conf}(x, k) + \alpha L_{loc}(x, p, g)] \qquad (2)$$

where $N$ is the number of matched prior boxes. $x_{ij}^{c} = [1, 0]$ is an indicator of matching $i^{th}$ 3D prior box with $j^{th}$ 3D ground truth box of class $c$.

The localization loss is Smooth L1 loss [19] between predicted 3D bounding box $(p)$ and 3D ground truth box $(g)$. Only positive matches are considered in the localization loss and negative matches are ignored. We regress offsets for the set $P = [x_{box}, y_{box}, z_{box}, l, w, h, \theta]$, where $l$ is length, $w$ is width, $h$ is height, $\theta$ is angle and $(x_{box}, y_{box}, z_{box})$ is the center of the 3D prior box $(d)$. The localization loss is given by (3).

$$L_{loc}(x, p, g) = \sum_{i \in Pos}^{N} \sum_{m \in P} x_{ij}^{c} smooth_{L1}(p_i^m - \hat{g}_j^m)$$

$$\begin{aligned}
\hat{g}_j^{xbox} &= \frac{(g_j^{xbox} - d_i^{xbox})}{d_i^l}; & \hat{g}_j^{l} &= \ln \frac{g_j^l}{d_i^l} \\
\hat{g}_j^{ybox} &= \frac{(g_j^{ybox} - d_i^{ybox})}{d_i^w}; & \hat{g}_j^{w} &= \ln \frac{g_j^w}{d_i^w} \\
\hat{g}_j^{zbox} &= \frac{(g_j^{zbox} - d_i^{zbox})}{d_i^h}; & \hat{g}_j^{h} &= \ln \frac{g_j^h}{d_i^h}
\end{aligned} \qquad (3)$$

The classification loss is the loss in class prediction. Both positive and negative matches are used to penalize the loss. For negative match prediction, the classification loss of background class is penalized. The confidence loss is softmax loss over multiple classes confidence $(k)$. The classification loss $L_{conf}$ is given by (4).

$$L_{conf}(x, k) = -\sum_{i \in Pos}^{N} x_{ij}^{c} \ln(k_i^c) - \sum_{i \in Neg} \ln(k_i^0) \qquad (4)$$

### C. Training Strategy

We follow a two-fold training strategy to train our model. Firstly, SL3D network is trained with same input (RGB or depth) to both of the feature extraction modalities. While training with RGB only images for both modality feature extraction networks, we still use depth images to generate 3D prior boxes. The feature extraction networks of both the modalities are initialized with weights from SSD [16] network trained for 2D object detection. We call these two models SL3D-RGB and SL3D-depth. Finally, we train SL3D model with RGB and depth image inputs for final 3D object detection. The weights of feature extraction network of RGB and depth modalities are initialized with weights from SL3D-RGB and SL3D-depth models respectively.

The SL3D model with MobileNetV2 feature extraction network is trained on SUN RGB-D dataset with SGD optimizer at a batch size of 8 and ResNeXt50 is trained with the same optimizer at a batch size of 4. The SL3D-RGB and SL3D-depth model is trained at a learning rate of $4.5 \times 10^{-4}$ for first 70K iterations, $4.5 \times 10^{-5}$ for next 15K iterations and $4.5 \times 10^{-6}$ for last 15K iterations. The SL3D model is trained with a learning rate of $1.5 \times 10^{-4}$ for 55K iterations and $1.5 \times 10^{-5}$ for 25K iterations.

During training, similar to SSD [16], hard negative mining method is applied to choose negative samples for the computation of training loss. The negative to positive sample ratio is kept at 3:1. We add horizontally flipped images to the training data and resize them to 300×300, no additional data is considered for training.

## V. Results and Discussion

The SL3D architecture is trained and tested on the SUN RGB-D dataset [17]. The dataset contains 5,285 RGB-D images for training and 5,050 RGB-D images for testing. The network is trained for 19 object classes. Similar to [11], 3D IoU metric is used for evaluation. The predicted 3D bounding box is considered as positive, if 3D IoU between predicted box and ground truth box is greater than 0.25. We compare accuracy in terms of mean Average Precision (mAP) on the test dataset, and inference time of the methods. We discuss the quantitative and qualitative results of our approach and present ablation studies.

### A. Quantitative Results

The evaluation of our approach on SUN RGB-D test dataset for 19 classes and 10 classes is given in Tab. I and Tab. II respectively. Our approach with ResNeXt50 feature extractor achieves 31.77 mAP with an inference time of 0.24s ($\sim$ 4

TABLE I
3D OBJECT DETECTION RESULTS FOR 19 CLASSES OF SUN RGB-D DATASET

| Method | bathtub | bed | bookshelf | box | chair | counter | desk | door | dresser | garbagebin | lamp | monitor | nightstand | pillow | sink | sofa | table | tv | toilet | mAP | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DSS [12] | 44.2 | 78.8 | 11.9 | 1.5 | 61.2 | 4.1 | 20.5 | 0.0 | 6.4 | 20.4 | 18.4 | 0.2 | 15.4 | 13.3 | 32.3 | 53.5 | 50.3 | 0.5 | 78.9 | 26.9 | 19.55 |
| Ours - ResNeXt50 | 51.6 | 76.2 | 26.2 | 4.5 | 55.7 | 6.9 | 21.2 | 0.9 | 26.3 | 25.4 | 17.2 | 8.7 | 47.2 | 12.7 | 38.9 | 53.8 | 41.2 | 6.5 | 82.6 | **31.77** | 0.24 |
| Ours - MobileNetV2 | 47.8 | 73.2 | 22.8 | 3.2 | 51.7 | 5.1 | 20.6 | 0.7 | 23.7 | 24.5 | 14.2 | 8.1 | 42.2 | 11.8 | 35.9 | 50.8 | 39.8 | 5.6 | 81.4 | 29.63 | **0.07** |

TABLE II
3D OBJECT DETECTION RESULTS FOR 10 CLASSES OF SUN RGB-D DATASET

| Method | bathtub | bed | bookshelf | chair | desk | dresser | nightstand | sofa | table | toilet | mAP | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COG [18] | 58.26 | 63.67 | 31.8 | 62.17 | 45.19 | 15.47 | 27.36 | 51.02 | 51.29 | 70.07 | 47.63 | 600 to 1800 |
| Ours - ResNeXt50 | 51.6 | 76.2 | 26.2 | 55.7 | 21.2 | 26.3 | 47.2 | 53.8 | 41.2 | 82.6 | **48.2** | 0.24 |
| Ours - MobileNetV2 | 47.8 | 73.2 | 22.8 | 51.7 | 20.6 | 23.7 | 42.2 | 50.8 | 39.8 | 81.4 | 45.4 | **0.07** |

fps) on NVIDIA GeForce 2080 Ti GPU and our approach with MobileNetV2 feature extractor achieves 29.63 mAP with an improvement in inference time of 0.07s ($\sim$ 15 FPS). Our approach surpasses DSS [12] by around 4.8 mAP and it is around 80$\times$ faster. COG [18] computes the object detection for 10 classes, so we compare our approach with 10 classes in Tab. II. Our approach with ResNeXt50 feature extractor surpasses the COG [18] by around 0.6 mAP and it is very very faster. The mAP of 'desk' in Tab. II has decreased compared to COG because of ambiguity with 'table' class, as we considered 'table' in our object classes. Our proposal-free single-stage approach of 3D object detection surpasses the other methods and is faster than other methods. The other compared methods assume Manhattan world assumption, whereas our approach predict the objects without any assumptions. The methods like DSS [12], COG [18] uses point cloud data that is converted from sparse depth data. Due to sparse data, the methods are unable to extract features for small objects (like garbage bin, lamp etc). Our approach fuses the multi-scale RGB and depth feature maps to extract features and detect objects with multi-layer predictions, which makes it more robust.

### *B. Qualitative Results*

The qualitative results of our approach with MobileNetV2 and ResNeXt50 feature extraction networks are shown in Fig. 3. The results show that our approach is able to detect the objects even if the objects are truncated (like sink, bathtub etc.) and occluded (like table, box etc.) and even small objects like garbage bin.

### *C. Ablation Studies*

*1) Input Data Type:* We present the study of our approach with ResNeXt50 feature extractor using different input data types (like RGB, depth, encoded depth etc.) in Tab III. We have encoded the depth image to a color image using a computationally inexpensive method presented in [26]. Essentially, we first normalize the depth values to [0, 255] and apply a jet colormap to transform a single channel depth image into a three channel color image. This method distributes the depth information to all three channels and it provides good object boundaries in terms of edges which helps in learning semantic feature representations.

TABLE III
ABLATION STUDY ON DIFFERENT TYPES OF INPUT DATA TO OUR APPROACH WITH RESNEXT50 FEATURE EXTRACTOR ON SUN RGB-D TEST DATASET.

| **Data Type** | **mAP** | **Time (s)** |
|---|---|---|
| RGB | 27.9 | 0.22 |
| Depth | 28.1 | 0.22 |
| Depth Encoded [26] | 28.7 | 0.22 |
| RGB and Depth | 31.05 | 0.24 |
| RGB and Depth Encoded | 31.77 | 0.24 |

The mAP of single input data like RGB, depth and depth encoded is almost similar. Our approach when combines RGB and depth encoded data improves the mAP as compared with only single input data. This shows that multi-scale fusion of RGB and depth feature maps significantly improve the 3D object detection results. The depth encoded data showed slightly higher performance than depth data, as colorization of depth data provided more distinctive edges which helped in learning good semantic features. We have also observed that the objects with less texture information can be detected well when we fuse RGB and depth encoded data, because less texture objects does not provide great appearance details in RGB images.

*2) Fusion types::* There are various methods in the literature to fuse the data from two modalities. We have studied *late fusion* and *deep fusion* of multi-scale RGB and depth feature

maps. In *late fusion*, we extract the feature maps of RGB and depth data using ResNeXt50 feature extractor, fuse the multi-scale mid and late feature maps by concatenation and apply $1\times1$ convolutional layer to shuffle and learn features for better fusion. In *deep fusion* scheme, we fuse the mid to late level feature maps by element-wise mean as shown in the network architecture.

The performance of our approach (with ResNeXt50 feature extractor) with *late fusion* scheme is 30.5 mAP and *deep fusion* scheme is 31.77 mAP. The *deep fusion* scheme provides slightly improved performance than *late fusion* scheme, because the former scheme enables interactions between RGB and depth feature maps which provides robust feature learning and improves accuracy.

## VI. Conclusion

The SL3D approach provides a proposal-free single-stage 3D object detection method from RGB-D images by leveraging multi-scale feature fusion of RGB and depth feature maps, and multi-layer predictions. The multi-scale feature fusion enables to fuse features from mid-level to high-level feature maps. The fusion of RGB and depth feature maps improves the performance of the object detection by exploiting data from both modalities. The prediction of the objects is achieved from multi-layer feature maps to account objects of different sizes in feature map space. We evaluated our approach using two feature extractors like ResNeXt50 and MobileNetV2 on publicly available SUN RGB-D dataset. The results show that our approach is better than other state-of-the-art approaches both in terms of mAP and inference time, which makes it a suitable approach for 3D object detection.

## Acknowledgment

This work is funded by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No. 765866.

## References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[2] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection." in *ICIP (1)*. IEEE, 2002, pp. 900–903.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 2005, pp. 886–893.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[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.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 580–587.

[7] E. Ahmed, A. Saint, A. E. R. Shabayek, K. Cherenkova, R. Das, G. Gusev, D. Aouada, and B. E. Ottersten, "Deep learning advances on different 3d data representations: A survey." *CoRR*, vol. abs/1808.01462, 2018.

[8] Z. Deng and L. J. Latecki, "Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images." in *CVPR*. IEEE Computer Society, 2017, pp. 398–406.

[9] S. Gupta, R. B. Girshick, P. Arbelaez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation." *CoRR*, vol. abs/1407.5736, 2014.

[10] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1329–1335.

[11] S. Song and J. Xiao, "Sliding shapes for 3d object detection in depth images." in *ECCV*, ser. Lecture Notes in Computer Science, vol. 8694. Springer, 2014, pp. 634–651.

[12] S. Song and J. Xiao, "Deep sliding shapes for amodal 3d object detection in rgb-d images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 808–816.

[13] J. Lahoud and B. Ghanem, "2d-driven 3d object detection in rgb-d images," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4632–4640.

[14] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks." in *CVPR*. IEEE Computer Society, 2018, pp. 4510–4520.

[15] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks." in *CVPR*. IEEE Computer Society, 2017, pp. 5987–5995.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Computer Vision - ECCV 2016 - 14th European Conference*, vol. 9905. Springer, 2016, pp. 21–37.

[17] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite." in *CVPR*. IEEE Computer Society, 2015, pp. 567–576.

[18] Z. Ren and E. B. Sudderth, "Three-dimensional object detection and layout prediction using clouds of oriented gradients." in *CVPR*. IEEE Computer Society, 2016, pp. 1525–1533.

[19] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 91–99.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015, arxiv:1506.02640.

[22] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2016, cite arxiv:1612.08242.

[23] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[24] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[25] G. Jocher and et al., "ultralytics/yolov5: v2.0," Jul. 2020.

[26] A. Eitel, J. T. Springenberg, L. Spinello, M. A. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition." in *IROS*. IEEE, 2015, pp. 681–687.

[27] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection." in *ECCV*, ser. Lecture Notes in Computer Science, vol. 9908. Springer, 2016, pp. 354–370.

[28] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals." in *ICLR (Poster)*. OpenReview.net, 2017.

[29] W. Liu, R. Ji, and S. Li, "Towards 3d object detection with bimodal deep boltzmann machines over rgbd imagery." in *CVPR*. IEEE Computer Society, 2015, pp. 3013–3021.

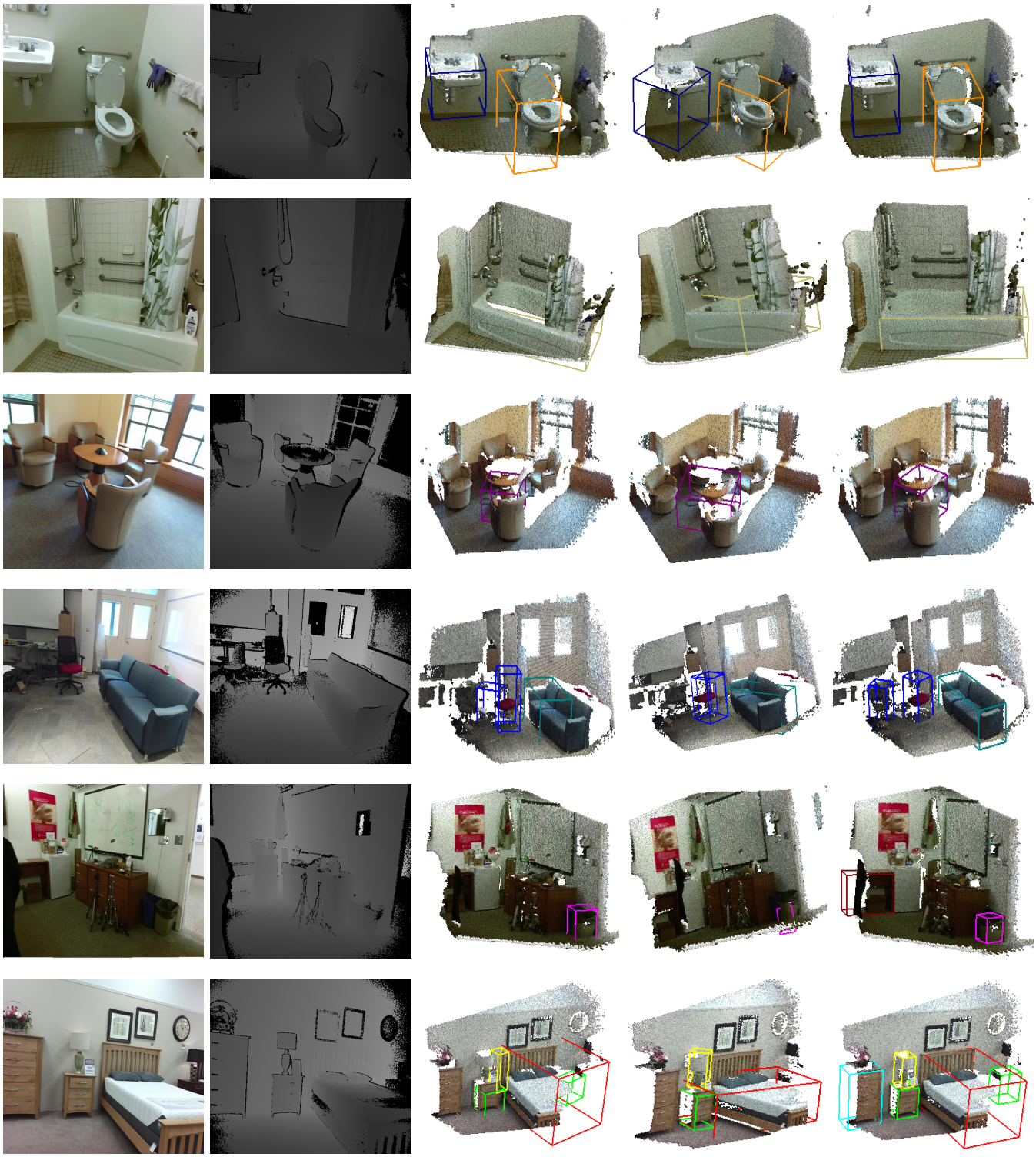


Fig. 3. Qualitative Results on SUN RGB-D dataset. RGB image ($1^{st}$ column), Depth Image ($2^{nd}$ column), Ours-MobileNetV2 ($3^{rd}$ column), Ours-ResNeXt50 ($4^{th}$ column), Ground truth ($5^{th}$ column)