Improving Point Cloud Semantic Segmentation by Learning 3D Object Detection

Ozan Unal[1], Luc Van Gool[1,2], and Dengxin Dai[1]

[1]Computer Vision Lab, ETH Zurich
[2]VISICS, ESAT/PSI, KU Leuven
{ozan.unal, vangool, dai}@vision.ee.ethz.ch

Abstract

Point cloud semantic segmentation plays an essential role in autonomous driving, providing vital information about drivable surfaces and nearby objects that can aid higher level tasks such as path planning and collision avoidance. While current 3D semantic segmentation networks focus on convolutional architectures that perform great for well represented classes, they show a significant drop in performance for underrepresented classes that share similar geometric features. We propose a novel Detection Aware 3D Semantic Segmentation (DASS) framework that explicitly leverages localization features from an auxiliary 3D object detection task. By utilizing multitask training, the shared feature representation of the network is guided to be aware of per class detection features that aid tackling the differentiation of geometrically similar classes. We additionally provide a pipeline that uses DASS to generate high recall proposals for existing 2-stage detectors and demonstrate that the added supervisory signal can be used to improve 3D orientation estimation capabilities. Extensive experiments on both the SemanticKITTI and KITTI object datasets show that DASS can improve 3D semantic segmentation results of geometrically similar classes up to 37.8% *IoU in image FOV while maintaining high precision bird's-eye view (BEV) detection results.*

1. Introduction

The goal of a truly autonomous vehicle is to provide a safer option of travel by removing the human element from the equation. However, this is not trivially accomplished as the vehicle needs to go beyond following a list of rules of the road and complete high level tasks such as path planning and collision avoidance in real time, which not only benefit from knowing the semantics of its immediate surrounding scene including drivable spaces but also the locations of nearby objects. It is therefore crucial for an established 3D semantic segmentation network to correctly identify and segment foreground object classes such as vehicles with high accuracy.

Point cloud semantic segmentation still remains to be a challenging and computationally expensive task. Investigating the current literature we draw the following observations: (1) While 3D semantic segmentation networks perform well for highly represented classes, they show a rapid decrease in performance for underrepresented classes that share similar geometric features. A common example for such a pairing is the car-truck categories, where due to the similarity in their geometric properties, truck segmentation often underperforms because of extensive false negative rates. (2) 3D object detection frameworks perform well for generating high precision 3D bounding boxes while also being capable of differentiation between the common foreground objects. For example in 3D vehicle detection, often the car and truck classes are treated as separate categories in commonly used datasets [10, 14, 3, 4], thus networks must extract class specific features to correctly classify the objects.

We therefore argue that 3D object detection as an auxiliary task can help improve 3D semantic segmentation results for foreground classes that are underrepresented. Here we will demonstrate, utilizing a car detection auxiliary task can have great benefits when segmenting classes such as trucks or other vehicles. However, in order to utilize both tasks in a unified system in terms of joint supervised training, a dataset is required that contains both supervisory signals. While almost all datasets for 3D object detection lack 3D semantic labels [10, 14, 4, 7], those that contain both annotations lack a preestablished benchmark for performance comparison [11, 28, 3].

To this end we propose a novel network that we call Detection Aware 3D Semantic Segmentation (DASS), a framework for 3D semantic segmentation that utilizes 3D object detection as an auxiliary task to improve its segmentation performance. Our proposed framework directly consumes

irregular point clouds via a PointNet++ [27] feature extractor to predict semantic labels for points that fall on the front view camera field-of-view (image FOV) while also generating high recall object proposals. DASS is trained using supervisory signals from two partial datasets [10, 2] that only contain a set of annotations for a single task and shows improvements of incredible margins for categories geometrically related to the detection class.

Our key contributions can be summaries as follows: (1) We introduce DASS, a framework for joint point cloud semantic segmentation and 3D object proposal generation from partial datasets. (2) We show that the 3D object detection auxiliary task can improve the generalizability of the shared feature space, enabling vast improvements in 3D semantic segmentation with results up by 37.8% intersection-over-union (IoU) for categories that share geometric features with the detected class. (3) We introduce no additional memory or computational cost compared to a baseline PointNet++ 3D semantic segmentation network [27], as the auxiliary head can be detached during inference. (4) We demonstrate that our proposed network can be used to generate high recall proposals for existing 2-stage 3D object detectors. Overcoming the capacity limitations of multitask training through a novel semantic feature fusion (SFF) connection, our proposed framework shows comparable birds-eye-view (BEV) detection and improved 3D orientation results when used with the second stage of PointRCNN [30]. Furthermore, the resulting network maintains real time inference at a 11Hz rate with only an added 0.15% memory cost, while simultaneously generating accurate 19-class 3D semantic masks.

2. Related Work

In this section, we consider the current approaches for 3D semantic segmentation and 3D object detection. Furthermore we briefly investigate existing multitask learning methods.

3D Semantic Segmentation: Point cloud semantic segmentation remains to be a challenging and computationally expensive task in literature. Current benchmarks are dominated mainly by convolutional architectures that project the point cloud onto various representations including spherical representations [33, 34], range images [21], BEV [6] and learned 2D representations [1].

DASS does not utilize a convolutional architecture but is built on a PointNet++ backbone [27]. PointNet [26, 27] proposed a framework that directly consumes point clouds as opposed to parsing to a highly sparse voxel space. [19, 31, 18] extend regular grid CNNs to irregular point configurations by utilizing novel point convolutions to extract local and global features without the loss of information inherent in the process of voxelization. While PointNet based methods tend to underperform [2], this enables DASS to establish encoder level interactions with existing PointNet based 3D object detection frameworks, enabling it to outperform existing convolutional benchmarks.

3D Object Detection: State-of-the-art 3D object detectors utilize various strategies to deal with the irregular format of point clouds in order to regress the 7 degrees of freedom of a 3D bounding box. Some methods utilize a single stage design, where the final bounding boxes are directly regressed [36, 12, 37, 24], while others prefer a two stage design, where the first stage generates coarse predictions using a region-proposal-network (RPN) and the second stage refines the proposals for the final predictions [30, 25, 5, 29].

DASS utilizes an auxiliary task of 3D object proposal generation. Trained with 3D semantic segmentation, the proposed network provides high recall proposals and thus can be used as a replacement RPN for the currently existing 2-stage architectures like PointRCNN [30] to simultaneously generate 3D detection results with semantic labels.

Compared to the first stage of PointRCNN [30], DASS completes semantic segmentation of 19 classes while overcoming performance drops in detection that originate from multitask learning. Compared to PointRCNN [30] that predicts binary masks for foreground points which provide explicit information to aid localization, DASS also exploits the unexplored semantic information within the surrounding scene to add additional constraints on the distribution of the bounding boxes, which help further improve orientation estimation.

Multitask Learning: Multitask learning (MTL) aims to leverage the supervisory signals of multiple tasks to improve the generalization capabilities of a model. It achieves this by utilizing encoder-level interactions to generate a shared representation [22, 9, 17], by using decoder-level interactions to improve single task results from multi-modal distillation [35, 38], or a set combination of both.

[32] shows that in an MTL setting, performance strongly varies depending on a wide range of parameters (e.g task type, label source) and thus architecture and optimization strategies must be selected on a per case basis. In general it is observed that encoder level interactions perform well for multiple classification problems while decoder level interactions have an advantage in dense prediction tasks.

While MTL has been tackled before in various task types [13, 16, 8, 20], DASS explores the yet unexplored setting of MTL with point cloud semantic segmentation and 3D object detection from two sources of point cloud data that are partially annotated. Through encoder-level interactions, DASS maintains the required inference time and memory for point cloud semantic segmentation. Exploiting decoder level interactions via an SFF layer allows DASS to overcome the performance limitations of encoder-focused MTL and maintain high precision regression for 3D detection.

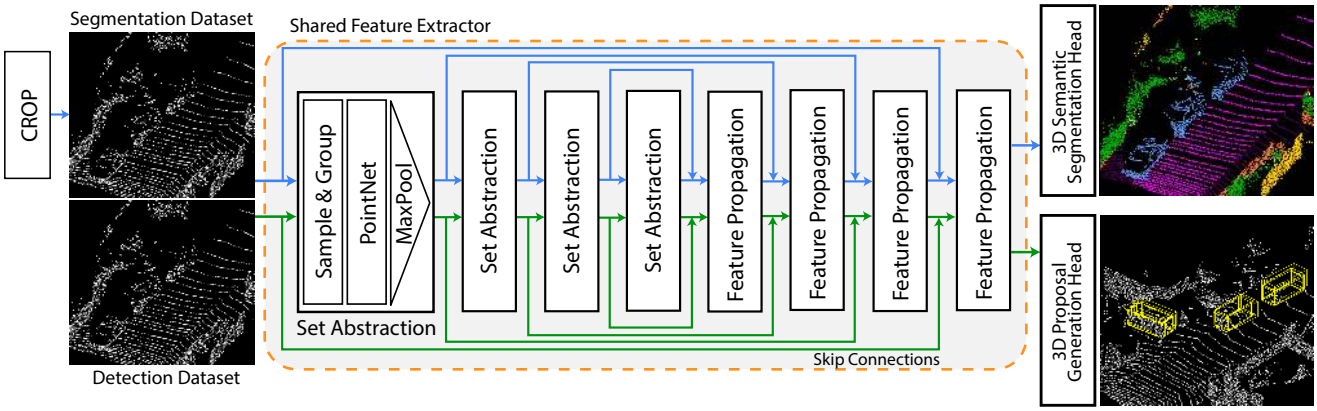

Figure 1. Network overview. The network is trained on two partial datasets: (1) with pointwise semantic labels [2] and (2) with 3D object annotations [10]. Since the detection datasets provides annotations for just the image FOV, the segmentation dataset is cropped to avoid introducing further domain shifts. A PointNet++ feature extractor is trained on both supervisory signals followed by the individual task heads for the 3D semantic segmentation task and the auxiliary 3D proposal generation task. Best viewed in color.

3. DASS: Detection Aware 3D Semantic Segmentation

In this section we present DASS, a 3D semantic segmentation network with auxiliary 3D object proposal generation, and its training procedure from partial datasets. The overall network pipeline is shown on Fig. 1.

3.1. Utilizing a Shared Feature Space from Partial Datasets

We start by defining the tasks of 3D semantic segmentation and 3D object proposal generation.

3D Semantic Segmentation: Point cloud semantic segmentation is the function ϕ_{seg} that assigns a set of semantic labels $L \in \mathbb{Z}^n$, for each point in a given point cloud $P \in \mathbb{R}^{(n \times d)}$ with n points of d dimensions, i.e. $\phi_{seg} : P \mapsto L$.

3D Object Proposal Generation: A bounding box for object o is represented by its 7 degrees of freedom $(x_o, y_o, z_o, h_o, w_o, l_o, r_o) \in \mathbb{R}^7$, where (x_o, y_o, z_o) define the box center, (h_o, w_o, l_o) define the box height, width, and length respectively, and r_o defines the object rotation around the y-axis in the camera coordinate system. In most autonomous driving cases, the pitch and roll are assumed to be zero. In essence, 3D object proposal generation is the function ϕ_{det} that generates k number of bounding boxes within a scene, i.e. returns the set of object bounding boxes $B \in \mathbb{R}^{(k \times 7)}$ for a given point cloud P, and proposal count k with $\phi_{det} : P \mapsto B$.

MTL from Partial Datasets: The goal of multitask training with 3D semantic segmentation and 3D object proposal generation is to find a function ϕ_{MT} that returns both per point semantic labels and 3D bounding boxes, i.e. $\phi_{MT} : P \mapsto (\phi_{seg}(P), \phi_{det}(P))$.

To learn the mapping of a point cloud P onto a target tuple (B, L) via supervised learning, a dataset is required that contains both ground truth semantic labels L and object bounding boxes B. However, amongst the datasets with annotated ground truth bounding boxes, most do not contain per point semantic labels [10, 7, 14, 4], while those that do lack an established benchmark for performance comparison [11, 28, 3]. Thus we are bound to two datasets, each with partial supervisory signals [10, 2].

DASS utilizes a shared feature extractor to project the point clouds onto a shared representation F. The functions ϕ_{seg} and ϕ_{det} can then be individually trained with the input and target tuples $(F(P_{seg}), y_{seg})$ and $(F(P_{det}), y_{det})$ respectively, with P the input point cloud and y the target labels, yielding an overall mapping of $\phi_{MT} : P \mapsto ((\phi_{seg} \circ F)(P), (\phi_{det} \circ F)(P))$.

In other words, DASS exploits the commonality of 3D semantic segmentation and 3D object detection by utilizing their supervisory signals in parallel. As seen in Fig. 1, a PointNet++ [27] feature extractor is forced to share weights between the primary and auxiliary task, with optimizer steps during training taken from a joint multitask loss. The benefits of such encoder-level interactions when multitask training from partial datasets are three fold: (1) The effective size of the dataset is increased; (2) The training signals of each task act as an inductive bias for the other, improving the generalization capabilities of the model. The feature vector for each point is forced to contain valuable information about its semantic context as well as the detected object class, enhancing segmentation capabilities by allowing better differentiation between geometrically similar classes; (3) By having the bulk of the network parameters reside in the shared feature extractor, the computational overhead and memory requirement of incorporating an additional task is drastically reduced during training. It is also important to note that during inference, the proposal gener-

ation head can be detached thus adding no additional cost. DASS can therefore maintain high inference rates while producing accurate semantic masks and 3D object proposals.

3.2. Joint Proposal Generation and Point Cloud Semantic Segmentation

As seen in Fig. 1, following the shared encoder-decoder, a proposal generation head ϕ_{det} and a semantic segmentation head ϕ_{seg} are appended to generate 3D semantic labels with coarse detection results. Every batch consists of two mini batches, each with the data from a single partial dataset. After iterative forward passes, a single backward pass is done from the accumulated gradients. In other words, the shared feature extractor is trained from both partial datasets, while the individual heads are trained from a single partial dataset. The shared feature extractor is trained using a multitask loss function given by the weighted sum of the individual task losses of each head, i.e. the first stage total loss is computed as:

$$\begin{aligned}\mathcal{L} =& w_{seg}\,\mathcal{L}_{seg}((\phi_{seg}\circ F)(P_{seg}), y_{seg}) \\ &+ w_{det}\,\mathcal{L}_{det}((\phi_{det}\circ F)(P_{det}), y_{det})\end{aligned} \tag{1}$$

with w_{seg}, $\mathcal{L}_{seg}$ denoting the 3D semantic segmentation weight and loss and w_{det}, $\mathcal{L}_{det}$ denoting the 3D object detection weight and loss.

Segmentation Loss: The 3D semantic segmentation head seen in Fig. 1 generates pointwise semantic labels. The head is trained using the cross-entropy loss with a weight vector $w_{classes}$ to deal with the class imbalance. The segmentation loss is given by

$$\mathcal{L}_{seg} = \mathcal{L}_{CE}(L, \hat{L}; w_{classes}) \tag{2}$$

with $\mathcal{L}_{CE}$ denoting the cross entropy loss, L and $\hat{L}$ denoting the sets of estimated semantic labels and their corresponding ground truths respectively.

Detection Loss: For the auxiliary 3D proposal generation task, we use a per-point bin based loss function following PointRCNN [30]. The surrounding area of each point is discretized into a set number of bins. This allows us to restate the problem of the bounding box center localization on the transverse plane (x, z) as a classification problem which are shown to be better fitted for encoder-focused architectures [32]. To achieve finer details, we allow a residual to be regressed for each bin. For a bin size of δ in a surrounding area of radius S, we define $k = \delta\,(k_{bin} + 1/2) + k_{res} - S$ for $k \in \{x, z\}$. Here $k_{bin} \in [\lfloor -S/\delta \rfloor, ..., -1, 0, 1, ..., \lfloor S/\delta \rfloor]$ defines the bin in which the target is located, and $k_{res} \in (-\delta/2, \delta/2)$ defines the residual within that bin. Similarly, the rotation estimation is also restated as a classification problem where the transverse plane is divided into a set number of angles α

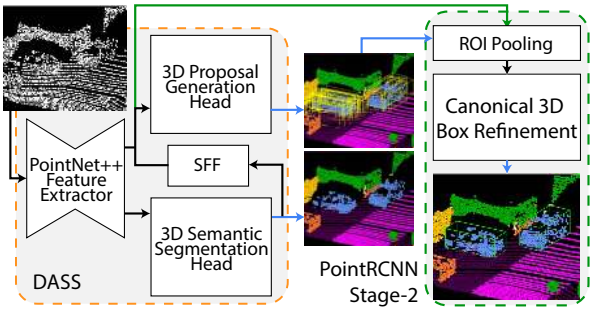

Figure 2. Network extension overview. DASS is used as the RPN of PointRCNN [30]. To further improve proposal generation results semantic feature fusion (SFF) is applied before the proposal generation.

where again we define $k = \alpha\,k_{bin} + k_{res}$ for $k \in \{r\}$ with $k_{bin} \in [0, 1, ..., \lfloor 2\pi/\alpha \rfloor]$ and $k_{res} \in (-\alpha/2, \alpha/2)$.

As all objects within a driving scene are ground bound and have similar sizes per class, the elevations y and size residuals (h, w, l) of bounding boxes follow very narrow distributions, allowing these values to be regressed directly. The resulting loss function is given by:

$$\begin{aligned}\mathcal{L}_{det} =& \sum_{k\in\{x,z,r\}} \Big(\mathcal{L}_{CE}(k_{bin}, \hat{k}_{bin}) + \mathcal{L}_{sL1}(k_{res}, \hat{k}_{res})\Big) \\ &+ \sum_{j\in\{y,h,w,l\}} \mathcal{L}_{sL1}(j, \hat{j})\end{aligned} \tag{3}$$

where hat denotes the ground truth and $\mathcal{L}_{sL1}$ denotes the smooth L1 loss.

All points that lie outside of ground truth bounding boxes do not contribute to the detection loss.

3.3. DASS in a 2-stage 3D Object Detection Pipeline

We make the following observation: Each semantic mask can act as a constraint on a bounding boxes distribution. For example cars are likely to leave substantial space between themselves and surrounding buildings; cars, cyclists and pedestrians are more likely to be rotated along the direction of the road or sidewalks respectively and away from paths of direct collision [23]. We therefore argue that, not only does the 3D semantic segmentation task benefit from the auxiliary 3D proposal generation, but the opposite also holds especially for BEV detection and 3D orientation. The 3D proposal generator can therefore benefit highly from knowing the semantic masks of specific background classes, thus DASS can be used to generate high recall proposals for existing 2-stage detectors.

In Fig. 2, we illustrate a pipeline for PointRCNN [30] that utilizes DASS as the primary proposal generator. The proposals are initially expanded to form regions of interests

which are then pooled with the shared features to generate the input of the second stage. The second stage applies canonical box refinement on the generated proposals to predict the final 3D detection results [30].

With the proposed extension, DASS becomes the first pipeline to generate 19-class point cloud semantic segmentation results with 3D object bounding boxes in real time at an operating frequency of 11Hz (Nvidia Titan Xp 12G GPU 2.2GHz) with a minimal added memory cost of just 0.15%. This can be highly beneficial when dealing with real world problems of navigation in complex scenes that involves on-the-go path planning and collision avoidance.

Semantic Feature Fusion: In MTL, the problem of invariance vs. sensitivity is always apparent [32]. The network may never converge to a state where it extracts a vital feature for one task because it contradicts with the objective of the other. With DASS, we tackle a currently unexplored area of MTL and consider two tasks that directly consume point clouds with 3D object detection and 3D semantic segmentation from partial datasets. While 3D semantic segmentation performs well under the limited capacity of the feature extractor, we observe a severe reduction in performance for high precision localization.

To overcome this issue, we introduce semantic feature fusion (SFF) as a decoder level interaction as seen in Fig. 1. During training, with gradient accumulation set to zero, we do a forward pass of the detection mini-batch through the shared feature extractor and the segmentation head to infer the per point semantic label likelihoods. SFF learns to summarize this high dimensional likelihood vector through 1D convolutional operations and provides the 3D proposal generation head a compact representation of the scene semantics. These are concatenated back with the shared features to be input to the 3D proposal generation head. By directly utilizing such compacted information, we minimize adding redundant information and further complexity to the system, while still allowing the detection head to directly extract inter class dependencies. With SFF, DASS achieves higher 3D detection recall across all thresholds, which is the goal of the first stage detector as precise localization is carried out in the second stage.

4. Experiments

In this section we look into the architecture specifications and training scheme of DASS from partial datasets. We report the results of our network and provide ablation studies on the training procedure and individual components.

4.1. Network Architecture

For the PointNet++ [27] encoder-decoder, 4 set-abstraction layers with multi-scale grouping are used with group sizes of 4096, 1024, 256, 64 points of increasing radii. Every group & sampling operations of the set abstraction layers are followed by a block of 3 linear layers for each of the two scales. The set abstraction layers feed into 4 feature propagation layers with skip connections to obtain per point feature vectors that are rich in both semantic and class-specific information. While using 2-scale grouping allows us to introduce scale invariance into our network, the hierarchical structure of the PointNet++ feature extractor captures better local properties which benefit both tasks.

Both the first stage 3D semantic segmentation and the 3D object proposal generation heads consist of a single 1D convolutional layer of size 128. Batch norm and ReLU activation is applied after every layer. The learning rate is set to 0.002. Adam optimizer is used with a one-cycle learning rate scheme. The weight decay is set to 0.001 with momentum of 0.9.

4.2. Training Scheme

Dataset: Due to a lack of unified dataset that contains both ground truths, two partial datasets are utilized for training a weight-sharing PointNet++ encoder-decoder structure [27], such that the resulting pointwise feature vectors include both object-class and semantic information. In specific, the 3D semantic segmentation pipeline is trained on the SemanticKITTI *train* set [2] while the 3D object proposal network is trained on the KITTI *train* set [10] with the car category.

Point Cloud Preprocessing and Augmentation: As KITTI [10] does not provide ground truth 3D bounding box annotations for the full 360° view, we crop the 3D point clouds such that all points lie within the image FOV. Applying the same transformations to both partial datasets is crucial as this avoids added domain shifts within the training process and thus prevents overfitting to domain specific features. This in turn means that our semantic segmentation network operates exclusively on image FOV despite the provided 360° labels from SemanticKITTI [2]. Each cropped region is then randomly subsampled to contain a 16384 points. If a scene contains less points after the cropping it is zero-padded until the fixed value is reached. The reflectance intensity values of all points within the point cloud are normalized by subtracting 0.5.

It is important to note that given a dataset that contains 360° annotations for both tasks, our method can easily be reapplied to provide 360° semantic labels and will again show the same benefits of adding no memory or computational cost during inference.

Two data augmentation schemes are implemented. (1) For both datasets, each scene is randomly rotated by an angle sampled from $[-10^\circ, 10^\circ]$, scaled by a scale factor sampled from $[0.95, 1.05]$ and flipped randomly with 0.5 probability. (2) For the object detection dataset only, following [36, 30] ground truth bounding box augmentation

Method	Parameters [Million]	Inference Time [s]	3D Semantic Segmentation	3D Object Detection
SqueezeSeg[33]	1	0.015	360°	-
SqueezeSegV2[34]	1	0.02	360°	-
DarkNet21[21]	25	0.055	360°	-
DarkNet52[21]	50	0.01	360°	-
PointRCNN	3.90	0.09	-	✓
Semantic Baseline	3.04	0.015	Image FOV	-
DASS	3.04	0.015	Image FOV	Proposal
DASS+RCNN	3.91	0.09	Image FOV	✓

Table 1. Approach summaries.

is applied, where ground truth boxes and their respective points are taken from various scenes to be implanted in another. The new box and its points are placed within the point cloud at the same location assuming that there is no existing overlapping box. The box with its points are then translated such that it lies on the ground plane. Furthermore, points above and below the box are removed from the point cloud.
Training: A mini-batch size of 8 is used for both tasks which yields cleaner gradients compared to an unbalanced grouping strategy. Due to the size differences of the two datasets, we define an epoch for the first stage as a single iteration over the SemanticKITTI [2] dataset with multiple shuffled cycles over the KITTI object dataset [10]. The two tasks are weighed by (1.5, 1) for 3D semantic segmentation and 3D object detection respectively following the inverse of their converged individual losses. The network is trained for 75 epochs.

For the training of the stage-1 regression head, only the points that lie inside a bounding box are considered in the loss function.

Following [15], all car objects that lie outside of the range x, y, z [[-40, 40], [-1, 3], [0, 70.4]] meters are filtered out. The mean car bounding box is set to have a height, width and length of (1.5, 1.6, 3.9) meters and the size of each object (h, w, l) is regressed as the difference from the mean anchor. The search scope is set to $3.0m$ resulting in a bin size of $\delta = 0.5m$ with 6 equal length bins, and the rotation range is set to 2π with 12 discrete heads. Vans are also considered within the car category.

To deal with the class imbalance apparent in the SemanticKITTI [2] dataset, weighted cross entropy is used with a 19 class weight vector $w_{classes}$ given by the inverse of a class' frequency within the entire set of point clouds.

4.3. Results

We report 3D semantic segmentation results from the SemanticKITTI [2] val set on image FOV and the recall for 3D object detection at varying thresholds from the KITTI [10] val set. We demonstrate on the KITTI [10] $test$ set that our network can be used in conjunction with existing 2-stage detectors to generate comparable BEV detection results, improved 3D orientation estimation, to maintain minimal memory cost, to operate in real time at 11Hz frequency, all while simultaneously generating 3D semantic masks. Further statistics on evaluated approaches can be found on Tab. 1. Example results can be seen in Fig. 4.

4.3.1 3D Semantic Segmentation Results

As DASS operates on image FOV, the performance cannot be evaluated on the SemanticKITTI [2] *test* set which requires full $360°$ annotations. Thus we opt to use the SemanticKITTI [2] *val* set on image FOV, where all results are evaluated using the official metric of per class mean intersection-over-union (mIoU).

The results can be seen on Tab. 2 where DASS is compared against the benchmark networks provided by SemanticKITTI [2]. The per point label predictions of various SemanticKITTI benchmark networks including SqueezeSeg [33], SqueezeSegV2 [34] and DarkNet [21, 2] have been released. Here, we reevaluate these published results on image FOV. Our proposed network shows an overall better classification performance than the benchmarks with a mIoU improvement of 0.8%, with emphasis drawn on classes that share geometric features with the car category. As observed, in the car, truck, and other vehicle categories, DASS outperforms all benchmark networks with incredible margins of $4.5\%, 32.7\%, 16.6\%$ respectively. Due to the added supervisory signals of the 3D object detection task, the task specific head can utilize the shared features to better distinguish classes of similar characteristics.

In Fig. 3 we demonstrate this differentiation capability of DASS amongst geometrically similar vehicle classes and compare it to existing benchmark segmentation networks [21, 34]. We additionally overlay the predicted bounding boxes of DASS+RCNN to illustrate the detection awareness of DASS. Here it is shown that DASS can better separate the truck (purple) and other vehicle (blue) classes from the overrepresented car (light blue) class thanks to its auxiliary detection task.

4.3.2 3D Object Detection Results

Here we provide results for the commonly reported *car* category. Following PointRCNN [30], we initially generate 9000 proposals which are then reduced to 100 using distance based sampling and non-maximum suppression (NMS) with a threshold of 0.8.

The first stage recall values are seen in Tab. 3. Here we observe both the benefits and detriments of multitask training. The increased generalization capabilities of the shared feature extractor allows the object proposal to generate higher recall proposals for lower thresholds of $0.1, 0.3$ and 0.5 with recall values seeing an increase of $0.52\%, 0.74\%$, and 0.75% respectively. However, the localization performance suffers at higher thresholds as the two tasks compete

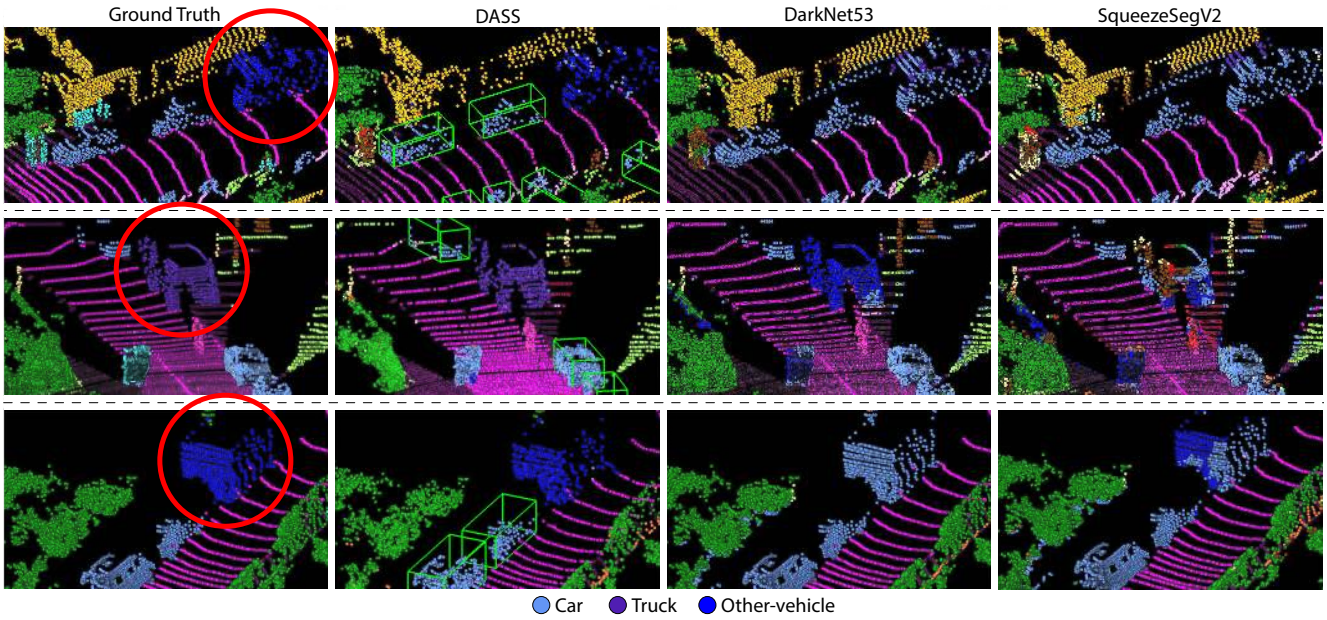

Figure 3. Point cloud semantic segmentation results on the SemanticKITTI [2] *val* set. DASS is compared against the ground truth labels [2], DarkNet53 [21] and SqueezeSegV2 [34]. To illustrate the detection awareness of DASS, we overlay the predicted bounding boxes from the DASS+RCNN network. Here we draw an emphasis on the differentiation capabilities of DASS for the vehicle classes, with corresponding colors and class tags given below the figure. Best viewed in color.

Method	mIoU	car	bicycle	motorcycle	truck	other vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other ground	building	fence	vegetation	trunk	terrain	pole	traffic sign
SqueezeSeg [33]	32.1	75.9	12.8	11.5	3.3	4.0	22.0	34.4	0.1	91.7	16.2	62.7	0.4	57.2	18.8	67.1	27.7	65.6	23.3	14.5
SqueezeSeg-CRF [33]	33.4	75.1	13.6	15.4	3.9	10.6	27.7	39.9	**0.2**	90.8	16.4	62.2	**0.6**	61.8	22.7	66.5	26.7	65.9	13.7	21.1
SqueezeSegV2 [34]	41.3	84.1	15.1	24.8	25.1	27.9	23.4	44.7	0.0	94.4	36.9	74.3	0.1	71.3	37.7	74.5	36.2	69.8	21.8	22.0
SqueezeSegV2-CRF [34]	42.6	85.5	18.5	39.3	23.2	31.4	33.7	54.6	0.0	94.3	32.3	73.4	0.2	69.3	39.4	71.6	37.5	69.2	14.8	20.9
DarkNet21 [21]	49.0	86.5	27.6	39.6	35.5	23.4	43.2	50.1	0.0	**95.9**	**40.9**	79.8	0.0	77.3	50.7	81.4	53.7	72.0	42.2	31.9
Darknet52-512 [21]	34.9	79.8	14.9	17.0	3.9	14.6	11.2	26.9	0.0	93.3	21.2	70.9	0.1	59.2	30.9	70.6	36.6	65.7	23.8	23.4
DarkNet53-1024 [21]	39.2	83.9	17.5	3.5	24.1	8.3	10.9	43.3	0.0	94.2	15.3	74.1	0.0	70.6	47.7	78.0	38.5	70.0	34.8	30.0
DarkNet53 [21]	51.0	86.9	26.6	**47.5**	34.0	27.2	**51.6**	62.7	0.0	**95.9**	39.7	**80.0**	0.0	77.8	**51.1**	81.3	53.6	71.9	**46.7**	33.8
Semantic Baseline	48.0	89.8	**33.7**	29.9	28.9	28.9	35.5	62.8	0.0	94.6	32.1	75.7	0.4	**82.1**	42.6	83.3	53.0	**73.6**	38.2	26.3
DASS	**51.8**	**91.4**	25.8	31.0	**66.7**	**43.8**	47.7	**70.8**	0.0	92.8	31.7	71.0	0.0	**82.1**	39.1	**83.5**	**56.6**	69.6	45.5	**35.1**

Table 2. 3D semantic segmentation results. All networks are evaluated on image FOV using the SemanticKITTI [2] validation set. Semantic Baseline denotes DASS trained without the auxiliary task.

for capacity. A feature that is highly beneficial for localization may be a nuisance for the 3D semantic segmentation task. Thus at higher thresholds of $0.7, 0.9$, DASS underperforms by $1.9\%, 0.45\%$ compared to the stage-1 of PointRCNN.

We use DASS as the stage-1 object proposal generator and append the stage-2 refinement network of PointRCNN to generate BEV and 3D orientation results, as we expect DASS to provide additional auxiliary information that can benefit such tasks (see Sec. 3.3). We call this network DASS+RCNN. In Tab. 4 the KITTI [10] *test* set results are given, however it should be noted that while DASS+RCNN is trained on the official train/val split (50/50), the reported PointRCNN results are obtained with a (80/20) split. Nonetheless our 3D semantic segmentation network still performs at a comparable level with PointRCNN [30]. In the BEV category DASS+RCNN falls just shy on every difficulty by $0.39\%, 1.54\%, 1.75\%$ but achieves better results in 3D orientation by $0.3\%, 0.48\%, 0.34\%$ while simultaneously generating 3D semantic masks ranging 19 classes and only causing an additional 0.15% memory requirement.

4.4. Ablation Studies

In this section we provide extensive ablation studies to analyze both the effectiveness of the added components. All components are evaluated on the KITTI [10] and SemanticKITTI *val* splits [2] for 3D object detection and 3D semantic segmentation respectively.

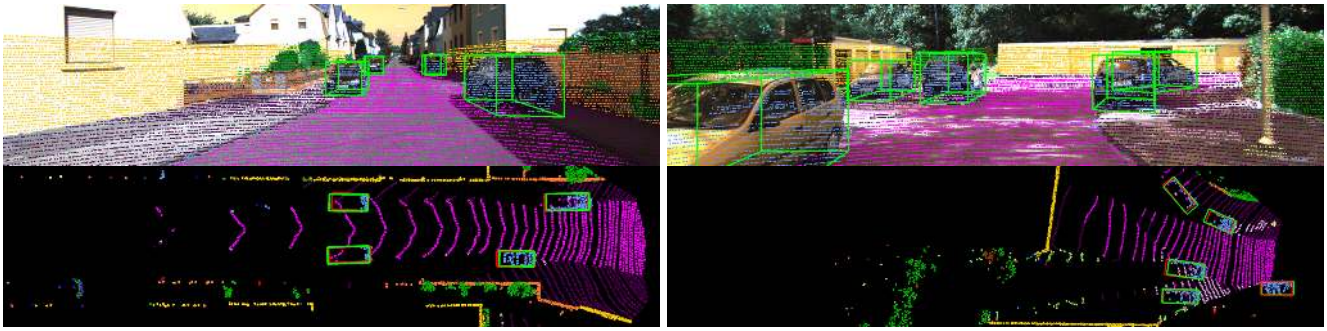

Figure 4. Example results from the KITTI [10] *val* set. Shown are (top) pointwise semantic labels and predicted bounding boxes in green overlayed onto the camera 2 image; (bottom) BEV results with ground truth boxes in red and predicted boxes in green. The multitask results are generated using DASS as the RPN for PointRCNN [30]. Best viewed in color.

	Recall at IoU [%]				
Method	0.1	0.3	0.5	0.7	0.9
PointRCNN[30]	96.84	95.71	93.49	**73.37**	**1.10**
DASS	96.90	96.19	93.80	68.95	0.40
DASS+SFF	**97.36**	**96.45**	**94.24**	71.47	0.65

Table 3. Recall results for 3D detection at varying IoU thresholds for the car class. All networks are evaluated on the KITTI [10] *val* set.

	BEV [%]			Orientation [%]		
Method	Easy	Moderate	Hard	Easy	Moderate	Hard
PointRCNN[30]	**92.13**	**87.39**	**82.72**	95.90	91.77	86.92
DASS+RCNN	91.74	85.85	80.97	**96.20**	**92.25**	**87.26**

Table 4. Evaluation of BEV and 3D orientation on the KITTI [10] *test* set for the car class. As seen DASS can be used as a RPN for PointRCNN [30] (DASS+RCNN) to achieve comparable BEV and 3D orientation results while generating 3D semantic masks ranging 19 classes.

Auxiliary 3D Object Proposal Generation: In Tab. 2 we provide a comparison of our proposed network trained without the aid of the auxiliary task, which we call *Semantic Baseline*. Similar to the comparison drawn in Sec. 4.3.1, we observe an overall increase in mIoU by 3.8%, with the performance boost mainly coming from the car, truck and other vehicle classes with mIoU increases of $1.6\%, 37.8\%, 14.9\%$ respectively.

However it should also be noted that some classes show inferior results when multitask training. As stated before, the shared feature space may never converge to a state where a crucial information for 3D semantic segmentation exists, if that feature contradicts with the objectives of the auxiliary detection task. An example can be observed by the drop in performance for the road and parking classes by up to 1.8% and 0.4% respectively. While drivable surfaces provide information regarding the boundaries of the vehicles (e.g. the elevation of the bounding box as it must lie on a drivable surface), the distinction between drivable surfaces (e.g. road vs. parking) does not provide any further information in that regard which results in misidentified region boundaries.

Semantic Feature Fusion: We evaluate the effectiveness of SFF by drawing comparisons to the baseline DASS. With its increased generalization capabilities, DASS matches or exceeds PointRCNN [30] RPN's 3D recall at lower IoU thresholds. However, as the two tasks compete for capacity, the network fails to extract the much needed task-specific features that aid high precision localization.

As seen in Tab. 3, providing the 3D object proposal head with summarized semantic context via an SFF layer improves its recall for all thresholds. In specific, the detection head mostly benefits from the semantically rich feature at higher IoU thresholds of 0.7 and 0.9 with recall increases of 2.52% and 0.25% respectively. This enables the network to achieve better localization by extracting further class-specific and inter-class dependencies.

5. Conclusion

In this work we proposed a Detection Aware 3D Semantic Segmentation (DASS) network to tackle limitations of current architectures. Our proposed network utilizes an auxiliary 3D object detection task to guide the shared feature representation into extracting localization features that allow better differentiation between geometrically similar foreground classes. Experiments on the SemanticKITTI dataset show that this significantly improves 3D semantic segmentation in image FOV without any additional memory requirement or computational overhead, as the auxiliary task head can be detached during inference. We further investigate the yet unexplored problem of multitask learning of 3D semantic segmentation and 3D object detection from point clouds. We showcase a 2-stage 3D object detection pipeline that utilizes DASS as a RPN with preexisting architectures and overcome the capacity limitations of multitask learning through semantic feature fusion. Experiments on the KITTI dataset show that DASS can improve 3D orientation estimation while preserving BEV detection results, operating in real time, costing negligible memory, and producing highly accurate 3D semantic masks.

Acknowledgements: This work was funded by Toyota Motor Europe via the research project TRACE Zurich.

References

[1] Iñigo Alonso, Luis Riazuelo, Luis Montesano, and Ana C Murillo. 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. *arXiv preprint arXiv:2002.10893*, 2020.

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9297–9307, 2019.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[4] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[5] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9775–9784, 2019.

[6] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast semantic segmentation of lidar point clouds for autonomous driving. *arXiv preprint arXiv:2003.03653*, 2020.

[7] Waymo Open Dataset. An autonomous driving dataset, 2019. https://waymo.com/open/.

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[9] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3205–3214, 2019.

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

[11] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2d2: Audi autonomous driving dataset, 2020.

[12] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11873–11882, 2020.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[14] R Kesten, M Usman, J Houston, T Pandya, K Nadhamuni, A Ferreira, M Yuan, B Low, A Jain, P Ondruska, et al. Lyft level 5 av dataset 2019. *https://level5. lyft. com/dataset*, 2019.

[15] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[16] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019.

[17] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

[18] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5239–5248, 2019.

[19] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019.

[20] Gregory P Meyer, Jake Charland, Darshan Hegde, Ankit Laddha, and Carlos Vallespi-Gonzalez. Sensor fusion for joint 3d object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[21] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[22] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[23] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.

[24] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019.

[25] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018.

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification

and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[28] Scale. Pandaset, 2020. https://scale.com/open-datasets/pandaset.

[29] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[30] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[31] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019.

[32] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey, 2020.

[33] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018.

[34] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[35] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018.

[36] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[37] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020.

[38] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2019.