MonoGRNet: A General Framework for Monocular 3D Object Detection

Zengyi Qin, Jinglu Wang and Yan Lu

Abstract—Detecting and localizing objects in the real 3D space, which plays a crucial role in scene understanding, is particularly challenging given only a monocular image due to the geometric information loss during imagery projection. We propose MonoGRNet for the amodal 3D object detection from a monocular image via geometric reasoning in both the observed 2D projection and the unobserved depth dimension. MonoGRNet decomposes the monocular 3D object detection task into four sub-tasks including 2D object detection, instance-level depth estimation, projected 3D center estimation and local corner regression. The task decomposition significantly facilitates the monocular 3D object detection, allowing the target 3D bounding boxes to be efficiently predicted in a single forward pass, without using object proposals, post-processing or the computationally expensive pixel-level depth estimation utilized by previous methods. In addition, MonoGRNet flexibly adapts to both fully and weakly supervised learning, which improves the feasibility of our framework in diverse settings. Experiments are conducted on KITTI, Cityscapes and MS COCO datasets. Results demonstrate the promising performance of our framework in various scenarios.

Index Terms—3D object detection, monocular, weakly supervised learning.

1 INTRODUCTION

A crucial task in scene understanding is 3D object detection, which aims to predict the amodal 3D bounding boxes of objects from input sensory data such as LiDAR point clouds and images. Compared to LiDAR based [1], [2], [3] and multi-view based [4], [5], [6] 3D object detectors, monocular image-based methods [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17] only take a single-view RGB image as input in inference, which means they have lower requirement on sensors and are less expensive in real-world implementation. If achieving satisfactory detection performance, they can become an important module in mobile robot perception. It would be more attractive if the monocular 3D detection can be learned without 3D labels, which require intensive labors. However, monocular 3D object detection is an ill-posed problem due to the depth information loss in 2D image planes, let alone the challenges when 3D annotations are not offered.

To accurately detect and localize the objects in 3D using a monocular image, recent approaches [10], [11], [12], [18] have been proposed to first predict the pixel-level depth and convert the monocular image to 3D point cloud representations, then apply the well-developed LiDAR or multi-view based 3D object detectors to the point cloud. These methods with hybrid structures can achieve impressive detection accuracy, but also have inevitable limitations. They introduce additional expensive computational cost for predicting high-resolution depth maps from images, making them hardly feasible in mobile platforms where the energy and computing resource are limited. Furthermore, pixel-level depth estimation does not aim at predicting depth of objects of targeting classes but all pixels in the whole images. The uncertainty in irrelevant or unreliable pixels could bring precision loss into the final 3D object detection. Apart from the pixel-level depth based approaches, there is another stream of methods [9], [14], [19], [20], [21] that first predict the sparse 2D representations such as keypoints and 2D bounding boxes, then utilize optimization to fit a 3D bounding box, which can be very efficient. However, when the object is truncated by the image boundaries, the sparse 2D representations are partly missing, which impose significant challenges to fitting the 3D bounding box. In light of this, we will not rely on post-processing in our object detection framework. In addition, we hope that our framework can be free of object proposals to reduce the computational burden and improve the simplicity and generality.

In this paper, we present MonoGRNet, a general framework for learning Monocular 3D object detection by Geometric Reasoning. Taking a monocular RGB image as input, MonoGRNet predicts the 3D bounding boxes of objects in a single forward pass and can be implemented in a straightforward way. The proposed framework decouples the 3D object detection task into four progressive sub-tasks that are easy to solve using a monocular image. The four sub-tasks are 2D object detection, instance-level depth estimation, projected 3D center estimation and local corner regression, which are solved in parallel in the proposed unified network shown in Figure 1. The network starts from 2D perception and then extends the geometric reasoning into 3D space. The instance-level depth estimation is crucial for bridging the 2D-to-3D gap, which is different from the computationally expensive pixel-level depth estimation utilized by many previous methods. Instance-level depth is defined as the depth of the 3D bounding box center, which can be interpreted as the depth of an object. Using the predicted instance-level depth, we backproject the estimated projected 3D center from the 2D image plane to 3D space to
obtain the 3D center location of the object. At the same time, the local corner regression provides the coordinates of the eight corners of the 3D bounding box with respect to its 3D center.

The proposed framework is general for variable learning schemes since sub-tasks are decoupled loosely. It can be extended from fully supervised learning to a weakly supervised learning scenario, where the ground truth 3D bounding boxes are not available in training but we have the ground truth 2D bounding boxes instead. Labeling 2D bounding boxes saves much labor compared to labeling 3D bounding boxes based on the investigation [22]. This helps to improve the feasibility and efficiency of applying monocular 3D object detectors to various scenarios. In addition, widely used 3D shape datasets, such as PASCAL3D+ [25] and ShapeNet [24], and their corresponding images make it easy to learn view angles of centered objects in a local extent. We present a geometric-guided method to learn the 3D location from labeled 2D bounding boxes and unlabeled frames, as well as an object-centric transfer learning method to learn the local corner regression with easily accessible 3D shape datasets. As we will see, the task decomposition that we propose is a crucial enabler of the flexible extension to weakly supervised learning of monocular 3D object detection. The network structure and loss functions mostly remain unchanged in such an extension.

Our experiments are conducted in three public datasets, including KITTI [25], CiteScapes [26] and MS COCO [27]. Notice that CiteScapes [26] and MS COCO [27] do not provide ground truth 3D bounding boxes. Most of the existing 3D object detectors require full supervision, so these two datasets have not gained attention of the 3D object detection community. Our quantitative results on KITTI [25] and qualitative results on all the three datasets demonstrate the promising performance of the proposed framework and its strong generalization capability to diverse training and testing scenarios. In summary, our contributions are:

- We propose to decompose the monocular 3D object detection task into four sub-tasks including 2D object detection, instance-level depth estimation, projected 3D center estimation and local corner regression. Such formulation frees the detector from object proposals and computationally expensive pixel-level dense depth estimation used by previous methods.
- We propose a unified network to efficiently solve the four sub-tasks in parallel, which can be trained in an end-to-end fashion. We also present a method to train the network using ground truth 2D bounding boxes and easily accessible additional data when the ground truth 3D bounding boxes are unavailable.
- We conduct comprehensive experiments on the KITTI [25], Cityscapes [26] and MS COCO [27] datasets, demonstrating the advantage of the proposed framework in diverse scenarios. We also conduct ablation studies to examine the effectiveness of the crucial components in our framework.

A preliminary version of our MonoGRNet has been published [7] and gained attention. In this manuscript, we make the following improvement. 1) We simplify the structure of MonoGRNet to emphasize the most important concepts that we propose, and at the same time the quantitative performance is improved. 2) We extend the framework from fully supervised to weakly supervised learning, and present an effective method to train the network using ground truth 2D bounding boxes and easily accessible additional data when the ground truth 3D bounding boxes are not available during training. 3) We provide extensive quantitative and qualitative experimental results to show the performance of our framework in different settings and examine the effectiveness of the key modules in ablation studies.

2 RELATED WORK

2.1 2D Object Detection

2D object detection deep networks are extensively studied. Region proposal based methods [28], [29] generate impressive results but perform slowly due to complex multi-stage
pipelines. Another group of methods [30], [31], [32], [33] focusing on fast training and inferencing apply a single stage detection. Multi-net [34] introduces an encoder-decoder architecture for real-time semantic reasoning. Its detection decoder combines the fast regression in Yolo [30] with the size-adjusting RoiAlign of Mask-RCNN [35], achieving a satisfied speed-accuracy ratio. All these methods predict 2D bounding boxes of objects while none 3D geometric features are considered.

2.2 3D object detection

Existing methods range from single-view RGB [8], [16], [36], [37], multi-view RGB [4], [38], to RGB-D [5], [39], [40]. When geometric information of the depth dimension is provided, the 3D detection task is much easier. MV3D [4] generates 3D object proposals from bird’s eye view maps given LiDAR point clouds, and then fuses features in RGB images, LiDAR front views and bird’s eye views to predict 3D boxes. AVOD [6] fuses the RGB and LiDAR information in the region proposal stage to reduce the missed detections. Given RGB-D data, F-PointNet [5] extrudes 2D region proposals to a 3D viewing frustum and then segments out the point cloud of interest object. Recently proposed state-of-the-art 3D object detectors include STD [41], Part-A2 Net [42] and UberATG-MMF [43].

The most related approaches to ours are using a monocular RGB image. Information loss in the depth dimension significantly increases the task’s difficulty. Performances of state-of-the-art such methods still have large margins to RGB-D and multi-view methods. Mono3D [44] leverages segmentation mask and contextual information to generate 3D object proposals. Mono3D++ [17] exploits pseudo 3D keypoints and shapes prior to localizing objects in 3D by minimizing matching loss. [6] combines monocular 3D detection with tracking in autonomous scenarios. MonoGRNet [7] proposes instance-level depth estimation to extend the 2D perception to 3D reasoning without reluctant intermediate representations. MonoPSR [15] leverages shape reconstruction from monocular images to facilitate 3D detection. Another line of research [10], [11] uses monocular images to generate pseudo point cloud, which is passed to existing point cloud based 3D detectors. Nevertheless, since extensive 3D labels are difficult to obtain in practice, the fully supervised methods have limitations in real-world applications.

2.3 Monocular Depth Estimation.

Pixel-level depth estimation networks [11], [45], [46] have been proposed. However, when regressing the pixel-level depth, the loss function takes into account every pixel in the depth map and treats them without significant difference. In a common practice, the loss values from each pixel are summed up as a whole to be optimized. Nevertheless, there is a likelihood that the pixels lying in an object are much fewer than those lying in the background, and thus the low average error does not indicate the depth values are accurate in pixels contained in an object. In addition, dense depths are often estimated from disparity maps that may produce large errors at far regions, which may downgrade the 3D localization performance drastically. Different from the pixel-level depth estimation methods, we propose an instance-level depth estimation (IDE) network that predicts the 3D center depth of objects. IDE does not require the densely labeled pixel-level depth for training and avoids the computationally expensive pixel-level monocular depth estimation in testing.

Instance-level depth has been studied in [47], [48], [49]. In [47], the depths of objects are regressed via a single fully convolutional neural network. In [48], the instance depth ordering is jointly learned with instance segmentation. In [49], the authors proposed a self-supervised method to learn instance depth from video sequences. To the best of our knowledge, our MonoGRNet is the first to introduce instance-level depth estimation in the context of monocular 3D object detection. The instance-level depth enables the back-projection from 2D to 3D to obtain the 3D center locations of targeted objects.

2.4 Weakly supervised object detection

Most existing studies focus on 2D object detection, while weakly supervised 3D detection has not been extensively explored. [50] starts by inferring the geometric structure implied in low-level and middle-level features to describe the objects of interest, then learns from high-level features by iterative training to detect objects in 2D. [51] trains the detection network by iteratively selecting a subset of bounding boxes that are the most reliable. ACol [52] utilizes the output heatmaps for complementary learning. [53] trains the weakly supervised localization network first on easy examples and then on hard ones. [54] localizes objects by clustering. Different from these previous studies on 2D detection, we aim at bridging the gap between weakly supervised learning and 3D object detection.

2.5 Transfer learning on object detection

One popular usage of transfer learning in object detection is model parameter initialization, where the networks trained on large-scale image recognition datasets are used as CNN backbones in object detectors [55], [56], [57], [58], [59]. Lim et al. proposed to borrow examples from existing classes to learn to detect objects from unseen classes [50]. Lampert et al. propose to use high-level attributes to detect object classes, making it easier to transfer to unseen classes [60]. Tang et al. incorporate the visual and semantic similarities of objects in the transferring process. An important step in transfer learning is narrowing the gap between the source and the target dataset. Previous work has focused on domain adaptation [62], [63], [64] at an image level, while we take a different route by ignoring the background and targeting at the object level, saving consideration on the gap between the source and target datasets in terms of the object sizes and background scenes in transfer learning.

3 Problem Statement

Given a monocular RGB image, the objective is to detect and localize objects of specific classes in the 3D space. A target object is represented by a class label and a 3D bounding box, which bounds the complete object regardless of occlusion or truncation. A 3D bounding box $B_{3D}$ is defined by a 3D
fully and weakly supervised scenarios. In this section, we first describe the four sub-tasks.

- (1) 2D object detection
- (2) instance-level depth estimation
- (3) projected 3D center estimation
- (4) local corner regression

The 3D bounding box prediction result can be directly derived by combining the output of the four sub-tasks. In this section, we first describe the four sub-tasks and the network structure, then detail the learning process in fully and weakly supervised scenarios.

4 Method

We propose MonoGRNet, a general framework for monocular 3D object detection. MonoGRNet takes a monocular image as input and outputs the 3D bounding boxes of objects in a single forward pass, and is free of the object proposal stage and the computationally expensive pixel-level depth prediction. The framework adapts to both fully supervised and weakly supervised learning. Since directly predicting 3D bounding boxes from a 2D monocular image could be difficult due to the dimension reduction, we propose to decompose the task into four progressive sub-tasks that are easier to solve using a monocular image, even when the ground truth 3D bounding boxes are not available. The sub-tasks are (1) 2D object detection, (2) instance-level depth estimation, (3) projected 3D center estimation and (4) local corner regression. The 3D bounding box prediction result can be directly derived by combining the output of the four sub-tasks. In this section, we first describe the four sub-tasks and the network structure, then detail the learning process in fully and weakly supervised scenarios.
bounding box \( B_{3D} \) in a local coordinate system described in Section 3 and Figure 2. To summarize, the first task gives \( B_{2D} \) and the object class, the second and third tasks give \( C \) and \( O \), and the forth task gives \( O \). \( C \) and \( O \) completely parameterize the \( B_{3D} \), which is the final output. Due to the well-decomposed task, our framework can also be extended from fully supervised learning to weakly supervised learning, which will be detailed in Section 4.4.

4.2 Network Structure

We design a unified end-to-end network to efficiently solve the four sub-tasks in parallel, which is illustrated in Figure 1. The network takes a monocular image as input and outputs a set of \( B_{3D} \) corresponding to the objects on the image. The four sub-tasks share the same backbone and only differ in head layers, which enable feature reuse and improve the inference efficiency. Compared to previous monocular 3D object detectors [10], [11], [15], [16], the proposed network does not require the dense pixel-level depth prediction, instance segmentation or ground plane estimation, and is free of extensive object proposals [29].

The input image is divided into an \( S_u \times S_o \) grid \( G \), where a cell is indicated by \( g \). The image is passed to a fully convolutional backbone network, whose output feature map is also of size \( S_u \times S_o \), followed by the head layers of the sub-tasks. The head layers do not down-sample the feature maps, so the resolution remains \( S_u \times S_o \) in total. The final prediction of the network is obtained by applying non-maximum suppression to the \( B_{3D} \).

4.3 Fully Supervised Learning

In fully supervised learning, the ground truth \( B_{3D} \) is provided, and the corresponding \( B_{2D} \) is obtained by projecting the \( B_{3D} \) to the 2D image plane. Here we formally formulate the loss functions of the sub-task under fully supervision. In this subsection, to distinguish between the network prediction and ground truth, the ground truth is modified by the \((\cdot)\) symbol in the loss functions.

We start from assigning ground truth to each grid \( g \) in the \( S_u \times S_o \) grid. A ground truth object is assigned to a cell \( g \) if the distance between the 2D bounding box and \( g \) is less than \( \sigma_{slice} \). If a \( g \) is assigned to multiple objects, we only choose the object with the smallest instance-level depth \( Z_c \).

In a frame, some \( g \) does not have any ground truth because they are too far from the objects. These \( g \) are considered as background, while the remaining is foreground. We use \( FG \) to denote the set of foreground \( g \).

The classification output is trained using softmax cross entropy (CE) loss and the 2D bounding box regression is trained by L1 distance loss:

\[
\mathcal{L}_P = \sum_{g \in G} CE(P^g, \hat{P}^g)
\]

\[
\mathcal{L}_{B_{2D}} = \sum_{g \in FG} ((u^g_{2D} - \hat{u}^g_{2D}) + (h^g_{2D} - \hat{h}^g_{2D}) + |\Delta u^g_{2D} - \Delta \hat{u}^g_{2D}| + |\Delta h^g_{2D} - \Delta \hat{h}^g_{2D}|)
\]

The instance-depth estimation, projected 3D center estimation and local corner regression are trained with L1 loss:

\[
\mathcal{L}_{Z_c} = \sum_{g \in FG} |Z_c^g - \hat{Z}_c^g|
\]

\[
\mathcal{L}_c = \sum_{g \in FG} ((|\Delta u^g_{c} - \Delta \hat{u}^g_{c}| + |\Delta v^g_{c} - \Delta \hat{v}^g_{c}|)
\]

\[
\mathcal{L}_O = \sum_{g \in FG} \sum_{k=1}^{8} |O_k^g - \hat{O}_k^g|
\]

The 3D bounding box refinement is trained with L1 loss:

\[
\mathcal{L}_{\Delta C} = \sum_{g \in FG} |\Delta C^g - (\hat{C}^g - C^g)|
\]

\[
\mathcal{L}_{\Delta O} = \sum_{g \in FG} \sum_{k=1}^{8} |\Delta O_k^g - (\hat{O}_k^g - O_k^g)|
\]

Finally, we sum up the losses to produce the final loss function to be minimized.

4.4 Weakly Supervised Learning

In weakly supervised learning, we consider the scenario where the training set provides 2D bounding boxes instead of 3D bounding boxes, as is stated in Section 3. We also assume that 3D data such as point clouds or depth maps are not available in neither training nor testing. As the ground truth 2D bounding boxes are provided, we use the same loss...
functions to train the 2D object detection branch as is in Section 4.3. The main challenge is to learn the remaining three tasks. For instance-level depth estimation and projected 3D center estimation, we propose the geometry-guided learning method, which makes full use of the projective geometry to guide the learning process using 2D bounding boxes and unlabeled frames. For local corner regression, we propose the object-centric transfer learning method that can effectively transfer the knowledge from another easily accessible dataset to the target object detector to facilitate the learning process.

4.4.1 Geometry-Guided Learning of 3D Location

We consider the second and third tasks, instance-level depth estimation and projected 3D center estimation. Their goal is to obtain the 3D location $C$. Using ground truth 2D bounding boxes as supervision, we propose to leverage the projective geometry and unlabeled frames to learn the two tasks. Denote the prior height of the object as $h_{3D}$ and the camera focal length along the $v$-axis as $f_v$. Since we have the height $h_{2D}$ of the ground truth $B_{2D}$, we can calculate a rough instance-level depth $Z_{c} = f_v \cdot h_{3D} / h_{2D}$ that is illustrated in Figure 4 (a). We regard $Z_c$ as a pseudo ground truth $Z_c$ and use it to replace $Z_c$ in Equation 3 to train the network. We remove the subscript $g$ for simplicity. Also, we regard $\hat{c} = (u_0, v_b)$, the center of the ground truth $B_{2D}$, as a pseudo ground truth $Z_c$ and use it to replace $Z_c$ in Equation 3 to train the network. Given that the pseudo ground truth is only a rough approximation, we will refine the network predictions soon.

A rough 3D location $C$ can be calculated from the estimated $Z_c$ and $c$ of the grid cell using Equation 1. Then we refine this prediction by regressing $\tilde{C} = (\tilde{X}_c, \tilde{Y}_c, \tilde{Z}_c)$, so that $\Delta \tilde{C} + C$ is closer to the real 3D location. Since the 3D ground truth is absent, we propose to utilize the first-order approximation of $\Delta \tilde{C}$ to train the network. Here we assume we already have the local corners $O$ and will explain how to obtain $O$ in Section 4.4.2. A coarse 3D bounding box $B_{3D}$ can be determined by $C$ and $O$. By projecting $B_{3D}$ to the image, we obtain a projected 2D bounding box, then we subtract it from the ground truth 2D bounding box to get $\Delta B_{2D} = (\Delta u_{2D}, \Delta h_{2D}, \Delta v_{u}, \Delta v_{h})$. The approximated $\Delta \tilde{C}$ is formulated as:

$$\Delta \tilde{C} = (\frac{\partial X_c}{\partial u_a} \Delta u_a, \frac{\partial Y_c}{\partial v_a} \Delta v_a, \frac{\partial Z_c}{\partial h_{2D}} \Delta h_{2D})$$

(8)

where the partial derivatives are:

$$\frac{\partial X_c}{\partial u_a} \approx \frac{Z_c}{f_u} \frac{\partial Y_c}{\partial v_a} \approx \frac{Z_c}{f_v} \frac{\partial Z_c}{\partial h_{2D}} = -f_v \cdot h_{3D}$$

(9)

In Equation 6 we replace $(\tilde{C}^g - C^g)$ with $\Delta \tilde{C}$ to train the network. The previous method Deep3DBox [9] minimizes the re-projection discrepancy to obtain 3D locations, but it regards the optimization as post-processing and has to predict the 2D bounding boxes before calculating the 3D bounding boxes in inference. Our approach uses the 2D bounding boxes to endow the network with the ability of 3D location estimation directly from image inputs, which means the pipeline can predict the 3D bounding boxes in an end-to-end fashion.

If neighboring image frames, such as video data, are available, we can impose another regularization based on a real-world kinematics prior that the acceleration of objects should be limited to a certain threshold. Figure 4 (b) shows the motion of an object across frames. We formulate this acceleration constraint as:

$$\mathcal{L}_a = \sum_{k=1}^{K} \sum_{n \in FN_k} \text{clip}(|a^k_n| - \alpha_a, 0, \beta_a)$$

$$a^k_n = \frac{v^k_n - v^k_{n+1}}{t_n - t_{n+1}} \quad v^k_n = \frac{E_{FG,k,n}[C] - E_{FG,k,n+1}[C]}{t_n - t_{n+1}}$$

(10)

where $FN_k$ is the set of indices of the frames in which object $k$ is present, $FG_{k,n}$ refers to the set of foreground cells of object $k$ in the $n$th frame, $a^k_n$ and $v^k_n$ are the instantaneous acceleration and velocity of the object relative to the camera, $t_n$ indicates the corresponding time. The gradient from $\mathcal{L}_a$ back-propagates to $Z_c$ and $\hat{c}$ through $C$. $\tilde{c}_a$ equals to zero if $|a^k_n|$ is less than the threshold $\alpha_a$, $\beta_a$ to clip the loss to avoid instability. We choose $\alpha_a = 0.3$ and $\beta_a = 3.0$ by grid search. We do not need the ground truth 2D bounding boxes in frame $2, 3, \cdots, N$, since the foreground region in these frames are estimated using the bounding boxes in the first frame and the inter-frame optical flow that is obtained using the off-the-shelf PWC-Net [65]. Note that motion consistency is employed in training, while a single image is enough in inference.

Note that the geometric-guided learning of 3D location detailed above requires the prior size (or at least the prior height) of the object and the camera intrinsics, including the principle point $(p_u, p_v)$ and focal lengths $(f_u, f_v)$. Each type of objects is assigned a prior size that equals to the average size of that type of objects. In terms of the camera parameters, since $(p_u, p_v)$ is always very close to the image center, when the camera intrinsics are unknown such as in MS COCO [27] dataset, the principle point is set to be the image center. For $(f_u, f_v)$, we set $f_u$ to be 0.8 times the image width and $f_v = f_u$, which is also adopted by [66]. If the users can provide an accurate size (height denoted as...
In this way, we eliminate the interference of background distributions and object sizes, making each instance scale-invariant. The 2D bounding boxes are obtained via off-the-shelf object detectors \cite{29} if they are not labeled.

The second step is transferring the knowledge to our monocular 3D object detector. Using the ground truth 2D bounding box, we crop out the objects and scale them to the same size that the teacher is trained. The teacher predicts the view angle of each object online. Using the annotated 2D bounding box \( B_{2D} = (w_{2D}, h_{2D}, u_p, v_p) \) and camera intrinsics, we convert the view angle to orientation. In our setting, only the orientation on the ground plane is considered. Let \( \varphi \) be the view angle around the axis perpendicular to the ground plane, the orientation \( \theta \) can be analytically calculated as \( \theta = \varphi - \arctan(\frac{h_{3D}}{w_{3D}}) \), where \( u_p \) is the horizontal principal point and \( f_u \) is the horizontal focal length of the calibrated monocular camera. Then we use the orientation and the prior size to compute the approximated ground truth \( \hat{O}_k \) to supervise the targeted 3D detector. We replace \( O_k \) with \( \hat{O}_k \) in Equation 5 to train the network.

4.4.2 Object-Centric Transfer Learning

We consider the fourth task, local corner regression. The ground truth corners \( O_k \) are not contained in the labeled \( B_{3D} \) and should be learned by introducing another source of knowledge. Note that the corners can be computed from the size and orientation of the \( B_{3D} \). For simplicity, we use the prior size of each class of objects, so the unknown parameters are reduced to the orientation. It is noticed that there are many easily accessible datasets (e.g., PASCAL3D+ \cite{23} and ShapeNet \cite{24}) annotated with object view angles. We present an object-centric transfer learning method (see Figure 5) to use the additional data to train the local corner regression branch, which saves labor-intensive annotation on new datasets. We use PASCAL3D+ \cite{23} as an example source dataset. It should be noted that only the ground truth view angle is required, which means the source dataset needs not to be annotated with the complete 6-DoF poses as is in PASCAL3D+ \cite{23}.

The first step is training a teacher network on the source dataset \cite{23} using its ground truth view angle. The teacher regresses the cosine and sine values of the view angle. Full annotations of 6-DoF poses are not required. The most critical operation is cropping out the objects and scaling them to a fixed size (we use \( 64 \times 64 \)) before feeding to the network. In this way, we eliminate the interference of background view angle to supervise the targeted 3D detector. We replace \( O_k \) with \( \hat{O}_k \) in Equation 5 to train the network.

5 EXPERIMENT

Different from most of the previous studies on 3D object detection, our evaluation is not limited to datasets that offer ground truth 3D bounding boxes for training. In addition to the popular KITTI \cite{25} dataset, we also experiment on the challenging Cityscapes \cite{26} and MS COCO \cite{27} dataset where the 3D bounding boxes are not provided. We evaluate both our fully supervised MonoGRNet (F) and weakly supervised version Mono3D (W).

5.1 Implementation

The whole framework is implemented using Python \cite{67} and Tensorflow \cite{68}. We employ VGG16 \cite{69} pretrained on ImageNet \cite{70} as the backbone network in MonoGRNet. We remove the original fully connected layers in VGG16 \cite{69} to obtain a fully convolutional backbone. For hyperparameters, we choose \( S_u \times S_v = 39 \times 12 \), which is the size of the grid \( G \), or namely, the feature map resolution of the head layers. The whole network is trained using Adam \cite{71} optimizer for 40 epochs with a constant learning rate of \( 10^{-5} \). L2 regularization is applied to model parameters at a decay weight of \( 5 \times 10^{-5} \).
5.2 Qualitative Results

Previous work could train 3D object detectors on KITTI [25] but not on Cityscapes [26] and MS COCO [27], since the last two do not offer labeled 3D bounding boxes. Our approach is not limited by 3D labels and can handle all three datasets.

5.2.1 On KITTI Dataset.

The detection results in various scenarios are shown in Figure 7, including the residential areas and the highways, with objects at short and long distances. The detector is robust enough to corner cases including strong light, shadows, truncation and occlusion. This robustness is crucial in practical use. Comparing to the fully supervised MonoGRNet (F), the weakly supervised MonoGRNet (W) exhibits promising qualitative performance.

5.2.2 On MS COCO Dataset.

MS COCO [27] dataset contains a wide variety of objects. Both the indoor and outdoor scenes are included. Although MS COCO does not provide ground truth 3D bounding boxes, the proposed method is not subject to this limitation and can learn from the 2D bounding box labels. The dataset does not offer the camera parameters, which
Our approach demonstrates potentials in general 3D object detection from a monocular image. We use our weakly supervised MonoGRNet (W) on MS COCO, which does not provide ground truth 3D bounding boxes to train the fully supervised MonoGRNet (F).

The experiment was conducted using MonoGRNet (W). We do not apply the motion consistency loss since the neighbouring frames are inaccessible. We only test MonoGRNet (W) since the ground truth is not available for training MonoGRNet (F). The visualization results are presented in Figure 8. This experiment shows the potentials of our method on general 3D object detection at low cost, i.e., based on cheap cameras can be deployed in various scenarios.

We test our MonoGRNet (W) on Cityscapes [26] dataset without using any label on Cityscapes [26] in training. We first train a Faster-RCNN [29] 2D object detector on the KITTI dataset and directly apply it to Cityscapes to obtain 2D bounding boxes, then our network learns 3D bounding boxes using the detected 2D bounding boxes using our weakly supervised method. The proposed method exhibits stable performance as is shown in Figure 9. This experiment demonstrates that we do not necessarily require manually labeled 2D bounding boxes when transferring to new datasets.
Recall

### 3D Object Detection Precision

**BEV Object Detection Precision**

Recall

Fig. 10: Recall-precision curve of 3D and BEV object detection on KITTI val set. F and W are short for full and weak supervision. The 3D IoU threshold is 0.3. Although MonoGRNet (W) is learned from the weak supervision of 2D bounding boxes, it exhibits promising performance compared to methods with full supervision from 3D bounding boxes.

#### 5.3 Quantitative Results

The quantitative experiments are conducted on the KITTI dataset but not on the Cityscapes and MS COCO where the ground truth 3D bounding boxes are not provided for quantitative evaluation. The evaluation criteria include the commonly used 3D average precision $AP_{3D}$, the bird’s eye view (BEV) average precision $AP_{BEV}$ and the average orientation similarity $AOS$. $AP_{3D}$ measures the intersection of union (IoU) between the ground truth 3D bounding box and the predicted one. If the IoU is greater than a given threshold, the ground truth is successfully recalled. A higher IoU threshold means the ground truth is more difficult to recall. Different from $AP_{3D}$, $AP_{BEV}$ ignores the vertical dimension, measuring the IoU from the bird’s eye view. $AOS$ measures the average orientation similarity between the predicted 3D bounding boxes and the ground truth. The calculation of $AOS$ is similar to that of average precision and more details can be found in Section 2.5 of [25]. All the three criteria have been widely accepted [4], [38] in the assessment of 3D object detectors. We follow the publicly available train-val split [4], [38] on KITTI [25] 3D object detection dataset.

#### 5.3.1 Comparison to Fully Supervised Methods

To the best of our knowledge, this is a pioneering work unifying the fully and weakly supervised learning of monocular 3D object detection. Most of the published work [7], [9], [44] only use fully annotated 3D bounding boxes as supervision. Figure 6 compares the speed and average precision, showing that MonoGRNet (F) requires the least computational time while achieving promising average precision.

The weakly supervised MonoGRNet (W) even demonstrates similar average precision with the recent full supervised method FQNet [72]. Figure 10 and Table 1 and Table 2 compare the $AP_{3D}$ and $AP_{BEV}$. Figure 12 illustrates the localization error with respect to distances. It is shown that we have promising performance compared to these approaches. Even though there is still a gap between the weakly and fully supervised methods, we believe the potentially large amount of weakly labeled data can further narrow the performance gap. In addition, 3D labels for large-scale real datasets are difficult to obtain in practice, and our method has the potentials to be an alternative approach to saving the annotation cost and promote the feasibility of monocular 3D object detection in future applications.

In Table 4, we also compare to the most recent state-of-the-art fully supervised methods [11], [15], [73]. Although these methods demonstrate better $AP_{3D}$ than the fully supervised MonoGRNet, our results are noteworthy due to three facts. First, these methods [11], [15], [73] on monocular 3D object detection utilize the 3D LiDAR point clouds as a powerful supervision to learn dense 3D reconstructions that improve the detection performance, while MonoGRNet targets at a more general scenario where we do not assume the existence of point clouds or dense depth data in any part of the training phase. In fact, a very large proportion of monocular images in the real world are acquired without 3D point clouds. Second, the computational cost and time consumption of these methods [11], [15], [73] are much more than our MonoGRNet. It is reported that the 3D reconstruction module [74] used by [73] takes $\sim$500 ms for each frame during inference, but MonoGRNet only requires
Fig. 11: Recall-precision curve of 3D object localization on KITTI val set. F and W are short for full and weak supervision. The distance thresholds are set to 1m, 2m and 3m, corresponding to the first, second and the third row.

Fig. 12: 3D localization error with respect to the ground truth distance on KITTI val set. F and W are short for full and weak supervision. MonoGRNet (F/W) demonstrates a robust localization performance.

∼60ms under the same GPU configuration. Third, one of the main contributions of this paper is that we unified fully supervised and weakly supervised monocular object detection in a single framework, which can flexibly adapt to various scenarios no matter whether the fully annotated 3D bounding boxes are available in training. To the best of our knowledge, this work is the first that achieves this.
TABLE 1: Average precision of 3D object detection on KITTI val set compared to previous fully supervised methods. A ground truth object is successfully recalled if the 3D intersection of union (IoU) between its true 3D bounding box and the predicted box is no less than a certain threshold. Recalling an object in 3D is much more difficult than in 2D image given the extensive searching space and the lack of 3D data.

| Method             | SP     | AP_{3D} (IoU=0.1) | AP_{3D} (IoU=0.2) | AP_{3D} (IoU=0.3) | AP_{3D} (IoU=0.5) | AP_{3D} (IoU=0.7) |
|--------------------|--------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                    | Easy   | Mod.   | Hard  | Easy   | Mod.   | Hard  | Easy   | Mod.   | Hard  | Easy   | Mod.   | Hard  |
| Mono3D [44]        | Full   |        |       | 54.71  | 43.67  | 39.59 | 41.21  | 33.40  | 28.89 | 28.29  | 23.21  | 19.49 |
| Deep3DBox [9]      | Full   |        |       | 82.57  | 69.74  | 60.86 | 69.97  | 56.62  | 48.59 | 54.30  | 43.42  | 36.57 |
| MF3D [36]          | Full   |        |       |        |        |       |        |        |       |        |        |       |
| FQNet [72]         | Full   |        |       |        |        |       |        |        |       |        |        |       |
| ROI-10D [75]       | Full   |        |       |        |        |       |        |        |       |        |        |       |
| MonoGRNet (F)      | Full   |        |       | 88.10  | 76.89  | 67.76 | 82.18  | 64.54  | 55.63 | 73.66  | 61.74  | 47.75 |
| MonoGRNet (W)      | Weak   |        |       | 80.70  | 64.42  | 55.42 | 69.60  | 53.70  | 45.45 | 56.16  | 42.61  | 35.36 |

TABLE 2: Average precision of bird’s eye view (BEV) object detection on KITTI val set compared to previous fully supervised methods. A ground truth object is successfully recalled if the BEV intersection of union (IoU) between its true 3D bounding box and the predicted box is no less than a certain threshold.

| Method             | SP     | AP_{BEV} (IoU=0.1) | AP_{BEV} (IoU=0.2) | AP_{BEV} (IoU=0.3) | AP_{BEV} (IoU=0.5) | AP_{BEV} (IoU=0.7) |
|--------------------|--------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                    | Easy   | Mod.   | Hard  | Easy   | Mod.   | Hard  | Easy   | Mod.   | Hard  | Easy   | Mod.   | Hard  |
| Mono3D [44]        | Full   |        |       | 55.47  | 44.25  | 40.14 | 46.26  | 35.22  | 34.23 | 32.76  | 25.15  | 23.65 |
| Deep3DBox [9]      | Full   |        |       | 84.29  | 70.40  | 61.49 | 71.27  | 58.88  | 49.92 | 57.14  | 47.20  | 39.06 |
| MF3D [36]          | Full   |        |       |        |        |       |        |        |       |        |        |       |
| FQNet [72]         | Full   |        |       |        |        |       |        |        |       |        |        |       |
| ROI-10D [75]       | Full   |        |       |        |        |       |        |        |       |        |        |       |
| MonoGRNet (F)      | Full   |        |       | 88.22  | 77.09  | 67.91 | 83.27  | 65.13  | 56.18 | 74.92  | 63.22  | 54.65 |
| MonoGRNet (W)      | Weak   |        |       | 81.12  | 64.76  | 55.76 | 71.38  | 60.11  | 46.27 | 58.61  | 48.75  | 41.49 |

TABLE 3: Average precision of 3D and bird’s eye view (BEV) object detection on KITTI val set compared to weakly supervised baselines. BA and OC are short for background-aware and object-centric respectively. MinProjErr denotes the baseline method derived from [9] to recover 3D location as described in Section 5.3.2. GeoGL indicates the proposed geometry-guided learning of 3D location.

| Orientation | Method            | AP_{3D} / AP_{BEV} (IoU=0.3) | AP_{3D} / AP_{BEV} (IoU=0.5) |
|-------------|-------------------|-------------------------------|-------------------------------|
|             | Easy   | Moderate | Hard  | Easy   | Moderate | Hard  | Easy   | Moderate | Hard  | Easy   | Moderate | Hard  |
| BA × 2      | MinProjErr       | 37.67 / 42.80                | 28.50 / 35.69                | 23.60 / 29.99                |
| BA × 1      | MinProjErr       | 39.30 / 43.92                | 29.80 / 36.39                | 24.40 / 30.53                |
| BA × 2      | GeoGL (Ours)     | 50.10 / 54.31                | 39.16 / 41.24                | 32.57 / 34.35                |
| BA × 1      | GeoGL (Ours)     | 54.44 / 57.36                | 41.40 / 47.59                | 34.39 / 35.77                |
| OC (Ours)   | GeoGL (Ours)     | 56.16 / 58.61                | 42.61 / 48.75                | 35.36 / 41.49                |

TABLE 4: Comparison to fully supervised methods involving LiDAR or only monocular information during training. The numbers are reported on KITTI val set.

| Method          | Time   | Data       | AP_{3D} (IoU=0.5) |
|-----------------|--------|------------|-------------------|
| P-LiDAR [11]    | > 0.5 s| LiDAR+Mono | Mono              |
| MonoPSR [15]    | 0.2 s  | LiDAR+Mono | Mono              |
| Ma et al. [73]  | > 0.5 s| LiDAR+Mono | Mono              |
| FQNet [72]      | > 0.4 s| Mono       | Mono              |
| MonoGRNet (F)   | 0.06 s | Mono       | Mono              |

TABLE 5: Average orientation similarity on KITTI val set. The three approaches use GeoGL to predict the 3D location. The only difference is how they learn object orientations.

| Method | AOS (IoU=0.3) | AOS (IoU=0.5) |
|--------|--------------|--------------|
|        | Easy         | Moderate     | Hard    | Easy   | Moderate | Hard   |
| BA × 2 | 64.23        | 62.86        | 49.96   | 64.14  | 56.93    | 49.85  |
| BA × 1 | 83.58        | 80.54        | 64.13   | 83.56  | 73.09    | 63.97  |
| OC (Ours) | 90.00     | 88.78        | 70.89   | 89.94  | 80.31    | 70.76  |

5.3.2 Comparison to Weakly Supervised Baselines

We also examine our approach by comparing to weakly supervised baselines. As no weakly supervised monocular 3D object detector has been published, we borrowed ideas from previous transfer learning methods [76] and state-of-the-art monocular 3D object detectors [7], [9] to construct baseline approaches that only require annotated 2D bounding boxes in training. In our object-centric transfer learning of orientation, the background is mostly ignored, and the teacher network mainly focuses on the object itself.
TABLE 6: Ablation study on loss functions for weakly supervised method on KITTI val set. The loss functions are crucial in learning the 3D location of objects using their ground truth 2D bounding boxes and the motion consistency across unlabeled frames.

| Loss Configuration | \( AP_{3D} \) (IoU=0.3) |
|--------------------|----------------------|
| \( L_{Zc} \), \( L_c \), \( L_{\Delta C} \), \( L_a \) | Easy | Moderate | Hard |
| ✓ | 38.28 | 28.56 | 24.13 |
| ✓ | 44.50 | 32.33 | 29.10 |
| ✓ ✓ | 52.72 | 40.59 | 33.71 |
| ✓ ✓ ✓ | 54.21 | 41.19 | 34.19 |
| ✓ ✓ ✓ ✓ | 56.16 | 42.61 | 35.36 |

Previous work [76] on transfer learning usually takes the whole image as input, making image-level transfer learning. Other work [77] on 3D object detection also leverages the background information around the objects to improve the detection performance. In the baseline methods, we enlarge the 2D bounding boxes before cropping the image as shown in Figure 5, allowing more background into the detection performance. In the baseline methods, we enlarge the whole image as input, making image-level transfer learning.

The baselines for orientation prediction are shown in Figure 5, allowing more background into the detection performance. In the baseline methods, we enlarge the whole image as input, making image-level transfer learning.

\[
\lambda \to \frac{(D_u^2, D_v^2, b_u, b_v)}{(w^2 D, h^2 D, b_u, b_v)} \text{ is expanded to}
\]
\[
B_{2D} = \left[(\lambda + 1)w^2 D, (\lambda + 1)h^2 D, b_u, b_v\right].
\]

\( B_{2D} \) is truncated by the image boundaries. As \( \lambda \) becomes larger, the baseline is increasingly similar to methods feeding the whole image as input.

In our geometry-guided learning of 3D location, the network learns from the 3D-to-2D projective geometric constrain and the motion consistency. Inspired by Deep3DBox [9], we develop another baseline method to obtain the 3D location from the 2D bounding box. First, we detect the 2D bounding boxes on image. Then, we find the corresponding 3D bounding box locations by minimizing their projection error with the associated 2D boxes. This baseline approach to recovering the 3D location is denoted as MinProjErr in our experiments. Results are shown in Table 5.

In our geometry-guided learning of 3D location, the network learns from the 3D-to-2D projective geometric constrain and the motion consistency. Inspired by Deep3DBox [9], we develop another baseline method to obtain the 3D location from the 2D bounding box. First, we detect the 2D bounding boxes on image. Then, we find the corresponding 3D bounding box locations by minimizing their projection error with the associated 2D boxes. This baseline approach to recovering the 3D location is denoted as MinProjErr in our experiments. Results are shown in Table 5.

It is clear that our geometry-guided learning method can bring significant performance improvement in all the listed criteria. The object-centric transfer learning performs better than the background-aware counterparts.

Recently, the work [78], [79] proposed self-supervised methods to learn a dense monocular depth predictor. The absolute relative error (AbsRel) achieved by these methods are 0.07m for [78] and 0.11m for [79], while the AbsRel of our weakly supervised instance-level depth estimation is only 0.05m. Besides, the dense pixel-level depth predicted by these methods is the depth of object surfaces, while the instance depth predicted by our MonoGRNet is the 3D center depth of an object. The latter can be directed used to compute the 3D center coordinates of the targeted objects via Equation 1, but the former requires additional steps that could bring extra computational cost.

5.3.3 Ablation Study on Loss Functions

The loss functions proposed in Section 4.4.1 are crucial in supervising the 3D location using labeled 2D bounding boxes. In order to examine the effectiveness of the loss functions, we experiment with different loss configurations and present the results in Table 6. If none of \( L_{Zc} \), \( L_c \) or \( L_{\Delta C} \) is applied, we obtain \( C \) by minimizing the re-projection error. It is shown that \( L_{Zc} \) and \( L_c \) can bring a considerable performance gain, while \( L_{\Delta C} \) can further refine the prediction.

6 CONCLUSION

We have presented the MonoGRNet framework for 3D object detection from monocular images. MonoGRNet decomposes the monocular 3D object detection task into four subtasks, which are 2D object detection, instance-level depth estimation, projected 3D center estimation and local corner regression. The task decomposition allows the 3D bounding boxes to be predicted in a single forward pass without the object-proposal stage or the computationally expensive pixel-level depth prediction. We also demonstrate that the framework can flexibly adapt to both fully and weakly supervised learning without changing the network structure or most of the loss functions. Extensive experiments are conducted on three public datasets, KITTI, Cityscapes and MS COCO. Although the last two does not offer ground truth 3D bounding boxes, our framework is still trainable in such a setting. Qualitative and quantitative results have shown the promising performance of our method in diverse scenarios.

REFERENCES

[1] B. Yang, W. Luo, and R. Urtasun, “Pixor: Real-time 3d object detection from point clouds,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 7652–7660.
[2] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4490–4499.
[3] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Bejborn, “Pointpillars: Fast encoders for object detection from point clouds,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 697–12 705.
[4] X. Chen, H. Mâ, J. Wan, B. Li, and T. Xia, “Multi-view 3d object detection network for autonomous driving,” in IEEE CVPR, vol. 1, no. 2, 2017, p. 3. 1, 3, 10
[5] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum pointnets for 3d object detection from rgb-d data,” arXiv preprint arXiv:1711.08488, 2017.
[6] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krähenbühl, T. Darrell, and F. Yu, “Joint monocular 3d vehicle detection and tracking,” arXiv preprint arXiv:1811.10742, 2018.
[7] Z. Qin, J. Wang, and Y. Lu, “Monognet: A geometric reasoning network for monocular 3d object localization,” AAAI, 2019.
[8] F. Chabot, M. Chauchoi, J. Rabarisoa, C. Teulière, and T. Chateau, “Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image,” in Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2040–2049.
[9] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecká, “3d bounding box estimation using deep learning and geometry,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017, pp. 5632–5640.
[10] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving,” ArXiv, vol. abs/1906.06310, 2019.
[11] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Weinberger, “Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving,” in CVPR, 2019.
X. Jing, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo, “Learning depth-guided convolutions for monocular 3d object detection,” arXiv preprint arXiv:1912.04799, 2019.

E. J.侨ovens, C. Zach, and K. Kahl, “Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss,” ArXiv, vol.1906.08070, 2019.

T. He and S. Soatto, “Monod3++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors,” arXiv preprint arXiv:1901.03446, 2019, 1, 3

M. Ding, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo, “Multinet: Real-time joint semantic reasoning for autonomous driving,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11867–11876. 1, 3, 5, 10, 12

C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, “Ssd: Single shot multibox detector,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2147–2156. 1, 3, 5

Y. Xiang, R. Mottaghi, and S. Savarese, “Beyond pascal: A benchmark for 3d object detection in the wild,” in IEEE Winter Conference on Applications of Computer Vision (WACV), 2014, 2, 7

K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” arXiv preprint arXiv:1703.06870, 2017, 3, 5

B. Xu and Z. Chen, “Multi-level fusion based 3d object detection from monocular images,” in Computer Vision and Pattern Recognition (CVPR), 2018, pp. 2345–2353. 3, 12

W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, “Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again,” in Proceedings of the International Conference on Computer Vision (ICCV 2017), Venice, Italy, 2017, pp. 22–29. 3

J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, “Object detection and aggregation neural network for object detection from point cloud,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 779–788. 3

E. Sangineto, M. Nabi, D. Culibrk, and N. Sebe, “Self paced deep learning and high-level feature learning,” in IEEE Transactions on Image Processing, 2015, pp. 6744–6752. 3
Z. Qin, J. Wang, and Y. Lu, “Triangulation learning network: from monocular to stereo 3d object detection,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

M. Braun, Q. Rao, Y. Wang, and F. Flohr, “Pose-rcnn: Joint object detection and pose estimation using 3d object proposals,” in 2019 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016, pp. 1546–1551.

D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan, “Learning features by watching objects move,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2701–2710.

J. Dai, Y. Li, K. He, and J. Sun, “R-rcnn: Object detection via region-based fully convolutional networks,” in Advances in neural information processing systems, 2016, pp. 379–387.

J. J. Lim, R. R. Salakhutdinov, and A. Torralba, “Transfer learning by borrowing examples for multiclass object detection,” in Advances in neural information processing systems, 2011, pp. 118–126.

C. H. Lampert, H. Nickisch, and S. Harmeling, “Learning to detect unseen object classes by between-class attribute transfer,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009, pp. 951–958.

F. Gustafsson and E. Linder-Norén, “Automotive 3d object detection without target domain annotations,” 2018.

J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko, “Lsda: Large scale detection through adaptation,” in Advances in Neural Information Processing Systems, 2014, pp. 3536–3544.

L. Spinello and K. O. Arras, “Leveraging rgb-d data: Adaptive fusion and domain adaptation for object detection,” in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 4469–4474.

D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume,” 2018.

D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, “Single-shot multi-person 3d pose estimation from monocular rgb,” in 2018 International Conference on 3D Vision (3DV). IEEE, 2018, pp. 120–130.

G. Van Rossum and F. L. Drake Jr, Python tutorial. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015.

D. Z. Matthew and F. Rob, “Visualizing and understanding convolutional networks,” in European Conference on Computer Vision, 2014, pp. 818–833.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in International Conference for Learning Representations, 2015.

L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, “Deep fitting degree scoring network for monocular 3d object detection,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, “Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6851–6860.

H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in Conference on Computer Vision and Pattern Recognition (CVPR). 2018.

F. Manhardt, W. Kehl, and A. Gaidon, “Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2069–2078.