Weakly Supervised 3D Object Detection from Point Clouds

Weakly Supervised 3D Object Detection from Point Clouds

Zengyi Qin*
Massachusetts Institute of Technology
qinzy@mit.edu

Jinglu Wang
Microsoft Research Asia
jinglwa@microsoft.com

Yan Lu
Microsoft Research Asia
yanlu@microsoft.com

ABSTRACT

A crucial task in scene understanding is 3D object detection, which aims to detect and localize the 3D bounding boxes of objects belonging to specific classes. Existing 3D object detectors heavily rely on annotated 3D bounding boxes during training, while these annotations could be expensive to obtain and only accessible in limited scenarios. Weakly supervised learning is a promising approach to reducing the annotation requirement, but existing weakly supervised object detectors are mostly for 2D detection rather than 3D. In this work, we propose VS3D, a framework for weakly supervised 3D object detection from point clouds **without using any ground truth 3D bounding box for training**. First, we introduce an unsupervised 3D proposal module that generates object proposals by leveraging normalized point cloud densities. Second, we present a cross-modal knowledge distillation strategy, where a convolutional neural network learns to predict the final results from the 3D object proposals by querying a teacher network pretrained on image datasets. Comprehensive experiments on the challenging KITTI dataset demonstrate the superior performance of our VS3D in diverse evaluation settings.

CCS CONCEPTS

• **Computing methodologies → Computer vision**; **Scene understanding**; **Vision for robotics**.

KEYWORDS

3D object detection; point clouds; weakly supervised learning

ACM Reference Format:
Zengyi Qin, Jinglu Wang, and Yan Lu. 2020. Weakly Supervised 3D Object Detection from Point Clouds. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413805

1 INTRODUCTION

As an essential challenge in scene understanding, 3D object detection focuses on detecting and localizing the 3D bounding boxes of objects from input sensory data such as images and point clouds. Since point clouds offer a 3D geometric perception of the world, many 3D object detectors [9, 20, 29, 47] use point clouds as input data. These 3D object detectors transform unorganized point clouds into structured and compact 3D bounding box representations, and have been considered as pivotal components in mobile robots and augmented reality. However, training the 3D object detection algorithms would require human annotators to label a huge amount of amodal 3D bounding boxes in unorganized 3D point clouds. The annotation process could be labour-intensive and time-consuming.

Most of the previous 3D detectors [9, 28–30, 33] are based on fully supervised learning and cannot adapt to scenarios where 3D labels are absent. Consequently, it is worthy of finding ways to achieve weakly supervised or even unsupervised learning of 3D object detection, which could reduce the requirement of training labels. Existing studies on weakly supervised learning of object detection mainly focus on 2D detection [5, 22, 23]. However, 2D detection does not provide a 3D geometric understanding of the scene, which is crucial in various applications such as self-driving. Previous approaches [25, 37] attempt to solve 3D object detection by leveraging non-parametric models without ground truth supervision, but they are not designed to provide the accurate 3D bounding boxes of objects. A recent work [40] focuses on semi-supervised 3D object detection, but it still assumes the existence of full 3D annotation for specific classes of objects.

In this work, we aim to develop a framework for weakly supervised 3D object detection from point clouds. We do not need ground truth 3D bounding boxes for training, but make full use of the commonly used data format, i.e., paired images and point clouds, for weak supervision. Without ground truth, the **key challenges** of 3D object detection are *1) how to generate 3D object proposals from unstructured point cloud* and *2) how to classify and refine the proposals to finally predict 3D bounding boxes.* To solve the **first** challenge, we propose an **unsupervised 3D object proposal module (UPM)** that leverages the geometric nature of scan data to find regions with high object confidence. Point cloud density [8] has been considered an indicator of the presence of an object. A volume containing an object could have a higher point cloud density, but the absolute density is also significantly affected by the distance to the scanner. Regions far away from the scanner are of low point cloud density even if they contain objects. To eliminate the interference of distance, we introduce the *normalized point cloud density* that is more indicative of the presence of objects. 3D object proposals are generated by selecting the preset 3D anchors with high normalized point cloud density. However, the object proposals are class-agnostic, since we cannot distinguish the class of an object based on the normalized point cloud density. The rotation of an object is also ambiguous under the partial observation of the captured point clouds on its surface. Therefore, the pipeline should be able to classify the proposals into different object categories and regress their rotations, which reveal the second challenge.

*The work was done when Zengyi Qin was an intern at MSR.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413805

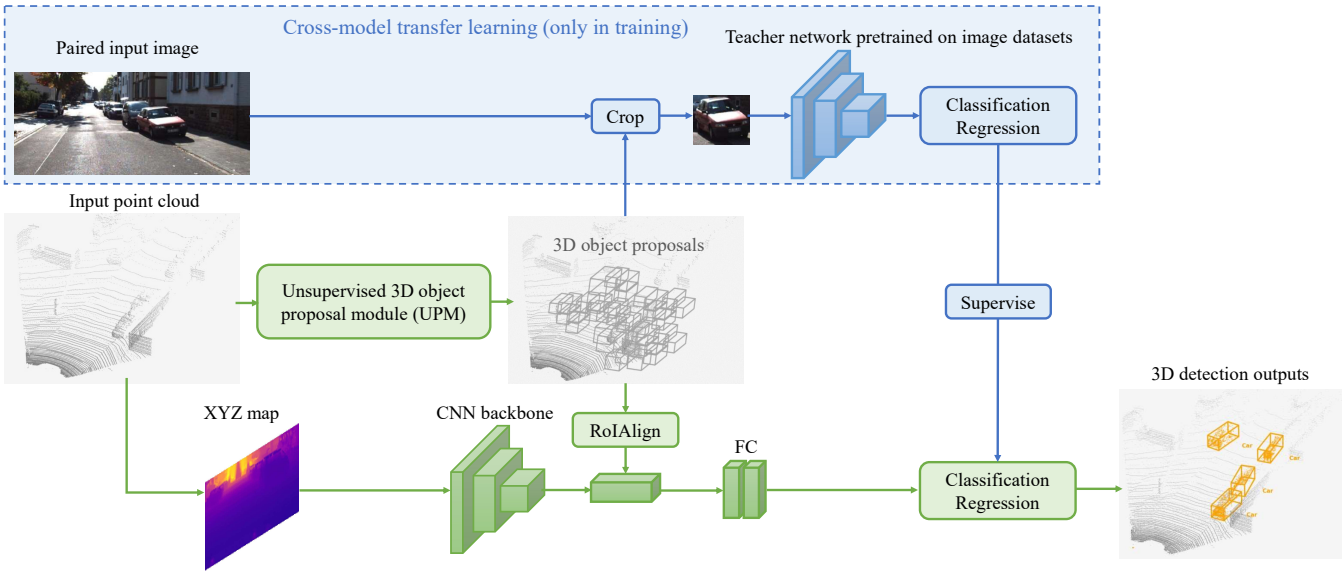

Figure 1: Overview of the proposed weakly supervised 3D object detection framework. The first key component is the unsupervised 3D object proposal module (UPM) that selects 3D anchors based on the normalized point cloud density. The second component is the cross-modal transfer learning module that transfers the knowledge, including object classification and rotation regression, from image datasets into the point cloud based 3D object detector.

To solve the **second** challenge, we propose a **cross-modal transfer learning** method, where the point cloud based detection network is regarded as a student and learns knowledge from an off-the-shelf teacher image recognition network pretrained on existing image datasets. The 3D object proposals produced by the UPM are projected onto the paired image and classified by the teacher network, then the student network mimic the behavior of the teacher during training. Using the teacher network as a media, we transfer the knowledge from the RGB domain to the targeted point cloud domain, which can save the annotation cost of 3D object detection on unlabeled datasets and facilitate the fast deployment of 3D object detectors in new scenarios. We notice that the teacher network is not always capable of supervising its student because of the gap between two different datasets, especially when the teacher is not confident of its own predictions. In light of this, we propose a rectification method that automatically strengthens confident supervisions and weakens uncertain ones. Thus the student learns more from reliable supervision signals while less from those unreliable. To validate the proposed approach and each of its components, we conduct comprehensive experiments on the challenging KITTI [15] dataset. Promising performance is shown under diverse evaluation metrics. Our method demonstrates over 50% improvement in average precision compared to previous weakly supervised object detectors. In summary, our contributions are three-fold:

- An unsupervised 3D object proposal module (UPM) that selects and aligns anchors using the proposed normalized point cloud density and geometry priors.
- An effective approach to transferring knowledge from 2D images to 3D domain, which makes it possible to train 3D object detectors on unlabeled point clouds.
- A pioneering framework for weakly supervised learning of 3D object detection from point clouds, which is examined through comprehensive experiments and demonstrates superior performance in diverse evaluation settings.

2 RELATED WORK

2.1 3D object detection

The objective of 3D detection is detecting objects of interest and localizing their amodal 3D bounding boxes. Existing approaches are based on full supervision, assuming that the accurate 3D ground truth is provided in the dataset. MonoGRNet [30] proposes predicting the 3D location by first estimating instance-level depth and the projected 3D center. TLNet [31] triangulates the objects using stereo images. MV3D [9] introduces bird's eye view (BEV) representation of point cloud data to construct region proposal network [32]. F-PointNet [29] detects object on image to reduce search space in LiDAR point cloud. VoxelNet [48] groups 3D point cloud into each voxel where a fix-length feature vector is extracted. AVOD [20] aggregates the image view and bird's eye view (BEV) to produce high-quality object proposals. STD [47] transforms the point cloud features from sparse to dense. The state-of-the-art performance of these models are established on sufficient training labels.

2.2 Weakly supervised object detection

Weakly supervised object detection assumes that the instance-level bounding box annotations are not provided by the training set, and the supervision can come from image-level annotations. Cho *et al.* [10] proposes to discover dominant objects and localize their 2D bounding boxes via a part-based region matching approach in a fully unsupervised fashion. With image-level labels, Han *et al.* [17]

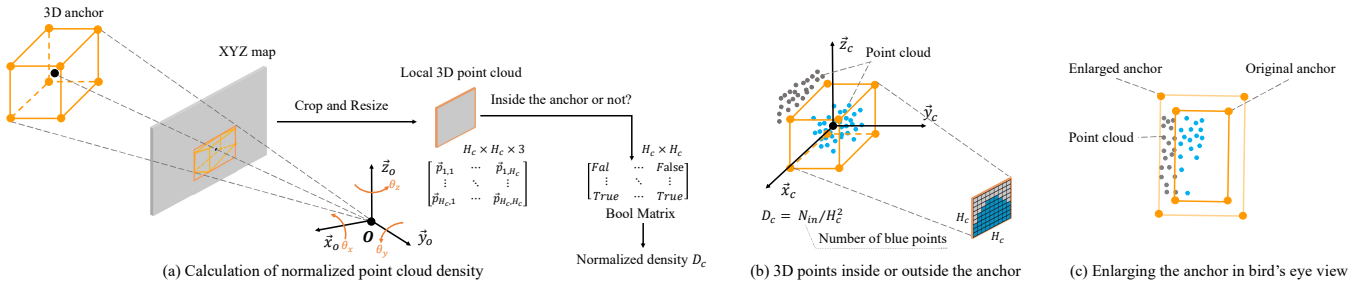

Figure 2: Normalized point cloud density. The point cloud density inside a volume is influenced by two factors that are 1) whether the volume contains an object and 2) the distance of the volume to the sensor. The density increases when an object is present but decreases as the distance grows. Our normalization strategy eliminates the influence of distance. (a) The preset 3D anchor is projected to the XYZ map, where its projection is cropped out and scaled to a square patch with fixed size $H_c \times H_c$ that is distance-irrelevant. The square patch represents H_c^2 3D points, where one pixel corresponds to one point. (b) Among the 3D points, N_{in} points are inside the 3D anchor. The normalized point cloud density D_c is calculated as N_{in}/H_c^2. (c) The grey and blue points are on the same object. If the original anchor fails to bound the object and only contains a part of it, the enlarged version would contain more points, i.e., the grey ones.

propose learning 2D localization by iterative training. Sangineto *et al.* [35] select object proposals that are more confident in training. WSDDN [5] modifies image classification networks to predict at the region level for object proposal selection and classification. OICR [39] and PCL [38] utilize online instance classification refinement and proposal cluster learning to improve the detection performance. OIM [26] leverages instance graphs to mine all possible instances using image-level annotations. Although there are studies on 2D object detection in unsupervised or weakly supervised settings, 3D detection without ground truth supervision is much less explored. The work of [25] attempts to solve the unsupervised 3D object detection by performing weighted clustering on point clouds, but is not designed to predict the amodal 3D bounding boxes. Reasoning at region level, object detection is already a nontrivial task on 2D images. Learning 3D detection without full supervision is more challenging and will be explored in this paper.

2.3 Cross-modal transfer learning

Knowledge distillation [19] is widely used for transferring supervision cross moralities, such as [2, 14, 16]. Huang *et al.* [21] minimize the neural activation distribution of the teacher and the student network. Sungsoo *et al.* [3] propose maximizing the high-level mutual knowledge between the teacher and its student instead of matching their activation or other handcrafted features. Gupta *et al.*[16] propose to distill semantic knowledge from labeled RGB images to unlabeled depth for 2D recognition tasks, e.g., 2D detection and segmentation, while the 3D geometric information in depth data is not fully utilized. Instead, our method does not require labels of the RGB images on the targeted dataset, and also explores the depth information for object proposal in 3D space.

3 APPROACH

3.1 Overview

The objective is to detect and localize amodal 3D bounding boxes of objects from the input point clouds. Unlike existing 3D detectors, we do not rely on the ground truth 3D bounding boxes on the targeted datasets during training. As is shown in Figure 1, the detection pipeline consists of two stages, 1) an unsupervised 3D object proposal model (UPM) and 2) a cross-modal transfer learning method. The first UPM stage outputs object proposals indicating the regions potentially containing the objects from point clouds. The second transfer learning stage classifies and refines the proposals to produce the final predictions by leveraging a teacher model pretrained on image datasets. LiDAR scanners are not a necessity in providing the input point clouds, which could also be obtained from a monocular image [44] or a pair of stereo images [43]. It is assumed that each frame of point clouds has a paired image in the training set, but this is not required in testing where only the point clouds are needed. This assumption is satisfied by most datasets.

3.2 Unsupervised 3D object proposal module

3D object proposals are defined as volumes potentially containing objects. We first preset 3D anchors and then select anchors with high object confidence as object proposals. The preset anchors are placed at an interval of 0.2m on the ground plane spanning [0, 70m] × [-35m, 35m]. Without ground truth supervision, it is not feasible to directly train a model to select anchors as the proposals from raw point clouds. Therefore, we explore the geometric nature of point clouds and leverage our prior knowledge to find potential regions with objects. The point cloud density inside a volume can indicate whether an object is contained in that volume. A high density represents a high objectiveness confidence. However, the point density is also significantly influenced by the distance to sensor. Distant points are much sparsely distributed than nearby points. Hence, we introduce a distance-invariant point density measurement, **normalized point cloud density** D_c, for effectively selecting potential candidate anchors.

3.2.1 Normalized point cloud density. We project the 3D point cloud to the front view to obtain the pixel-wise XYZ map, where each pixel has three channels indicating the 3D coordinates. The empty pixels are filled by inpainting [4]. By projecting the 3D anchor to the front view, we get a 2D box that bounds the projection,

as illustrated in Figure 2 (a). We crop the patch of XYZ map inside the bounding box and resize it into $H_c \times H_c$ by interpolation, after which we obtain H_c^2 3D points, where each point is denoted as $\vec{p}_{i,j}$ as is shown in Figure 2 (a). It is noteworthy that so far the resized front-view patch of each anchor consists the *same* number of 3D points regardless of the distance-relevant sparsity of point cloud. Some of these points are inside the anchors, denoted as *True* in the boolean matrix in Figure 2 (a), while those outside the anchor are *False*. Among those points, if there are N_{in} points inside the 3D anchor, its point cloud density D_c can be expressed as N_{in}/H_c^2. N_{in} does not include the points on the ground planes, which are identified by RANSAC [12] based plane fitting. D_c is not influenced by the distance because we interpolate the front-view patch to the same size. If a targeted object is contained in an anchor, D_c of this anchor should be above a certain threshold δ.

The normalized point cloud density requires calculating N_{in}, i.e., identifying how many points among those H_c^2 points are inside the 3D anchor. Figure 2 (b) illustrates how to distinguish whether point $\vec{p}$ is inside the anchor. By transforming the 3D point $\vec{p}$ from the camera coordinate system to the local anchor coordinate system, we obtain $(q_x, q_y, q_z) = ((\vec{p} - \vec{c}) \cdot \vec{x}_c, (\vec{p} - \vec{c}) \cdot \vec{y}_c, (\vec{p} - \vec{c}) \cdot \vec{z}_c)$, where $\vec{c}$ denotes the translation from the origin to the anchor center, $\vec{x}_c$, $\vec{y}_c$ and $\vec{z}_c$ are the axis aligned to anchor dimensions. Denote half of the anchor size along each axis as l_x, l_y and l_z. The element $b_{i,j}$ in the boolean matrix of Figure 2 (a) is determined by:

$$b_{i,j} = (|q_{i,j,x}| < l_x) \wedge (|q_{i,j,y}| < l_y) \wedge (|q_{i,j,z}| < l_z) \qquad (1)$$

where $q_{i,j,x}$, $q_{i,j,y}$, $q_{i,j,z}$ are the transformed XYZ coordinates of points at index of (i, j) in the patch. Please note that this process can be done in parallel in GPU implementation for all the points in all the front-view patches. Different from the forward pass in a convolutional neural work, the whole computation does not involve extensive floating point operations, demanding only a little computational resource and time cost. Learning is not required here, which further enhances the efficiency of the pipeline.

3.2.2 Anchor selection and alignment. The anchors with $D_c < \delta$ are considered negative examples and are efficiently removed. The 3D object proposals are selected from the remaining anchors. We enlarge each anchor by $1+\epsilon$ times as is in Table 2 (c). If the enlarged anchor does not contain extra points excluding the points in the original anchor, the original anchor is selected as one of the object proposals. This ensures that the anchor bounds the whole object rather than only a part of it, otherwise the enlarged anchor could contain extra points. ϵ is small enough to avoid the enlarged anchor containing point clouds from the neighbouring objects.

The selected anchors may not well aligned with the targeted object. We observe that some points should be close to the anchor's rectangular surfaces if the anchor fits to the target. In light of this, we shift the anchor in a small range so that it is better aligned with the points it contains. Specifically, for the axis of $\vec{x}_c$, we find the point $\vec{p}_{i,j}$ that is inside the anchor and has the greatest projected length $|q_{i,j,x}|$, then we move the anchor along $\vec{x}_c$ to align the closest surface with $\vec{p}_{i,j}$. The alignment along $\vec{y}_c$ and $\vec{z}_c$ are in the same way. Similar to the calculation in Equation 1, this anchor alignment is done in parallel for all anchors efficiently.

3.3 Image to point cloud knowledge transfer

The object proposals produced by UPM are not the final detection outputs. It is observed that some selected anchors will contain objects not belonging to the targeted category. For instance, an anchor of car class may bound points of trees, possessing a a higher point cloud density than the selection threshold. As no ground truth is provided, it is difficult to recognize object category from point clouds. In addition, since the object rotations are hard to determine based on partially observed point cloud, object proposals from UPM are of the same rotations as pre-defined. One possible way to enhance the recognition ability is introducing knowledge from another domain. There is a large quantity of RGB data [46] that are already labeled. We expect to transfer the knowledge from the RGB domain to the point cloud domain. To this end, we propose a cross-modal transfer learning method, where the point cloud based 3D detector is regarded as student and learns knowledge from a teacher network pre-trained on large-scale image recognition datasets.

3.3.1 Image based teacher network. The teacher is an image recognition and view point regression network employing the VGG16 [36] architecture and is pre-trained on the ImageNet [34] and PASCAL VOC [11] with image-level classification labels and view point labels provided by [46]. This is consistent with previous work on weakly supervised object detection [5, 26, 38, 39] where annotated bounding boxes are assumed absent during training. Taking an image with no more than one object as input, the teacher network classifies the image as background or a class of objects, and at the same time regresses the object view point as its rotation. The view point regression is considered a multi-bin classification problem, where we predict the probabilities of 16 angle bins divided from a unit circle. The rotation output is the expectation value of the angles of all bins. Neither of the datasets has overlaps with KITTI dataset where we evaluate our approach in the experiment. The teacher is used as an off-the-shelf model in training of the 3D object detection model, shown as the blue branch of Figure 1.

3.3.2 Point cloud based student network. The student represents the second stage of a point cloud based 3D object detector, consisting of a VGG16 [27] backbone, a RoIAlign [18] layer and the fully connected layers, as is illustrated in the green branch in Figure 1. The input point clouds are converted to a front-view XYZ map before fed into the backbone similar as the work [45]. Using the image paired with the point clouds, we can distill the recognition knowledge from the teacher network to the student. More specifically, we project each object proposal produced by UPM to both the RGB image and the front-view XYZ map. Then we crop out the projection on the image and recognize the object proposal using the teacher network. At the same time, we use RoIAlign [18] to extract encoded features of each proposal from the student backbone and feed the features to fully connected layers to predict the object class and rotation. During training, each object proposal has two predictions from the teacher and the student respectively. The student learns to imitate the confidence of the teacher using the rectified cross-entropy loss as described in the followings.

When distilling the capability from an off-the-shelf teacher into a student of a different dataset, there could inevitably occur problems. First, the teacher may not be confident of some of its own outputs.

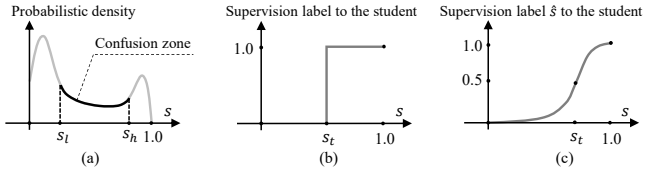

Figure 3: Supervision rectification. Given an object proposal to be classified, if the confidence of the teacher network is within the confusion zone, the loss value will be masked out.

Such confusing outputs are supposed not to teach the student. More specifically, as is shown in Figure 3 (a), when the classification score s from the teacher is within the confusion zone, s will not be used to train the student branch. Only when the teacher is confident enough will the student learn from it. Second, all the predictions above the confusion zone should not be treated equally as is in Figure 3 (b), where the supervision label to the student is a binary value. In Figure 3 (b), if the score predicted by teacher network is above s_t, the supervision label to the student network will be 1.0, indicating an absolute positivity. However, this makes all positive labels indistinguishable. In light of this, we propose supervision rectification, where we rectify the teacher's output s to produce a soft label $\hat{s}$ as is illustrated in Figure 3 (c). The rectified cross entropy loss is formulated:

$$\mathcal{L}_r = -\left[\hat{s}\log(\tilde{s}) + (1-\hat{s})\log(1-\tilde{s})\right] \cdot \mathbb{1}(s \notin [s_l, s_h]) \quad (2)$$

where $\tilde{s}$ is the prediction of the green branch to be supervised. $\mathbb{1}(\cdot)$ refers to the indicator function whose value is 1 iff s from the teacher is not in the confusion zone shown in Figure 3 (a). The relationship between $\hat{s}$ and s is given by the following function:

$$\hat{s} = \frac{1 + e^{(s_t - 1)k}}{1 + e^{(s_t - s)k}} \quad (3)$$

See Figure 3 (c) for the curve of Equation 3, where s_t is a soft threshold of classification confidence and k controls slope. If k is extremely large, any confidence s below s_t will be mapped to 0. The rectified label takes into account the confidence of teacher network. Higher confidence leads to stronger positivity in the supervision labels provided to the student network.

4 EXPERIMENT

The evaluation is done on the KITTI [15] dataset, where the publicly available training and validation set are split following [7–9] Both of the splits contain half of the whole training set and has no overlap in terms of the video sequences where the frames come from.

Metrics. Various metrics including recall rate and average precision (AP) with different intersection of union (IoU) thresholds are utilized to provide a thorough evaluation of the proposed method. If the IoU between a predicted bounding box and a ground truth bounding box is no less than a given threshold, then the prediction is considered correct and the ground truth is recalled. Recall rate measures the proportion of ground truth that is recalled. AP calculates the precision average across different recall thresholds. These metrics have been widely adopted in previous works [11, 30, 31, 38, 39].

There are three fundamental questions that we aim to answer: 1) How is the quantitative performance of the proposed detection framework and its comparison to existing methods? 2) How does the performance change with respect to different types of input signals including monocular images, stereo images and LiDAR scans? 3) How important is the unsupervised 3D object proposal module to the whole framework?

4.1 Implementation.

The framework is built on Tensorflow [1]. Both the teacher and the student network employ the VGG16 [27] backbone. In the teacher network, we scale the input images to a fixed size 64×64 to introduce scale invariance. In the student network, RoI features are from the last conv layer of the backbone. The features are resized to 7×7 and passed to three consecutive fully connected layers with 1024 and 512 hidden units, where the last layer outputs the classification probability and the multi-bin probability for rotation prediction. There are 16 bins in total. In supervision rectification, we consider object proposals with $s > s_h$ as positive examples and $s < s_l$ as negative examples. In training, a mini-batch has 1024 positive and 1024 negative examples. The top 512 object proposals are kept in inference. For the hyperparameters, we choose $H_c = 32, \delta = 0.5, \epsilon = 0.2, s_t = 0.6, s_l = 0.4$ and $s_h = 0.6$ by grid search. The whole network is trained using Adam [24] optimizer for 40 epochs with a constant learning rate of 10^{-4}. L2 regularization is applied to model parameters at a decay weight of 5×10^{-5}.

Input type. A frame of input point clouds could be obtained from three sources including a monocular image, a pair of stereo images and LiDAR scans. For the monocular image, we feed it to DORN [13] to predict the pixel-level depths then convert the depths to 3D point clouds. For the stereo images, we feed them to PSMNet [6] to produce the depths that are transformed to 3D point clouds. For LiDAR, point clouds are directly accessible. Corresponding to the data types, we train and evaluate three versions of our framework. At test time, a single forward pass from input point clouds to the output 3D bounding boxes takes 44ms on a Tesla P40 GPU, demanding 9.39 billion floating point operations (FLOPS). The fast anchor selection stage involves 9.56 million FLOPS that is only 0.1% of the total computational cost. Most of the resource is consumed by the backbone network, meaning that the efficiency can be improved when a lighter backbone is employed.

4.2 Weakly supervised object detection

Three state-of-the-art weakly supervised detection methods [38, 39, 42] are compared. PCL [38] iteratively learns refined instance classifiers by clustering the object proposals. OICR [39] adds online instance classification refinement to a basic multiple instance learning network. MELM [42] builds a min-entropy latent model to measure the randomness of object localization and guide the discovery of potential objects. The original papers do not provide results on the KITTI [15] dataset, but the authors have made their code publicly available. Strictly following the guidelines of the code base, all the models are retrained and evaluated on the KITTI [15] dataset. Since these methods cannot predict 3D bounding boxes, the comparison would be mainly in 2D domain. Three versions of our VS3D are also evaluated, corresponding to monocular, stereo and LiDAR inputs.

Table 1: Object detection 2D recall on the public KITTI validation set comparing with weakly supervised methods.

Method	Input	Recall (IoU = 0.3)			Recall (IoU = 0.5)			Recall (IoU = 0.7)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
PCL [38]	Mono	49.08	32.68	29.76	24.04	15.23	13.41	6.461	3.948	3.356
OICR [39]	Mono	56.42	43.96	39.55	25.63	18.71	16.59	6.191	4.547	3.768
MELM [42]	Mono	64.52	58.37	53.56	27.70	24.81	21.93	7.234	6.275	5.099
VS3D	Mono	94.09	87.26	76.41	90.12	80.76	67.45	63.83	51.28	40.38
VS3D	Stereo	94.30	87.86	76.84	90.99	82.53	69.13	64.61	53.29	41.70
VS3D	LiDAR	90.92	83.57	73.71	86.60	77.02	65.11	60.31	48.26	38.51
VS3D	Mono + LiDAR	94.48	88.27	77.62	**92.65**	82.58	69.33	65.10	53.07	42.04
VS3D	Stereo + LiDAR	**95.00**	**88.63**	**78.15**	91.08	**83.03**	**70.11**	**65.58**	**53.93**	**42.74**

Table 2: Object detection average precision (AP) on KITTI validation set comparing with weakly supervised methods.

Method	Input	AP_{2D} / AP_{3D} (IoU = 0.3)			AP_{2D} / AP_{3D} (IoU = 0.5)		
		Easy	Moderate	Hard	Easy	Moderate	Hard
PCL [38]	Mono	5.916 / -	4.687 / -	3.765 / -	1.878 / -	1.058 / -	0.935 / -
OICR [39]	Mono	13.50 / -	8.604 / -	8.045 / -	6.481 / -	2.933 / -	3.270 / -
MELM [42]	Mono	8.054 / -	7.282 / -	6.882 / -	2.796 / -	1.486 / -	1.476 / -
VS3D	Mono	77.73 / 55.90	73.82 / 48.83	65.71 / 40.92	76.93 / 31.35	71.84 / 23.92	59.39 / 19.34
VS3D	Stereo	79.04 / **70.72**	75.90 / 63.78	67.55 / 52.03	79.03 / 40.98	72.71 / 34.09	59.77 / 27.65
VS3D	LiDAR	78.64 / 65.96	74.41 / 59.76	66.24 / 49.78	74.54 / 40.32	66.71 / 37.36	57.55 / 31.09
VS3D	Mono + LiDAR	82.46 / 69.75	78.84 / 63.47	69.36 / 52.76	81.60 / 41.83	72.43 / 39.22	64.31 / 32.73
VS3D	Stereo + LiDAR	**82.84** / 70.09	**78.99** / **65.25**	**69.83** / **55.77**	**81.95** / **42.43**	**73.21** / **41.58**	**64.34** / **32.74**

Table 3: The gap between the weakly supervised VS3D and fully supervised methods in 3D object detection.

Method	Input	Sup. type	AP_{3D} (IoU = 0.3)		
			Easy	Moderate	Hard
Deep3DBox [28]	Mono	Full	54.30	43.42	36.57
MonoGRNet [30]	Mono	Full	72.17	59.57	46.08
VoxelNet [48]	LiDAR	Full	**89.32**	**85.81**	**78.85**
VS3D	Mono	Weak	55.90	48.83	40.92
VS3D	LiDAR	Weak	65.96	59.76	49.78

Table 4: Using our unsupervised object proposal module to improve existing weakly supervised object detectors.

Method	Input	AP_{2D} (IoU = 0.3)		
		Easy	Moderate	Hard
PCL [38]	Mono	5.916	4.687	3.765
OICR [39]	Mono	13.50	8.604	8.045
MELM [42]	Mono	8.054	7.282	6.882
PCL [38] + UPM	Mono + LiDAR	16.69	16.53	13.90
OICR [39] + UPM	Mono + LiDAR	**30.87**	**27.97**	**26.98**
MELM [42] + UPM	Mono + LiDAR	18.41	17.65	15.55

Table 1 presents the recall under different IoU thresholds using the top 10 predictions per frame. It is shown that our method outperforms MELM [42] by 20% to 50%, which could be interpreted as a huge margin in terms of recall rate. It can be observed that the margin grows as the evaluation metric becomes stricter, which means our predictions contain more high-quality examples. Table 2 reveals the average precision of 2D and 3D object detection. It is clear that our VS3D has superior performance over the compared baselines. For example, the AP_{2D} of VS3D is over 50% higher than the baselines under IoU threshold 0.3 and 0.5. Three baseline approaches utilize selective search [41] to generate object proposals, which has been a prevailing unsupervised object proposal approach. However, the rich 3D geometric information is left unexplored in these methods. The baseline approaches can be improved using our object proposal methods as is shown in Table 4. We also compare our weakly supervised VS3D with fully supervised methods in Table 3. Results are obtained on the public KITTI [15] validation set widely used by previous work [7–9].

An interesting phenomenon could be observed by comparing our VS3D with different input data types. Generally speaking, if the evaluation metric is in 3D and the IoU requirement is high, the LiDAR based version would be at an advantage. But for 2D metrics such as 2D recall and AP_{2D}, as well as 3D metrics with low IoU

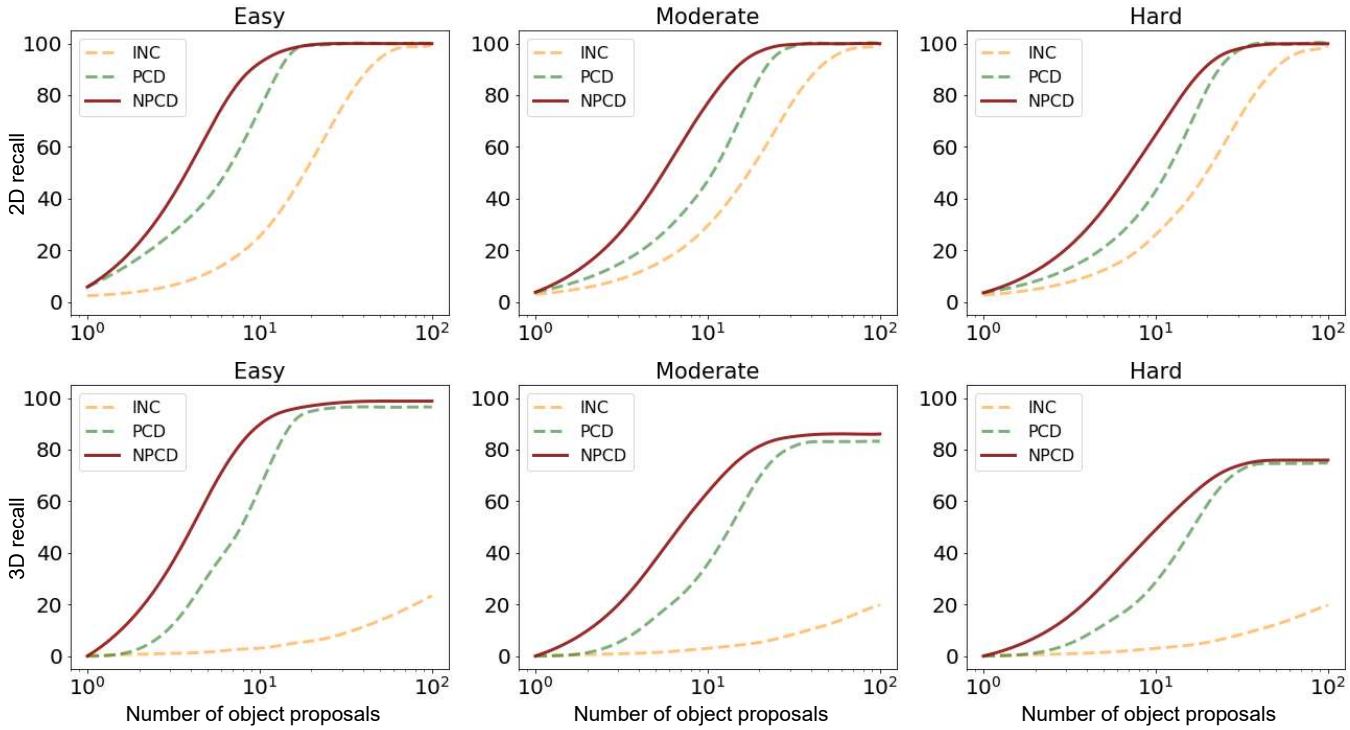

Figure 4: Bounding box recall using different unsupervised 3D object proposal methods. The first and second rows indicate 2D and 3D recall respectively. Results are obtained on KITTI validation set under the IoU threshold of 0.1. NPCD represents the proposed approach based on normalized point cloud density. PCD denotes point cloud density. INC is an inclusive method where all the anchors are kept. It is shown that NPCD uses fewer object proposals to reach a higher 3D recall rate.

threshold, the monocular and stereo versions could have a better performance. This phenomenon could be interpreted as follows. For a 3D metric with high IoU thresholds, the requirement of 3D localization could be far higher, and LiDAR is good at providing such a geometric precision. The point clouds generated by monocular and stereo images cannot reach the precision as a LiDAR does. On the contrary, for a 2D metric or a 3D metric with low IoU thresholds, the requirement of 3D localization is much lower. The point clouds generated from images have a higher resolution that LiDAR point clouds and are more suitable for semantic scene understanding, which is why the image-based approaches have better performance.

In addition to the quantitative results, we also present the qualitative results in Figure 5. The three columns correspond to the Mono, Stereo and LiDAR version of VS3D respectively. It is shown that for objects faraway from the camera origin, the LiDAR version works better than the Mono and Stereo versions, which is reasonable because the quality of point cloud generated from images decreases as the distance increases. In the LiDAR version, predicted 3D bounding boxes are aligned with the ground truth boxes, which is hard to achieve when the ground truth are not available in training.

4.3 Ablation study on UPM

The proposed unsupervised 3D object proposal module (UPM) selects and align predefined anchors with high objectiveness confidence, removing over 98% of the total anchors that are redundant. Our UPM is based on the normalized point cloud density (NPCD) that is a distance-invariant indicator of the presence of objects. In order to validate the effectiveness of our approach, we replace NPCD with another two strategies and compare the bounding box recall rate. The first is an inclusive strategy (INC), where the predefined anchors are all kept without being filtered. The second is based on point cloud density (PCD), where the PCD is measured without the proposed normalization step. Results are shown in Figure 4. It is clear that our NPCD demonstrates better performance than INC and PCD. The gap between NPCD and PCD is mainly due to the normalization step. PCD can reflect the objective confidence in an object proposal, but is severely influenced by distance. Most of the distant anchors are filtered because they have a low point cloud density, even if they contain objects. Therefore, objects far away will be missing in the proposals unless the distance interference is removed, which is achieved in NPCD.

5 CONCLUSION

This paper presents a pioneering work on weakly supervised learning of 3D object detection from point clouds. Our pipeline consists of the unsupervised 3D object proposal module (UPM) and the cross-modal transfer learning module. UPM takes the raw point cloud as input and outputs the 3D object proposals. Without ground truth supervision, UPM leverages the normalized point cloud density to identify the 3D anchors potentially containing objects. Object

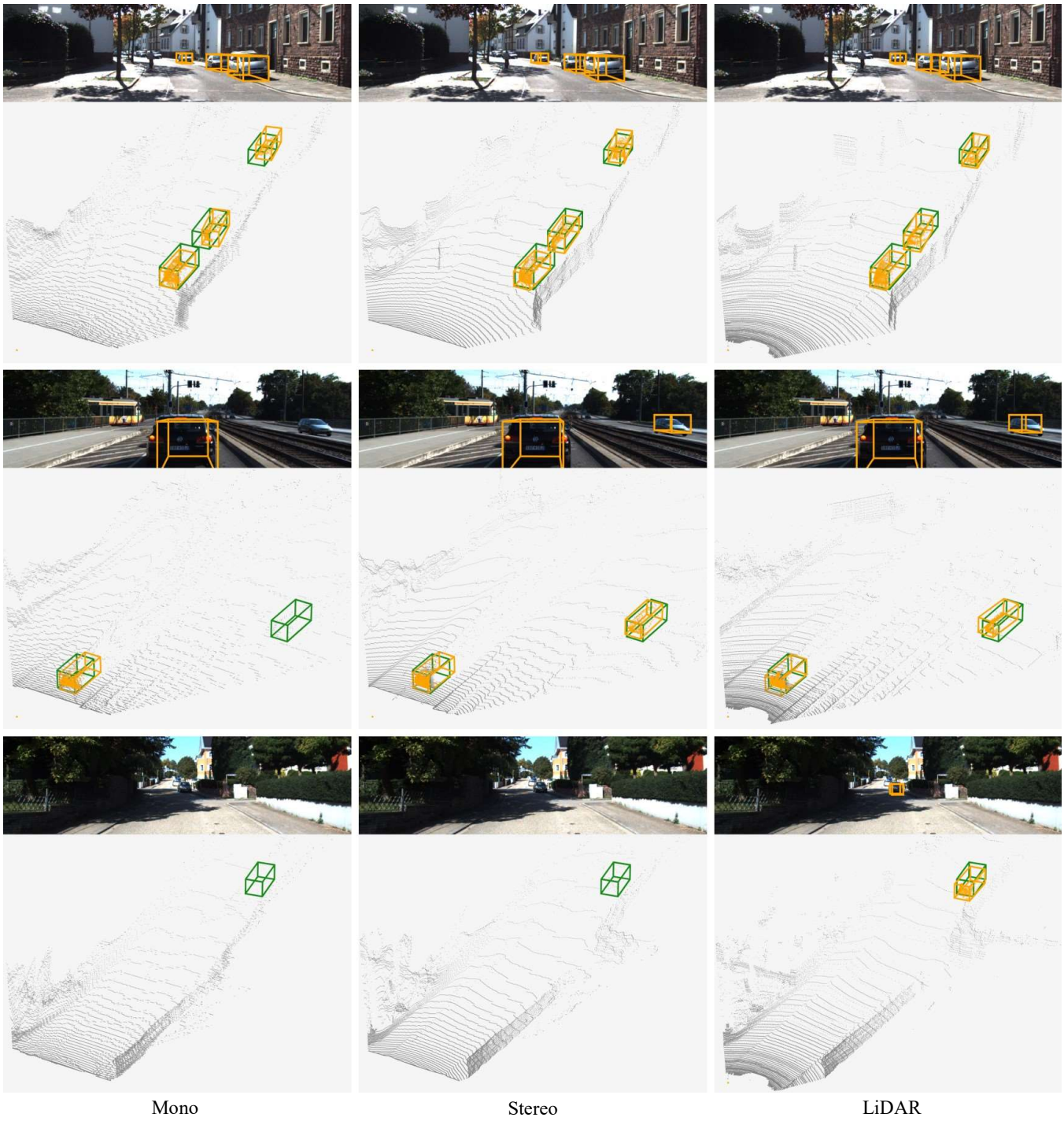

Figure 5: Qualitative results of VS3D on KITTI validation set. Predictions are shown in orange while the ground truths are in green. LiDAR based VS3D is more robust in detecting distant objects in the shadow, as is shown in the third row.

proposals predicted by UPM are classified and refined by the student network to produce the final detection results. The point cloud based student network is trained by an image based teacher network via transferring the knowledge from existing image datasets to the point cloud domain. Comprehensive experiments demonstrate our promising performance in diverse evaluation settings. Our method can potentially reduce the need of manual annotation and facilitate the deployment of 3D object detectors in new scenarios.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

[2] Abrar H Abdulnabi, Bing Shuai, Zhen Zuo, Lap-Pui Chau, and Gang Wang. 2018. Multimodal recurrent neural networks with information transfer layers for indoor scene labeling. *IEEE Transactions on Multimedia* 20, 7 (2018), 1656–1671.

[3] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. 2019. Variational Information Distillation for Knowledge Transfer. *arXiv: Computer Vision and Pattern Recognition* (2019).

[4] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 417–424.

[5] Hakan Bilen and Andrea Vedaldi. 2016. Weakly Supervised Deep Detection Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[6] Jia-Ren Chang and Yong-Sheng Chen. 2018. Pyramid Stereo Matching Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5410–5418.

[7] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 2016. Monocular 3d object detection for autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2147–2156.

[8] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 2015. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*. 424–432.

[9] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. 2017. Multi-view 3d object detection network for autonomous driving. In *IEEE CVPR*, Vol. 1. 3.

[10] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. 2015. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. *computer vision and pattern recognition* (2015), 1201–1210.

[11] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88, 2 (2010), 303–338.

[12] Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.

[13] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. 2018. Deep Ordinal Regression Network for Monocular Depth Estimation. In *Computer Vision and Pattern Recognition (CVPR)*.

[14] Nuno C Garcia, Pietro Morerio, and Vittorio Murino. 2018. Modality distillation with multiple stream networks for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 103–118.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 3354–3361.

[16] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. 2016. Cross modal distillation for supervision transfer. In *IEEE conference on computer vision and pattern recognition (CVPR)*. 2827–2836.

[17] Junwei Han, Dingwen Zhang, Gong Cheng, Lei Guo, and Jinchang Ren. 2015. Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. *IEEE Transactions on Geoscience and Remote Sensing* 53, 6 (2015), 3325–3337.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. *arXiv preprint arXiv:1703.06870* (2017).

[19] Geoffrey E Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv: Machine Learning* (2015).

[20] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krähenbühl, Trevor Darrell, and Fisher Yu. 2018. Joint Monocular 3D Vehicle Detection and Tracking. *arXiv preprint arXiv:1811.10742* (2018).

[21] Zehao Huang and Naiyan Wang. 2017. Like What You Like: Knowledge Distill via Neuron Selectivity Transfer. *arXiv: Computer Vision and Pattern Recognition* (2017).

[22] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Cross-Domain Weakly-Supervised Object Detection Through Progressive Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[23] V. Kantorov, M. Oquab, Cho M., and I. Laptev. 2016. ContextLocNet: Context-aware Deep Network Models for Weakly Supervised Localization. In *Proc. European Conference on Computer Vision (ECCV), 2016*.

[24] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference for Learning Representations*.

[25] Kamran Kowsari and Manal H Alassaf. 2016. Weighted Unsupervised Learning for 3D Object Detection. *International Journal of Advanced Computer Science and Applications* 7, 1 (2016).

[26] Chenhao Lin, Siwen Wang, Dongqi Xu, Yu Lu, and Wayne Zhang. 2020. Object Instance Mining for Weakly Supervised Object Detection. *AAAI*.

[27] D Zeiler Matthew and Fergus Rob. 2014. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision*. 818–833.

[28] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Košecká. 2017. 3d bounding box estimation using deep learning and geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 5632–5640.

[29] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. 2017. Frustum PointNets for 3D Object Detection from RGB-D Data. *arXiv preprint arXiv:1711.08488* (2017).

[30] Zengyi Qin, Jinglu Wang, and Yan Lu. 2019. MonoGRNet: A Geometric Reasoning Network for Monocular 3D Object Localization. *AAAI* (2019).

[31] Zengyi Qin, Jinglu Wang, and Yan Lu. 2019. Triangulation Learning Network: from Monocular to Stereo 3D Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (2017).

[33] Thomas Roddick, Alex Kendall, and Roberto Cipolla. 2018. Orthographic feature transform for monocular 3D object detection. *arXiv preprint arXiv:1811.08188* (2018).

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[35] Enver Sangineto, Moin Nabi, Dubravko Culibrk, and Nicu Sebe. 2019. Self paced deep learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence* 41, 3 (2019), 712–725.

[36] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[37] Li Sun, Cheng Zhao, Zhi Yan, Pengcheng Liu, Tom Duckett, and Rustam Stolkin. 2019. A Novel Weakly-Supervised Approach for RGB-D-Based Nuclear Waste Object Detection. *IEEE Sensors Journal* 19, 9 (2019), 3487–3500.

[38] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Loddon Yuille. 2018. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence* (2018).

[39] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. 2017. Multiple Instance Detection Network with Online Instance Classifier Refinement. In *CVPR*.

[40] Yew Siang Tang and Gim Hee Lee. 2019. Transferable Semi-Supervised 3D Object Detection From RGB-D Data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 1931–1940.

[41] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. 2013. Selective Search for Object Recognition. *International Journal of Computer Vision* 104, 2 (2013), 154–171.

[42] Fang Wan, Pengxu Wei, Jianbin Jiao, Zhenjun Han, and Qixiang Ye. 2018. Min-Entropy Latent Model for Weakly Supervised Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1297–1306.

[43] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. 2019. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. In *CVPR*.

[44] Xinshuo Weng and Kris Makoto Kitani. 2019. Monocular 3D Object Detection with Pseudo-LiDAR Point Cloud. *ArXiv* abs/1903.09847 (2019).

[45] Bichen Wu, Forrest N. Iandola, Peter H. Jin, and Kurt Keutzer. 2016. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. *CoRR* (2016).

[46] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. 2014. Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

[47] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. 2019. STD: Sparse-to-Dense 3D Object Detector for Point Cloud. *ICCV* (2019). arXiv:1907.10471 http://arxiv.org/abs/1907.10471

[48] Yin Zhou and Oncel Tuzel. 2018. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *computer vision and pattern recognition* (2018), 4490–4499.