Monocular 3D Detection With Geometric Constraint Embedding and Semi-Supervised Training

Peixuan Li and Huaici Zhao

Abstract—In this work, we propose a novel one-stage and keypoint-based framework for monocular 3D object detection using only RGB images, called KM3D-Net. 2D detection only requires a deep neural network to predict 2D properties of objects, as it is a semanticity-aware task. For image-based 3D detection, we argue that the combination of a deep neural network and geometric constraints are needed to synergistically estimate appearance-related and spatial-related information. Here, we design a fully convolutional model to predict object keypoints, dimension, and orientation, and combine these with perspective geometry constraints to compute position attributes. Further, we reformulate the geometric constraints as a differentiable version and embed this in the network to reduce running time while maintaining the consistency of model outputs in an end-to-end fashion. Benefiting from this simple structure, we propose an effective semi-supervised training strategy for settings where labeled training data are scarce. In this strategy, we enforce a consensus prediction of two shared-weights KM3D-Net for the same unlabeled image under different input augmentation conditions and network regularization. In particular, we unify the coordinate-dependent augmentations as the affine transformation for the differential recovering position of objects, and propose a keypoint-dropout module for network regularization. Extensive experiments on the popular KITTI 3D detection dataset indicate that the KM3D-Net surpasses state-of-the-art methods by a large margin in both efficiency and accuracy. And also, to the best of our knowledge, this is the first application of semi-supervised learning in monocular 3D object detection. We surpass most of the previous fully supervised methods with only 13% labeled data on KITTI.

Index Terms—Autonomous Vehicle Navigation, computer vision for automation, object detection, segmentation and categorization.

I. INTRODUCTION

This work focuses on 3D object detection using only monocular RGB images for autonomous driving. 3D object detection plays an essential role in autonomous vehicle perception and robotic navigation. Most of the existing methods heavily rely on LiDAR data to obtain accurate depth information [1]–[3]. However, LiDAR systems have some disadvantages such as a high price, high energy consumption, a short working life, and being adverse to the shape of current vehicles. Alternatively, every car has an onboard monocular camera, which is cost-effective, energy-conserving, and installation-flexible, and is therefore drawing increasing attention from the computer-vision community [4]–[11].

3D detection from only one monocular image is a naturally ill-posed problem for the reason that missing the depth information introduces ambiguities of inverse projection from the 2D image plane to 3D space. Straightforward solutions simultaneously regress object dimension, orientation, and position by means of a convolutional neural network (CNN) [12]–[14]. One central issue in these approaches is the limited capacity of deep learning models to explicitly describe the geometry of objects and map all appearances to the same 3D objects. Recent work [15] introduces a semantic modeling of objects through 3D bounding box (BBox) and point cloud. As a result, this confinement of the geometry model to a bounding box with semantic labels as point constraints is a substantial challenge for computer vision.

Monocular 3D detection with geometric constraints is a naturally ill-posed problem due to the absence of depth information. A naive approach is to estimate object dimension, orientation, and position with deep learning models. However, this approach is challenged by the limited capacity of CNNs to explicitly describe the geometry of objects and map all appearances to the same 3D objects. Recent work [15] introduces a semantic modeling of objects through 3D bounding box (BBox) and point cloud. As a result, this confinement of the geometry model to a bounding box with semantic labels as point constraints is a substantial challenge for computer vision.
the four uncertainty constraints in 2D BBox. Since the variance of the box dimension estimate is typically small and is strongly tied to the appearance of a particular object category, it also can be accurately restored by image features. Orientation can be predicted by the vanishing point computed by the cross-product of the keypoints. Therefore, we propose a learning-based vanishing point to module (VP2O) to predict the orientation, which aggregates keypoint information through the attention mechanism. Appearance-related information is predicted by adding extra parallel branches after feature extraction in a 2D detector [14]. For spatial-related information, we reformulate the nonlinear optimization in the projection space as a task of solving the overdetermined system, which can pass error differentials back to CNN through the singular value decomposition (SVD) operator. Importantly, with our reformulation, the whole framework, which comprises a CNN and geometric reasoning, can maintain consistency between appearance- and spatial-related prediction in an end-to-end fashion during training. Our method extends the ability of a CNN by explicitly embedding a perspective geometric model rather than increasing the depth or width, making it possible to improve accuracy while maintaining high efficiency.

Semi-supervised training has been proven to be effective by image classification in exploiting unlabeled data [18]–[21]. However, it has gained little attention in 3D detection, where labeling data are much more expensive. To bridge this gap, we propose a semi-supervised training method for the scene where 3D annotations are scarce by extending our simple KM3D-Net. Our method is inspired by the II-model [19], a semi-supervised baseline method of image classification. It proposes that different predictions of a good model should be assembled at the same input with different network regularization and input augmentation. In this letter, we show how to extend this idea to the 3D detection task. In particular, we propose a keypoint-dropout that randomly drops the keypoints in our geometric reasoning module for network regularization instead of regular dropout [22]. Our keypoint-dropout can be regarded as an explicit dropout of a known feature. For input augmentation, we formulate coordinate-dependent augmentation as the affine transformation to align the coordinates of one object in two augmentations. These two strategies allow our model to predict a stochastic variable in the same input. Our unsupervised loss penalizes the difference of 3D properties provided by KM3D-Net using the mean square difference.

Our contributions are summarized as follows: 1) A simple and efficient architecture combines the strengths of a CNN and perspective geometry and achieves real-time 3D object detection using only monocular images. 2) A differentiable geometric reasoning module can be embedded in a CNN to estimate spatial-related information while maintaining the consistency of CNN output. 3) A semi-supervised training method exploits unlabeled images in monocular 3D object detection. 4) Experiments on the popular KITTI 3D detection benchmark demonstrate that the proposed method outperforms state-of-the-art methods by large margins in both accuracy and speed.

II. RELATED WORK

Monocular 3D Detection Including Extra Data. Mono3D [5] focuses on generating 3D object proposals by using instance segmentation, object contour, and ground-plane assumptions. To remedy the lack of scene depth, alternative methods [6], [11], [12], [23], [24] incorporate stand-alone depth to extend image information. Given the depth map, these methods detect 3D objects either by joint optimization [11], multi-stage fusion [6], [12], [23], or LiDAR-based methods after transforming them to point clouds [24]. Another family of methods represent cars as keypoints in the form of a wire-frame template, and use external annotated CAD models to produce synthetic data for training, [11], [25]–[27]. Although extra networks and annotated data increase the accuracy of detection, they need labor-intensive annotations work and additional computation in both training and inference time. Such computation may not be appropriate for unmanned vehicles with limited computational resources.

Monocular 3D Detection Using RGB Images Only. Compared with the aforementioned approaches, most recent works try to detect 3D objects from only RGB images. Deep3DBox [7] infers 2D BBox, dimension, and orientation, and generates a 3D position by combining geometric constraints of 3D points and 2D lines. MonoGRNet [8] optimizes the 3D BBox in an end-to-end manner by fusion of 2D detection, instance depth estimation, 3D location estimation, and local corner regression. M3D-PRN [4] proposes a 3D proposal network with depth-aware convolution to simultaneously generate 2D and 3D object proposals, and uses 3D-2D geometric constraints as a post-processing module to improve precision. MonoDIS [9] isolates the group parameters to simplify the training dynamics by disentangling the 3D transformation. All these methods incorporate a 2D proposal network as the base architecture, which causes the bottleneck of inference time. The approach most related to ours, RTM3D [10], uses keypoints as the intermediate and combines the geometries to further refine the estimation inferred by a single-stage network. Although the fast CNN structure is adopted, its geometric constraint module is still time-consuming and non-differentiable which prevents it from jointly optimizing the output of the CNN.

III. MONOCULAR 3D OBJECTS DETECTION

As shown in Fig. 1, our method comprises a fully convolutional network, which predicts appearance-related properties of an object: dimension, orientation, and ordered list of 2D perspective keypoints, followed by a geometric reasoning module(GRM), which implements the point-to-point geometric constraints for 3D position prediction. The complete framework combines the strengths of both CNNs and perspective geometry and is trainable in end-to-end fashion via the standard backpropagation algorithm [28].
A. Appearance-Related Property Prediction.

Our model has a backbone architecture to extract the features of the entire image and a series of detection heads: 1) Main center: Inspired by [10], [14], we associate an object and its properties by a single point at 2D BBox center, called the main center. The head produces a main center heatmap \( M \in R^{h \times w \times c} \), where \( c \) is the object categories. 2) Keypoints: The keypoint detection head estimates nine ordered 2D perspective keypoints projected from the center and corners of the 3D object BBox. These are geometrically and semantically consistent across different instances of certain object categories. We estimate these keypoints by the regression of offset coordinate \( M_k \in R^{h \times w \times 18} \) from the main center. 3) Dimension: The residual value \( \delta D \in R^{h \times w \times 3} \) is regressed to restore object dimensions by \( \bar{H}, \bar{W}, \bar{L} \), followed [9], where \( \bar{H} = 1.63, \bar{W} = 1.53, \bar{L} = 3.88 \) are statistical average sizes in the KITTI dataset. 4) Orientation: The orientation is needed to design the differentiable geometric reasoning module. A separate branch on the CNN backbone has typically been added to predict the orientation [9], [10]. We found that keypoint prediction branches have encoded clues to orientation. Actually, we can compute the orientation directly through \( VP_i = KR_{col(i)}, i \in 1, 2, 3 \), where \( VP_i \) is the vanishing point obtained by the cross-product of the corresponding four keypoint coordinates [29]. A straightforward solution is to utilize predicted keypoints to compute all \( VP \) following least squares optimization. However, this optimization-based method is difficult to design into a differentiable version, and is sensitive to outliers or the quality of predicted keypoint locations. To solve this problem, we propose a vanishing point to orientation module (VP2O) using a learning-based method to robustly aggregate the keypoint locations for orientation estimation. As shown in Fig. 2, we employ the attention mechanism [30], [31] to guide the information passing from the keypoint branch. A keypoint obtains its weight for orientation prediction through \( 1 \times 1 \) convolution and a max/avg pooling layer. The passed information flow is not all useful because it contains noise, so attention can serve as a gate function to decide how much information from keypoints should contribute to orientation prediction. The attention aggregated feature generates the local orientation feature map \( O_i \in R^{h \times w \times 8} \) and trains it following the multi-bin based method [7]. 5) 3D Confidence: The 3D BBox confidence \( P_{3D} \in R^{h \times w \times 3} \) is predicted by the IoU between estimation and ground truth. The quality of the final 3D BBox is also related to the confidence of the main center. We combine these in the Bayesian rule \( Pro = Pro_{2D}^{m} \times Pro_{3D} \) to obtain the final 3D confidence. \( Pro_{2D}^{m} \) are extracted by the corresponding heatmap after the sigmoid function. During the training step, we define the multi-task loss as

\[
L_{sup} = \omega_mL_m + \omega_{kc}L_{kc} + \omega_DLD + \omega_OVERRIDE + \omega_TLT + \omega_{conf}L_{conf}, \tag{1}
\]
where the main center heatmap loss $L_{km}$ is the penalty-reduced focal loss followed [10], [14], [32]. The keypoint offset coordinate loss $L_{kc}$ is a depth-guided $L_1$ loss for dynamically adjusting the punishing coefficient of different scaling objects,

$$L_{kc} = \frac{1}{N} \sum_j \sum_i g(Z_j) \left\| kp_j^c - \hat{kp}_j \right\|_1$$

$$g(Z) = \begin{cases} \alpha Z & \text{if } Z < a \\ \log_{10}(Z + 1 - \alpha) + \alpha a & \text{if } Z \geq a \end{cases}$$ (2)

where we set $\alpha = 0.01$ and $a = 5$ in our experiments. Dimension loss $L_D$ is an ordinary $L_1$ loss with respect to the ground truth. Orientation loss $L_D$ is the multi-bin [7], [14], which splits the orientation field into two bins and employs a hybrid discrete-continuous loss to train each bin. For the confidence loss $L_{conf}$, we adopt the 3D IoU between the predicted 3D BBox and ground truth as the training target $\hat{p}$. We normalize the 3D IoU to be between $[0,1]$ as

$$\hat{p} = \begin{cases} 1 & \text{IoU}_{3D} > 0.75 \\ 0 & \text{IoU}_{3D} < 0.25 \\ 2\text{IoU}_{3D} - 0.5 & \text{otherwise} \end{cases}$$ (3)

and utilize a standard binary cross-entropy loss for training.

**B. Geometric Reasoning Module**

Given the predicted nine keypoints $\hat{kp}$, dimension $\hat{D}$, and orientation $\hat{O}$, the object pose $T$ can be determined by minimizing the re-projection error of keypoints from the 3D BBox corners and center,

$$\hat{T} = \arg \min_T \sum_i \left\| kp'_i(T, \hat{D}, \hat{O}) - \hat{kp}_i \right\|_2,$$ (4)

where $kp'(T, \hat{D}, \hat{O})$ can be defined as

$$kp'(T, \hat{D}, \hat{O}) = K \left[ \begin{array}{c} R(\hat{O}) \times 3 \times 3 \ T^{1 \times 3} \\ 0 \end{array} \right] \text{diag}(\hat{D}) \text{Cor}$$

$$\text{Cor} = \begin{bmatrix} 1/2 & -1/2 & -1/2 & 1/2 & 1/2 & -1/2 & -1/2 & 1/2 & 0 \\ 1/2 & 1/2 & -1/2 & -1/2 & 1/2 & 1/2 & -1/2 & -1/2 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1/2 \end{bmatrix}.$$ (5)

To embed this geometric relationship in KM3D-Net, we normalize $\hat{kp}$ to $kp$ by camera intrinsics to simplify the representation, rearrange 5 to $R(\hat{O}) \text{diag}(\hat{D}) \text{Cor} + T - kp = 0$, and reformulate it as a linear system:

$$\begin{bmatrix} -1 & 0 & \hat{kp}_y^c \\ 0 & -1 & \hat{kp}_y \\ \vdots & \vdots & \vdots \\ -1 & 0 & \hat{kp}_y \\ 0 & -1 & \hat{kp}_y \end{bmatrix}_{18 \times 3}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = G_{18 \times 1} \begin{bmatrix} kp_x \\ kp_y \\ kp_z \end{bmatrix}_3$$ (6)

$$G = \begin{bmatrix} \frac{1}{2} \cos(\theta) + \frac{1}{2} \sin(\theta) - \hat{kp}_x & \frac{1}{2} \cos(\theta) - \frac{1}{2} \sin(\theta) - \hat{kp}_x \\ \frac{1}{2} \cos(\theta) - \frac{1}{2} \sin(\theta) - \frac{1}{2} \hat{kp}_x & \frac{1}{2} \cos(\theta) + \frac{1}{2} \sin(\theta) - \frac{1}{2} \hat{kp}_x \\ -\frac{1}{2} \hat{kp}_x & -\frac{1}{2} \hat{kp}_x & 1 \end{bmatrix}$$ (7)

The position $T = [X, Y, Z]^T$ can be recovered through a pseudo-inverse using the SVD operator. We backpropagate it using matrix calculus [33], [34]. To equip this differentiable geometric reasoning module, the position error $L_T = ||T - \hat{T}||_2$ can be passed through keypoints, dimension and orientation to maintain the inherent consistency between them.

**IV. SEMI-SUPERVISED APPROACH**

We propose a semi-supervised training approach to further improve the model accuracy for the scene where labeled training data are scarce, including labeled and unlabeled data to jointly optimize the model.

Fig. 3 shows unlabeled data training. We use KM3D-Net to evaluate the same input image twice with different augmentation.
and regularization. The unsupervised loss penalizes different estimation for the same object by taking the mean square difference (MSE loss). Input augmentation has coordinate-independent and coordinate-dependent components. The first component is random color jittering. The second includes random horizontal flips, shifting, and scaling. We formulate these operations as the affine transformation to convert or restore the coordinates of the main center and keypoints. This uniform expression of coordinate-dependent augmentation in matrix form can satisfy the network’s differentiability. We propose keypoint dropout for network regularization by randomly dropping keypoints during position prediction. It is important to notice that one keypoint can provide two geometric constraints, and at least two keypoints can compute the position information of three degrees of freedom. Thus to drop some of the nine keypoints is reasonable in computing position information. Keypoint dropout can be regarded as a special case of dropout [22], which has two benefits: 1) it makes the model more accurate in predicting the un-dropout keypoints; and 2) the inference that contains all the keypoints has more generalization power. Input augmentation and keypoint dropout make the same network weights output a right image and used random scaling (between 0.6 to 1.4), random shifting in the image range, and color jittering as data augmentation. In the inference step, after imposing $3 \times 3$ max pooling on the heatmap of the main center and keypoints, we extracted the object location by filtering the main center with a reduced $3 \times 3$ BBox of ground truths in the coordinate frame of the left and right image and used random scaling (between 0.6 to 1.4), random shifting in the image range, and color jittering as data augmentation. In the inference step, after imposing $3 \times 3$ max pooling on the heatmap of the main center and keypoints, we extracted the object location by filtering the main center with a threshold of 0.4, and used it to select the remaining parameters from other output.

V. EXPERIMENTS

A. Dataset and Implementation Details

We evaluated our experiments on the KITTI and nuScenes 3D detection benchmarks. The KITTI dataset contains 7481 labeled training images and 7518 unlabeled test images. Since the ground truth of the test set is not available, we trained and evaluated our model in three splittings [4], [7], [8], [13], [16]: train1, val1; train2, val2; and train, test. The nuScenes dataset is a large-scale 3D detection dataset that contains more than 1000 scenes in Boston and Singapore [35], with 28130 training images and 6019 validation images. We implemented our deep neural network in PyTorch and trained using the Adam optimizer with a base learning rate of 0.0001 for 200 epochs, reduced $10 \times$ at 100 and 180 epochs. We projected the 3D BBox of ground truths in the coordinate frame of the left and right image and used random scaling (between 0.6 to 1.4), random shifting in the image range, and color jittering as data augmentation. In the inference step, after imposing $3 \times 3$ max pooling on the heatmap of the main center and keypoints, we extracted the object location by filtering the main center with a threshold of 0.4, and used it to select the remaining parameters from other output.

B. Results

We report two evaluation metrics on the KITTI dataset: average precision for 3D IoU ($AP_{3D}$) and bird’s eye view ($AP_{BEV}$). Since KITTI uses $AP^{40}$ instead of $AP^{13}$, and existing approaches only report accuracy at 11 points on the val1 or val2 set, we give the average precision under 40 points in the test set and 11 points in the val1 and val2 sets for a fair comparison.

3D Object Detection with Supervised Training on KITTI Dataset. Tables I and II show the published results for $AP_{3D}$ and $AP_{BEV}$, respectively, with fully supervised training on the KITTI dataset. We also report their single image inferencing time and accelerator. We focus on the car class, as have most previous studies. Our method outperformed all previous methods in
TABLE I

COMPARISON OF 3D DETECTION METHODS FOR CAR CATEGORY, EVALUATED BY METRIC $AP_{3D}$ ON $val_1$ / $val_2$ / $test$ SET ON KITTI. EXTRA MEANS THE EXTRA TRAINING DATA. THE BEST RESULTS ARE HIGHLIGHTED IN RED

| Method                     | Extra | Depth | Time  | $IoU > 0.5 (val_1/val_2)$ | $IoU > 0.7 (val_1/val_2/test)$ |
|----------------------------|-------|-------|-------|---------------------------|---------------------------------|
| 3D Object Detection with Supervised Training on nuScenes Dataset. Work on monocular 3D object detection rarely reports the results of nuScenes, because the detection performance of monocular cameras still has a big gap compared to stereo or LiDAR sensors on the KITTI dataset. Our method still surpassed the published work on MonoDIS, as shown in Table III.

Semi-Supervised Approach. For semi-supervised training, we used the $train1$ set and testing set as the labeled and unlabeled images, respectively, and validated the performance on the $val1$ set. Since ours is the first semi-supervised method for monocular 3D detection, we compared our semi-supervised training to supervised training in various experimental settings. Fig. IV shows that our semi-supervised technique performed more efficiently as the amount of labeled data decreased. Note that with our semi-supervised training, only 500 labeled data points could achieve better accuracy than most methods that adopt supervised training with a complete annotation set.

C. Ablation Study

We conducted a series of experiments to validate the contribution of different strategies in our model. A1) GRM. Two settings were considered to verify GRM’s validity. One was only to use GRM in the inference phase and remove GRM of position supervision in the training phase. The other was to remove GRM in all phases. In this case, we added a parallel depth prediction branch after feature extraction and combined the main center to compute position.

A2) Keypoint coordinates of regression vs. heatmap. For prediction keypoints in the heatmap, we added a parallel heatmap branch, whose each channel contained the corresponding semantic points of all objects in the whole picture.

A3) 3D confidence prediction; A4) Data augmentation. Table V shows the $AP_{3D}$ and $AP_{BEV}$ for each variant of the proposed strategies. The performance dropped greatly without our geometric reasoning module, indicating the validity of geometric constraints in spatial-related information prediction. Using heatmaps to estimate keypoints also suffered from performance terms of $AP_{3D}$ and $AP_{BEV}$ on both accuracy and running speed. Specifically, DLA-34 as the backbone achieved the best accuracy and was faster than almost all previous methods. Using ResNet-18, we achieved the best speed with 47 FPS, and accuracy also outperformed most other approaches. The improvement of run time came from discarding the region proposal process scheme and embedding geometric consistency constraints in the CNN. Better accuracy indicates that combining the respective strengths of neural networks and geometry can achieve better performance than using them alone.

3D Object Detection with Supervised Training on nuScenes Dataset. Work on monocular 3D object detection rarely reports the results of nuScenes, because the detection performance of monocular cameras still has a big gap compared to stereo or LiDAR sensors on the KITTI dataset. Our method still surpassed the published work on MonoDIS, as shown in Table III.

Semi-Supervised Approach. For semi-supervised training, we used the $train1$ set and testing set as the labeled and unlabeled images, respectively, and validated the performance on the $val1$ set. Since ours is the first semi-supervised method for monocular 3D detection, we compared our semi-supervised training to supervised training in various experimental settings. Fig. IV shows that our semi-supervised technique performed more efficiently as the amount of labeled data decreased. Note that with our semi-supervised training, only 500 labeled data points could achieve better accuracy than most methods that adopt supervised training with a complete annotation set.

C. Ablation Study

We conducted a series of experiments to validate the contribution of different strategies in our model. A1) GRM. Two settings were considered to verify GRM’s validity. One was only to use GRM in the inference phase and remove GRM of position supervision in the training phase. The other was to remove GRM in all phases. In this case, we added a parallel depth prediction branch after feature extraction and combined the main center to compute position; A2) Keypoint coordinates of regression vs. heatmap. For prediction keypoints in the heatmap, we added a parallel heatmap branch, whose each channel contained the corresponding semantic points of all objects in the whole picture. A3) 3D confidence prediction; A4) Data augmentation. Table V shows the $AP_{3D}$ and $AP_{BEV}$ for each variant of the proposed strategies. The performance dropped greatly without our geometric reasoning module, indicating the validity of geometric constraints in spatial-related information prediction. Using heatmaps to estimate keypoints also suffered from performance
degradation, especially for the moderate and hard groups. We believe that this is because the heatmap predictions of different keypoints are semantically ambiguous, and cannot estimate the keypoints of the truncated region.

**Effects of Vanishing Point on Orientation Module.** We investigated the effects of the vanishing point on the orientation module by replacing it with a separate regression branch. Table VI shows that removing the VP2O module greatly degraded performance, which demonstrates that the VP2O module enables better information communication by focusing on useful features.

**Extreme testing.** We conducted some extreme testing to further illustrate the potential performance of the proposed method. We input a different number of keypoints to the geometric reasoning module to compute position information. Each keypoint was chosen randomly. As shown in Fig. 4, even with only two keypoints, our geometry reasoning module still performed acceptably. This powerful ability is the main reason for the overall effectiveness of our method.

**VI. CONCLUSIONS AND FUTURE WORK**

In this work, we presented a novel one-stage framework for monocular 3D object detection. Our formulation embeds a differentiable module of perspective geometry in CNN to promote running efficiency and optimize network outputs. Combining the strengths of both CNN and geometric constraints, our method outperformed image-only methods in both efficiency and accuracy by large margins. Another contribution of this work is a semi-supervised training method in settings where labeled data are scarce. Our method only requires unlabeled data and its intrinsic camera parameters, making it practical in scenarios and providing flexibility to further develop the semi-supervised training of 3D object detection.

**REFERENCES**

[1] B. Yang, W. Luo, and R. Urtasun, “PIXOR: Real-time 3D object detection from point clouds,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7652–7660.

[2] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum pointnets for 3D object detection from RGB-D data,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 918–927.

[3] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 770–779.

[4] G. Brazil and X. Liu, “M3D-RPN: Monocular lifting of 2D detection to 6D pose and metric shape,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2069–2078.

[5] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, “Disentangling monocular 3D object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 7074–7082.

[6] Z. Qin, J. Jin, and Y. Lu, “MonoGRNET: A geometric reasoning network for monocular 3D object localization,” in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 8851–8858.

[7] Z. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, “Monocular 3D object detection for autonomous driving,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2069–2078.

[8] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, “Disentangling monocular 3D object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 1991–1999.
[10] P. Li, H. Zhao, P. Liu, and F. Cao, “RTM3D: Real-time monocular 3D detection from object keypoints for autonomous driving,” 2020.

[11] P. Li, H. Zhao, P. Liu, and F. Cao, “RTM3D: Real-Time Monocular 3D detection from object keypoints for autonomous driving,” in *Computer Vis. - ECCV 2020 - 16th Eur. Conf., Glasgow*, Springer, vol. 12348, pp. 644–660, 2020, doi: 10.1007/978-3-030-58580-8_38.

[12] B. Xu and Z. Chen, “Multi-level fusion based 3D object detection from monocular images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2345–2353.

[13] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, “GS3D: An efficient 3D object detection framework for autonomous driving,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 1019–1028.

[14] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” 2019, arXiv:1904.07850.

[15] A. Naiden, V. Paunescu, G. Kim, B. Jeon, and M. Leordeanu, “Shift R-CNN: Deep monocular 3D object detection with closed-form geometric constraints,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1057–1066.

[16] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, “Deep fitting degree scoring network for monocular 3D object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1057–1066.

[17] P. Li, X. Chen, and S. Shen, “Stereo R-CNN based 3D object detection for autonomous driving,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7644–7652.

[18] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” *Adv. Neural Inform. Process. Syst. 32: Annual Conf. Neural Inform. Process. Syst.*, 2019, NeurIPS 2019, Vancouver, BC, Canada, 2019, pp. 5050–5060.

[19] S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” in *ICLR*, 2017.

[20] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, “Semi-supervised learning with ladder networks,” in *Proc. Adv. Neural Inf. Proc. Syst. 28: Annual Conf. Neural Inf. Proc. Syst.*, eds. C. Cortes, N. D. Lawrence, D. M. Lee Sugiyama, and R. Garnett, Eds, Curran Associates, Inc., 2015, pp. 3546–3554.

[21] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Proc. Int. Conf. Learn. Representations, ICLR*, OpenReview.net, 2017.

[22] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6851–6860.

[24] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-lidar from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8445–8453.

[25] F. Chabot, M. Chaouch, 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2040–2049.

[26] M. Zeeshan Zia, M. Stark, and K. Schindler, “Are cars just 3D boxes?: jointly estimating the 3D shape of multiple objects,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3678–3685.

[27] J. K. Murthy, G. S. Krishna, F. Chhaya, and K. M. Krishna, “Reconstructing vehicles from a single image: Shape priors for road scene understanding,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 724–731.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proc. IEEE*, 1998, pp. 2278–2324.

[29] S. Yang and S. Scherer, “CubeSLAM: Monocular 3D object SLAM,” *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.

[30] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation Networks,” 2017, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, 2020, doi: 10.1109/TPAMI.2019.2913372.

[31] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6850–6859. [Online]. Available: http://dblp.uni-trier.de/db/conf/iccv/iccv2019.html MaWLZO19

[32] H. Law and J. Deng, “Cornerenet: Detecting objects as paired keypoints,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.

[33] M. B. Giles, “An extended collection of matrix derivative results for forward and reverse mode automatic differentiation,” 2008.

[34] C. Ionescu, O. Vantzos, and C. Sminchisescu, “Matrix backpropagation for deep networks with structured layers,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2965–2973.

[35] H. Caesar et al., “nuscenes: A multimodal dataset for autonomous driving,” *IEEE/CVF Conf. Compu. Vis. Pattern Recognition CVPR*, 2020, doi: 10.1109/CVPR42600.2020.01164.

[36] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang, “Rethinking Pseudo-LiDAR representation,” in *Comput. Vis. - ECCV 2020 - 16th Eur. Conf.*, A. Vedaldi, B. Isufi, T. Brox, and J.-M. Frahm, Eds., Glasgow, UK, vol. 12358. Springer, vol. 12358, pp. 311–327, 2020, doi: 10.1007/978-3-030-58601-0_19.

[37] D. Beker et al., “Monocular differentiable rendering for self-supervised 3D object detection,” in *Comput. Vis. - ECCV 2020 - 16th Eur. Conf.*, A. Vedaldi, B. Isufi, T. Brox, and J.-M. Frahm, Eds., Springer, vol. 12358, pp. 514–529, 2020, doi: 10.1007/978-3-030-58589-1_31.

[38] Y. Chen, L. Tai, K. Sun, and M. Li, “Monopair: Monocular 3D object detection using pairwise spatial relationships,” *IEEE/CVF Conf. Compu. Vis. Pattern Recognition, CVPR*, 2020, pp. 12090–12099, doi: 10.1109/CVPR42600.2020.01211.