QuadroNet: Multi-Task Learning for Real-Time Semantic Depth Aware Instance Segmentation

Kratarth Goel
Zoox Inc.
kratarth@zoox.org

Praveen Srinivasan
Zoox Inc.
praveen@zoox.org

Sarah Tariq
Zoox Inc.
sarah@zoox.org

James Philbin
Zoox Inc.
james@zoox.org

Abstract

Vision for autonomous driving is a uniquely challenging problem: the number of tasks required for full scene understanding is large and diverse; the quality requirements on each task are stringent due to the safety-critical nature of the application; and the latency budget is limited, requiring real-time solutions. In this work we address these challenges with QuadroNet, a one-shot network that jointly produces four outputs: 2D detections, instance segmentation, semantic segmentation, and monocular depth estimates in real-time (>60fps) on consumer-grade GPU hardware. On a challenging real-world autonomous driving dataset, we demonstrate an increase of +2.4% mAP for detection, +3.15% mIoU for semantic segmentation, +5.05% mAP@0.5 for instance segmentation and +1.36% in $\delta <$ 1.25 for depth prediction over a baseline approach. We also compare our work against other multi-task learning approaches on Cityscapes and demonstrate state-of-the-art results.

1. Introduction

Safety-critical, real-world vision applications such as autonomous driving require fast and accurate performance of several semantic and geometric scene understanding tasks. These signals are critical inputs to down-stream tasks such as obstacle detection and tracking, scene understanding, motion forecasting and motion planning. In this work we focus on jointly solving four tasks that are important for autonomous driving applications: 2-D object detection for high recall detection of dynamic objects (e.g. cars, pedestrians, bicylists); semantic segmentation for understanding image regions (e.g. drivable surface, debris, road boundaries); Instance segmentation for generating precise object boundaries; and monocular depth estimation which enables 3-D scene understanding needed for cross-modality association and motion planning. Additionally, solving all these problems from just camera data provides redundancy within the perception stack.

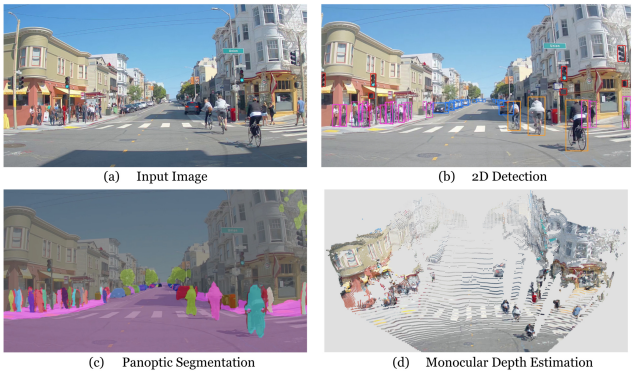

Figure 1. (a) The input image into our QuadroNet architecture, (b) The output of 2D detection head, (c) The output from the panotic segmentation branch. (d) 3D point cloud generated from the monocular depth estimate output by the network.

Computational resources for such applications are often constrained by cost, size and power requirements of the underlying hardware platform. Hence, using a single network to jointly solve multiple perception problems has emerged as a key strategy for satisfying the requirements of fast and accurate results in compute-constrained settings. At runtime, sharing large portions of the network among several tasks reduces the overall inference latency [16]. Furthermore, the training process benefits not only from the sharing of model architecture and parameters among the tasks but also from the ability to introduce richer loss functions that promote task co-learning [15, 19, 21]. However, naively combining multiple tasks into a single network can often lead to reduced accuracy for each task due to limited model capacity.

Other approaches favor combining two or more independently trained networks, each capable of solving a subset of the desired set of tasks into a single network ([28], [31]). While such approaches can leverage state-of-the-art networks for each task, they are unlikely to yield a fast network due to potentially redundant computation.

In this work we propose a novel real-time network,

called *QuadroNet*, that outputs all four signals jointly. Following the basic approach of [15], our proposed network architecture is trained end-to-end in a multi-task setting. Standard jointly-trained multi-task architectures typically have independent task-specific sub-networks that share only a feature extraction backbone. Such networks do not explicitly model inter-task relationships such as the coincidence of object/region boundaries and depth discontinuities. While joint training of these tasks results in improved accuracies for all tasks, we see still greater improvements in accuracy when enforcing cross-task consistency through our novel formulation.

In summary, the key contributions of this paper are:

1. A new formulation for monocular depth estimation that leverages recent state-of-the-art works that output discrete depths ([12], [10]) while improving precise depths via the introduction of continuous depth residuals.
2. A novel instance segmentation formulation that allows us to dramatically improve the accuracy of instance segmentation and monocular depth estimation, and demonstrates the benefits of joint multi-task learning by reasoning about consistency across different tasks in the output space and not just by sharing parameters in the feature extraction backbone.
3. A real-time architecture for jointly outputting 2D bounding boxes, semantic segmentation, instance segmentation and monocular depth at $>$60 fps.

2. Related Work

Joint semantic/geometric models: Recent works have proposed models that address several semantic and geometric tasks at once and are jointly trained. [15] propose a network that outputs semantic segmentation, instance segmentation and monocular depth estimates with homoscedastic task weighting. We also leverage homoscedastic task weighting, but unlike their approach we output 2D boxes, and introduce novel formulations for both instance segmentation and monocular depth estimation. [8] output semantic segmentation and monocular depth in their network but do not produce object detection or instance segmentation. [24] perform 2D object detection followed by 3D box prediction for those detections. [20] perform monocular depth estimation, 2D detection and 3D box prediction in the same network. However, these two networks do not address instance or semantic segmentation, unlike our work.

Monocular depth estimation: [25] demonstrated the possibility of using machine learning to learn the task of monocular depth estimation using a discriminatively-trained Markov Random Field. [11] proposed the use of deep convolutional neural networks in a fully-supervised setting (using lidar data for ground truth) for the monocular depth estimation task. Recent state-of-the-art monocular depth estimation works ([12], [23], [10], [17]) have similarly applied convolutional neural networks (CNNs) with innovations in specialized loss functions, network architectures and layers to achieve impressive results. Following these recent innovations, our work similarly leverages CNNs with lidar supervision, but we use a novel output representation that provides precise depths while also avoiding boundary artifacts (which manifest in 3D "long tails" for objects, as observed by [28] using the DORN model [12]) of previous works.

Segmentation: Previous works have studied semantic and instance segmentation ([6, 14]), typically in isolation. [16] introduced the *panoptic segmentation* task, a unification of the semantic and instance segmentation tasks. They demonstrated the benefits of using the same network for several joint tasks including detection, semantic segmentation and instance segmentation, achieving the same accuracy with a simpler joint network as with several specialized networks. However, their work did not address the 3D geometry of the input image scene or introduce cross task consistency.

Multi-task Learning: Recently there has been a great push towards multi-task learning, motivated by applications that demand efficiency from all their components [26]. Recent works like [15] present a joint task learning framework which uses homoscedastic uncertainities to balance loss for multi-task learning. While we adopt this technique in our work, we go beyond the network architecture described in [15] of a feature extraction backbone shared by the tasks. In addition to having a shared backbone, we modify the task output spaces to enforce consistency between related tasks such as instance segmentation and per-pixel depth, which we call depth-aware instance segmentation (DAIS). Finally, unlike [15] our network also performs the task of 2D bounding box detection. Some other recent works like [21, 19] provides a way to use attention modules to reduce the number of parameters while still performing competitively to a dense baseline. However they to do not model the inter-task relationships explicitly, and hence their multi-task approach fails to perform much better than their baselines. Some other approaches like [32] model the consistency between different task, but require all tasks to be dense pixel wise prediction, leaving out 2D Object Detection.

3. QuadroNet Architecture

Our network architecture consists of a RetinaNet [18] inspired backbone that produces a multi-resolution feature pyramid map which is used as the input to task-specific network branches. To form the backbone, feature maps (starting with the input image) are progressively downsampled and then upsampled again (augmented with skip connections) to form the multi-resolution feature pyramid map (Figure 2a).

2D Detection Task: We formulate the task of 2D detec-

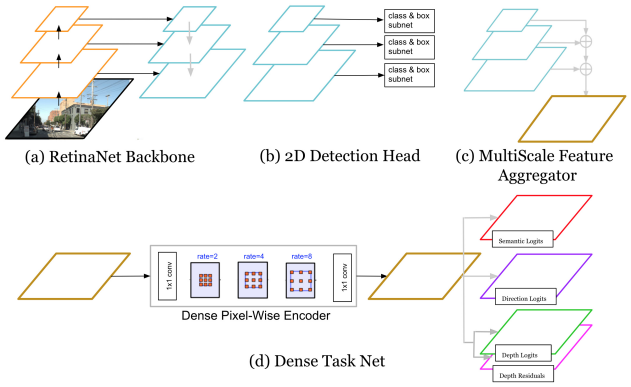

Figure 2. Overview of QuadroNet. (a) The RetinaNet feature extraction backbone gives us access to multi-level features (shown in blue), that capture both high level semantics as well as fine grained structures at high resolutions. (b) For the task of 2D detection, we have a classification and a box regression on every level of the pyramid. (c) For the other dense pixel-wise tasks, we aggregate the feature maps at all levels of the feature pyramid. To do this, all feature maps are upsampled to a common high-resolution and number of channels and combined via summation. (d) We pass this multiscale feature map through a dense pixel-wise encoder in order to develop more high level semantics of the scene required for these tasks. This feature map is task-agnostic, and is then passed through task-specific heads that output the semantic segmentation, direction logit and depth map logits plus residuals for the depth bin.

tion similar to RetinaNet [18]. At each level of the feature pyramid, we attach two structurally identical sub-networks, one for outputting the class and the other to regress the boxes with respect to each anchor as shown in Figure 2b.

Multi Scale Feature Aggregator: For pixel-wise tasks, high-resolution feature maps incorporating context from multiple scales have been shown in the literature ([6], [33]) to produce good results. In our architecture, we create such a feature map by upsampling all pyramid-level feature maps to a common resolution (1/8 scale), and then combining them in a computationally efficient manner via summation (Figure 2c). Similar to the feature map used in the semantic segmentation branch in [16] each upsampling stage consists of 3x3 convolution, batch-norm, ReLU, and 2x bilinear upsampling. This results in a task-agnostic feature-map which is shared across all of our dense pixel-wise tasks, rather than using this for a single task as in [16].

In past works atrous convolutions [5] have been shown to be effective in capturing long-range information [30]. However, they are not well-suited to real-time uses since they are significantly more inefficient on GPUs. Instead of using the atrous convolutions as a context module and tool for spatial pyramid pooling [6], we use them on top of our task-agnostic feature map developed at the end of the multiscale feature aggregation module. This significantly reduces the number of atrous convolutions needed in our backbone, while still preserving most of its advantages and keeping our architecture efficient.

Once we have obtained this task-agnostic high-resolution feature map incorporating information from multiple scales, we apply a bottleneck module that uses a 1x1 convolution to decrease the number of channels, perform a series of atrous convolution with increasing dilation rate (2, 4 and 8) and apply another 1x1 convolution to restore the number of channels. We call this structure the Dense Pixel-Wise Encoder (DPWE), visualized in Figure 2d. Once this feature map is obtained, we supply it into task-specific outputs. We discuss each of these task-specific formulations below.

Semantic Segmentation: A final head with 1x1 convolution, 4x bilinear upsampling, and softmax layers is used to generate the per-pixel class labels at the original image resolution.

Monocular Depth Estimation: As is common for the task of monocular depth estimation [12], we use a spacing-increasing discretization (SID) in order to quantize a depth interval $[\alpha, \beta]$ into K non-overlapping discrete depth bins. This approach divides a given depth interval uniformly in log space to down-weight the training losses in regions with large depth values, so that our network predicts higher accuracy depth for nearby objects.

Following [12], we define the left edge of the i-th depth bin as:

$$t_i = \exp\left(\log\alpha + \frac{\log(\beta/\alpha) * i}{K}\right) \tag{1}$$

where $t_i \in \{t_0, \ldots, t_{K-1}\}$ are the K left edges and t_{i+1} is the corresponding right edge. If a ground-truth depth for a pixel is g_d, it is assigned the bin index $i \in \{0, \ldots, K-1\}$ if $g_d \in [t_i, t_{i+1})$. In our implementation, we have $K = 48$ depth bins spanning the depth range of 1m to approximately 80m.

To further increase the precision of the predicted depth, we follow the MultiBin approach of [22] and predict a residual log depth for each discrete depth bin. We define the midpoint of a bin i in log space as:

$$m_i = \frac{\log(t_{i+1}) + \log(t_i)}{2} \tag{2}$$

For a ground truth depth g_d that falls in bin i, the ground truth residual r_d is computed as:

$$r_d = \frac{(\log(g_d) - m_i)}{\log(t_{i+1}) - \log(t_i)} \tag{3}$$

At inference time, a 1x1 convolution is applied to the DPWE feature map to produce K softmax logits $l_0, \ldots l_{K-1}$ per pixel corresponding to the likelihood that the depth at that pixel falls in corresponding depth bin, e.g. $[t_i, t_{i+1})$ for logit l_i. A parallel 1x1 convolution produces K predicted residuals, $r_0, \ldots, r_{K-1}$ per pixel. The network does not impose any limits on the predicted residual values.

As a result for each pixel K different depths are implicitly predicted, one for each depth bin. For a particular pixel and depth bin i, the predicted depth d_i is decoded as:

$$d_i = \exp(m_i + r_i(\log(t_{i+1}) - \log(t_i))) \quad (4)$$

As noted earlier in [28], some state-of-the-art monocular depth networks exhibit artifacts at the boundaries of objects. We hypothesize that at object boundaries such networks have uncertainty about whether image pixels belong to the object or the background. In these areas, the network may predict an expected depth from a true depth distribution that is bimodal.

Instead of simply selecting the depth $d_{\hat{i}}$ corresponding to the largest logit $l_{\hat{i}}$ per pixel, we compute a smoothed version over a local neighborhood of $N = (\hat{i}-1, \hat{i}, \hat{i}+1)$ as $\sum_{j \in N} P_j d_j$ where $P_j = \exp(l_j) / \sum_{k \in N} \exp(l_k)$, or a probability distribution over the logits in the local neighborhood. This provides a more precise depth estimate in cases where neighboring logits have similar values. As can be seen in our qualitative examples in Figure 4, cars and other foreground objects have sharp boundaries with out approach.

Instance Segmentation : As a baseline, we adopt the formulation proposed in MaskLab [4]. MaskLab generates semantic segmentation logits and direction prediction logits for an image. Given the detected RoIs we first perform RoI pooling of the semantic channel corresponding to the predicted class of the RoI (e.g., the pedestrian channel) from the semantic segmentation logits. In order to exploit the direction information, we perform the same assembling operation in [4] to gather regional logits (specified by the direction) from each direction channel. As shown in Figure 3 the cropped semantic segmentation logits along with the pooled direction logits are then used for foreground/background segmentation. We call this approach Semantic-Aware Instance Segmentation (SAIS).

Depth-Aware Instance Segmentation (DAIS): While semantic segmentation logits might be useful in segmenting out the predicted class of the RoI from the rest of the background, they offer little else in terms of separating instances. By contrast, monocular depth estimates of the image can help us clearly segment instances, as instance boundaries would often coincide with large depth discontinuities. Based on this reasoning, we create a new formulation called Depth Aware Instance Segmentation (DAIS) that incorporates the same pooled direction logits as in the Semantic-Aware Instance Segmentation approach, but replaces the semantic segmentation logits with the monocular depth logits of the RoI as shown in Figure 3.

In addition to improving the accuracy of instance segmentation, this approach incentivizes the monocular depth estimator to predict depth logits that correspond to a coherent instance mask via end-to-end training. This is due to the fact that the instance mask output is a direct function of the monocular depth logits. This also provides dense supervision for the task of monocular depth estimation, which can help where lidar supervision is sparse.

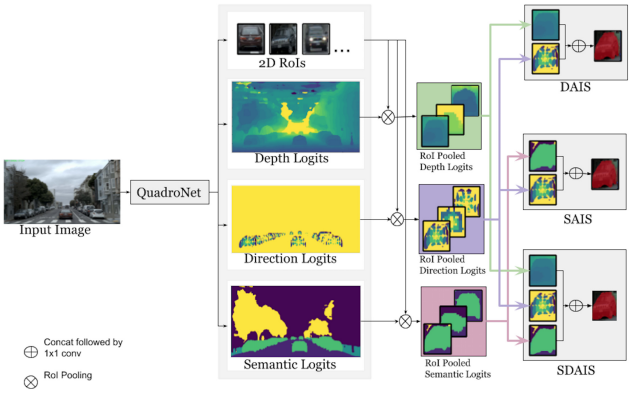

Figure 3. The complete pipeline for our novel formulations of instance segmentation based on different semantic and geometric priors. An image is passed through the QuadroNet architecture, which outputs 2D detection RoIs, monocular depth estimation logits, semantic segmentation logits and direction logits. Each 2D detection RoI is used to perform RoI-pooling of the different logits for different instance segmentation formulation. DAIS uses both the monocular depth logits and direction logits to determine instance boundaries, SAIS uses the same formulation as MaskLab [4], SDAIS uses all of monocular depth, semantic segmentation and direction logits to aggregate support for instance segmentation. After RoI-pooling, the feature-maps in each of these formulation are concatenated on the channel dimension and a 1x1 convolution is performed to output a binary mask for the instance. This predicted mask is backpropogated against the ground truth from the dataset.

Semantic & Depth-Aware Instance Segmentation (SDAIS) : For completeness, we also experiment with using both the semantic segmentation logits and monocular depth logits in conjunction with the direction logits, and call this formulation Semantic & Depth-Aware Instance Segmentation (SDAIS). This formulation is shown in Figure 3 as well. As can be seen from the figure, for each detected RoI we use the semantic segmentation logit of the predicted class, the pooled direction logits as well as the monocular depth logits for the RoI. All three are then concatenated followed by a simple 1x1 convolution to estimate the instance mask.

In this formulation we jointly reason about all four tasks together in the same output space. We are using the 2D boxes predicted by the detection head along with the predicted class, and we are directly using the predicted logits for semantic segmentation, instance center direction and monocular depth estimation. As our network is trained end-to-end, this provides us a way to jointly model the output space of all these tasks, thus allowing them to learn features that are cross-task consistent. We show empirically in Tables 2 and 4 that this yields stronger consistency than

jointly learning the feature extraction backbone, and having independent output spaces for each task.

Joint Training and Loss functions: Our multi-task framework has both categorical (e.g. classification head for 2D detection, semantic segmentation) as well as regression (e.g. 2D bounding box regression, mono-depth residuals etc.) outputs. For 2D detection we have a classification loss ($\mathcal{L}_c$ for 2D boxes) as well as a regression loss ($\mathcal{L}_b$ with respect to anchors for the boxes), following the same formulation as Lin et al. [18]. For the task of semantic segmentation, we have a pixel-wise loss for classifying each pixel into one of the semantic classes $\mathcal{L}_s$. To predict the direction logits, we also have a classification loss, to classify each pixel into one of the direction bins $\mathcal{L}_d$. The formulation of the direction logits is based on prior work in [4]. Our approach for monocular depth estimation outputs for each pixel a set of logits corresponding to discrete depth bins. This uses a softmax cross-entropy loss as other classification tasks, represented by $\mathcal{L}_m$. We also output a residual of the depth to output a continuous depth offset to the depth bin centers to provide higher-precision depth estimates. We use a smoothed-L1 loss to supervise the residuals represented by $\mathcal{L}_r$. Along with these, we have an instance mask-loss that is applied after the instance segmentation branch. We use a binary cross entropy loss to predict this mask and denote it by $\mathcal{L}_i$. As mentioned above, these different losses might have very different scales and units. A naive approach would be to use hand-tuned parameters λ_t for the t-th task's loss $\mathcal{L}_t$ to come up with a joint loss for end-to-end training. The total loss $\mathcal{L}_{total}$ using this formulation would be: $\mathcal{L}_{total} = \sum_t^T \lambda_t \mathcal{L}_t$.

Brute-force search to tune the hyper-parameters would be very expensive and might result in sub-optimal solutions, as the space of hyper-parameters increases exponentially in the number of tasks. Hence we use the homoscedastic task uncertainty approach in [15], introducing an additional network parameter σ_t as a measure uncertainty for each task $t \in [1, \ldots, T]$. For mathematical stability, we use $s_t = \log \sigma_t^2$ as the parameter of the network. Thus this modified total loss function for training $\mathcal{L}_{total}^h$ is given by:

$$\mathcal{L}_{total}^h = \sum_t \tau_t \exp(-s_t)\mathcal{L}_t + \frac{s_t}{2} \tag{5}$$

where τ_t is 1 for classification tasks and 0.5 for regression tasks.

Task	ADD	Existing Dataset
2D Detection	2,261,677 / 143,459	7481 / 7518 (KITTI [13])
Panoptic Segmentation	104,587 / 2,363	2975 / 1525 (Cityscapes [9])
Monocular Depth	111,720 / 10,968	23,488 / 697 (KITTI [13])

Table 1. Comparison of ADD to popular existing dataset in each domain for number of (train / test) images for relevant for self-driving.

4. Dataset

We are not aware of any publicly available dataset which allows for jointly training and evaluating the four tasks of interest with a reasonable amount of labeled data for all of the tasks. Instead, we collected a very large scale dataset on which we trained and evaluated our approach. Our dataset consists of more than two million 1920x1200 sized images collected in different cities, in different weather and lighting conditions, using cameras and lidars mounted to provide a 360-degree field-of-view around a vehicle. Images in the dataset are hand-labeled with one or more 2D bounding boxes for 3 classes (cars, pedestrians and Cyclists same as KITTI [13]), pixel-wise semantic segmentation and pixel-wise instance segmentation (with the same set of semantic classes as Cityscapes [9]). We also generate sparse pixel-wise ground truth depths from lidar data. We present a comparison of this dataset, which we title ADD (Autonomous Driving Dataset) with the most popular existing datasets in Table 1. However, for comparison of our work to other multi-task learning system we publish results on the Cityscapes dataset. The Cityscapes dataset has semantic and instance segmentation groundtruth available for 5000 images (split into train, validation and test sets), they also make an estimate of depth available using SGM algorithm on stereo images. Other recently released dataset like nuScenes [1] and argoverse [2] lack the semantic segmentation groundtruth essential for this work.

5. Experimental Evaluation

We split our metrics into Table 2, 3 and 4 to study each task with relevant baselines. However experiments with the same name across tables are the exact same model, just evaluated on the task relevant for the table.

Semantic Segmentation & Instance Segmentation Results: We summarize all the results for both semantic segmentation and instance segmentation in Table 2. We use the same definition of IoU and mAP@0.5 as defined by the Cityscapes evaluation suite. We observe, that when we add the monocular depth task to (2D Detection + SS + SAIS), we see small improvements in all classes for semantic segmentation, especially the road class, which improves from 94.97% (2D Detection + SS + SAIS) to 95.63% (2D Detection + SS + SAIS + MD). However, when we introduce the DAIS formulation, we see a huge increment in the mAP0.5 metric for instance segmentation versus SAIS, from 54.78% (2D Detection + SS + SAIS) to 59.69% (2D Detection + SS + DAIS + MD) which is a +4.91% increase. We believe this increase is due to the fact that the semantic segmentation logits which are an input in the SAIS formulation do not distinguish among instances of the same semantic classes. They simply help separating the foreground semantic class from the rest of the background. However, monocular depth logits provide information about individual instance bound-

Method	Dataset	Semantic Segmentation IoU					IS	Time
		mIoU	Car	Person	Rider	Road	mAP@0.5	[ms]
FC-HarDNet-70 [3]	ADD	69.3	83.19	52.69	52.35	95.69	NA	19.7
2D Detection + SS		71.13	84.44	63.44	67.95	93.88	NA	18.7
2D Detection + SS + SAIS		73.10	86.22	65.08	70.64	94.97	54.78	20.1
2D Detection + SS + SAIS + MD		73.09	86.28	65.10	70.89	95.63	54.81	21.3
2D Detection + SS + DAIS + MD		74.22	87.18	66.07	71.62	**98.12**	59.69	21.8
2D Detection + SS + SDAIS + MD		**74.28**	**87.27**	**66.12**	**71.97**	98.09	**59.83**	21.95
DeepLabv3+ [7]	ADD	77.81	89.40	69.52	75.12	97.37	NA	>1k
FC-HardNet-70 [3]	Cityscapes [9]	75.85	95.66	84.52	67.36	98.51	NA	15.1
Kendall *et al.* [15]		78.54	95.30	84.91	69.54	**98.42**	39.0	>1k
2D Detection + SS + SAIS		79.15	95.81	85.26	69.98	97.95	50.52	18.6
2D Detection + SS + SAIS + MD		79.17	95.78	85.19	70.03	97.83	50.68	19.8
2D Detection + SS + DAIS + MD		80.64	**96.17**	85.89	71.27	98.27	53.15	20.0
2D Detection + SS + SDAIS + MD		**80.73**	96.10	**86.65**	**71.46**	98.39	**53.19**	20.1
DeepLabv3+ [7]	Cityscapes [9]	82.13	96.40	87.95	73.26	98.69	NA	>1k

Table 2. Comparison of semantic segmentation (SS) and instance segmentation (IS) metrics. **First:** these metrics present an ablative analysis of our techniques and a comparison with a real-time network trained on our ADD dataset, FC-HarDNet-70 [3]. We observe a huge improvement for DAIS as compared to SAIS for mAP@0.5 on the IS task. Joint training for tasks using our framework results in improvements across the board. Overall, we observe an improvement in both the mIoU metrics for SS and mAP@0.5 for IS when using the SDAIS formulation over the baseline of just solving for Detection + SS. Our SDAIS approach also performs better than the FC-HardNet-70 network by a significant amount across all categories while running more quickly. **Second:** we provide results of a non-real-time, state of the art method, DeepLabv3+, trained and evaluated on ADD, to compare with our methods. Note, both FC-HardNet-70 and DeepLabv3+ solve just one task (SS), while QuadroNet performs up to four tasks. **Third:** we also compare our performance on Cityscapes with other multi-task learning approaches like Kendall *et al.* [15] and real-time approaches like FC-HardNet-70 [3]. **Fourth:** we also provide results of DeepLabv3+ [7] on Cityscapes [9] for completeness.

Method	Dataset	Average Precision (AP)								
		Car			Pedestrian			Cyclist		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
SqueezeDet [29]	ADD	87.82	81.47	72.84	75.85	65.19	62.47	80.66	65.41	61.15
2D Detection		96.84	91.18	83.19	90.91	84.19	81.77	89.81	74.19	72.88
2D Detection + SS		97.42	91.89	83.71	92.17	86.85	81.55	90.58	75.91	72.91
2D Detection + SS + SAIS		97.32	91.85	**84.92**	93.82	87.72	84.25	**91.78**	77.67	74.59
2D Detection + SS + SAIS + MD		98.06	91.97	84.66	**94.92**	**88.64**	85.41	91.54	77.42	74.12
2D Detection + SS + DAIS + MD		98.12	92.07	84.47	94.79	**88.64**	85.48	91.54	**77.82**	**74.74**
2D Detection + SS + SDAIS + MD		**98.13**	**92.09**	84.45	94.81	88.63	**85.53**	91.57	77.81	74.72

Table 3. Detection metrics for our various approaches on ADD. We observe steady improvements in detection mAP metrics with introduction of related tasks. We also compare with other state-of-the-art real-time approaches like SqueezeDet [29].

aries via depth discontinuities. The SDAIS approach (2D Detection + SS + SDAIS + MD) improves on these results a bit further, providing the best overall result for both semantic segmentation and instance segmentation.

We also provide a comparison of our approach against a state-of-the-art, real-time, single-task network architecture, FC-HarDNet-70 [3]. Our approach performs significantly better across several metrics while having faster runtime, even while performing four tasks vs. the single task (semantic segmentation) of FC-HarDNet-70. As an additional comparison, we retrain DeepLabv3+ [7] on our ADD dataset and evaluate on the same test set on which we evaluate our models. DeepLabv3+ [7] takes more than 1 second per image to process while providing only somewhat better results than our approach. Finally we also compare our approach to other multi-task learning methods like Kendall *et al.* [15], other real-time approaches like FC-HardNet-70 [3], and other state-of-the-art models like DeepLabv3+ [7] on Cityscapes [9]. For multi-task as well as real-time methods, SDAIS gives the overall best results 80.73% mIoU as compared to 78.54% mIoU for Kendall *et al.* [15] and 75.85% for FC-HardNet-70 [3]. This further proves the advantages of jointly reasoning about output spaces of different tasks (allowing learning of joint representations) rather than training unrelated tasks in a multi-task setting.

2D Detection Results: We evaluate object detection

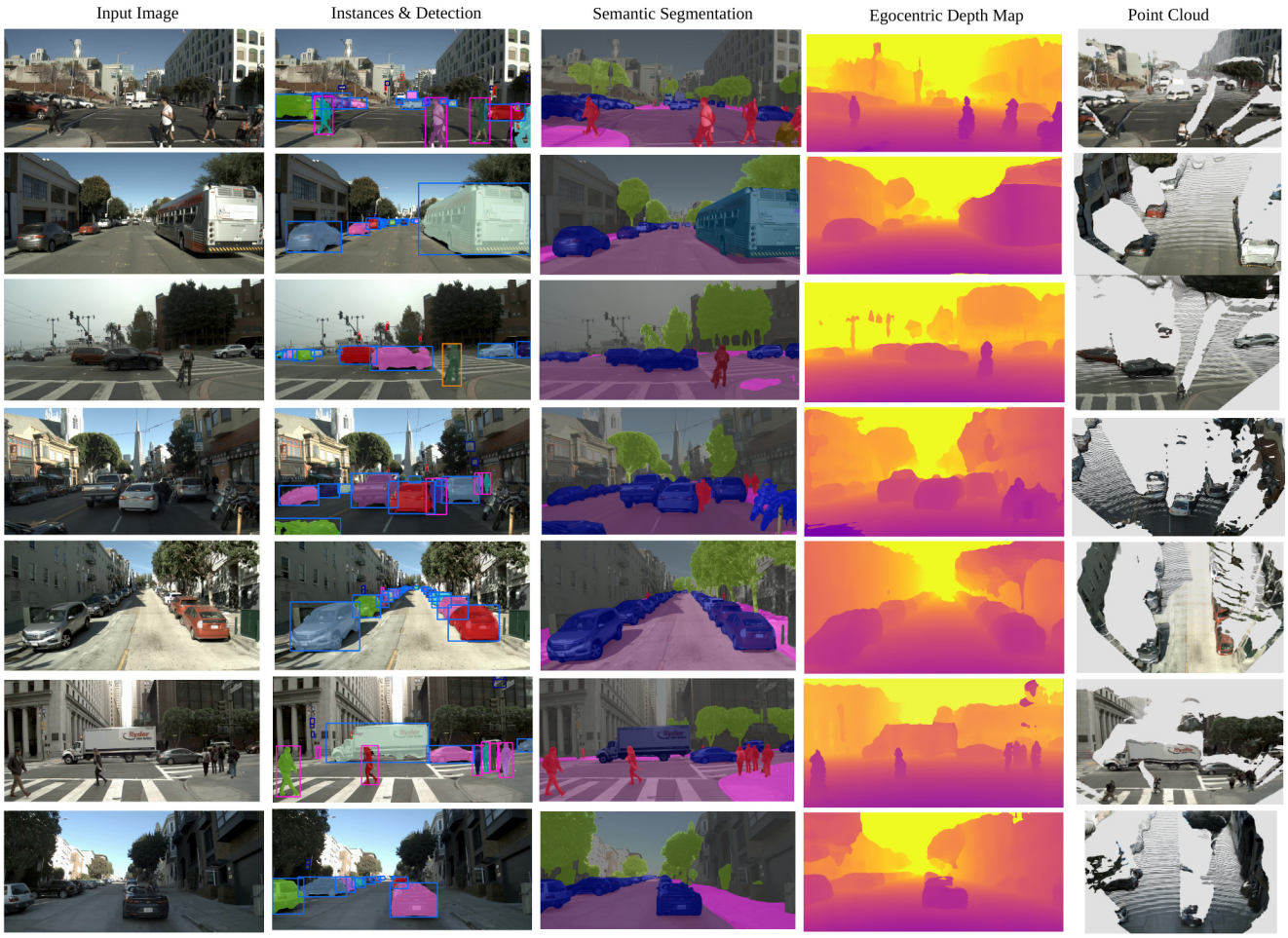

Figure 4. Qualitative examples of the outputs of our network. From left to right, columns show the original input image, 2D detections and instance segmentation, semantic segmentation, egocentric depth map and a point cloud generated using the depth map and colored by the image pixels. More results are available in the supplementary material. **Best viewed in color.**

performance using the KITTI criteria, which requires an intersection-over-union overlap score of 0.7 for cars and 0.5 for pedestrians and cyclists. We also use the same definition of "Easy", "Medium" and "Hard" as KITTI. We report the average precision (AP) numbers for each class and difficulty in Table 3. When we jointly train our network for the task of semantic segmentation along with 2D detection (2D Detection + SS), we see that performance improves over just the baseline (2D detection); the "Moderate" task improves by +0.71% (91.18% to 91.89%) for "car", +2.66% (84.18% to 86.85%) for "pedestrian" and +1.72% (74.19% to 75.91%) for "cyclist". Furthermore, when we also train for instance segmentation using the SAIS formulation (2D Detection + SS + SAIS) we see further improvements over 2D Detection + SS; "pedestrians" improve by +0.87% (86.85% for to 87.72%), cyclists by +1.76% (75.91% to 77.67%), and "Hard" cars improve by +1.21% (83.71% to 84.92%). When we also jointly train for monocular depth estimation (2D Detection + SS + SAIS + MD), the metrics seem to benefit similarly as now we add more information about the world as we reason about the depth for each pixel. We continue to see small improvements as we experiment with the DAIS and SDAIS formulations of instance semantic segmentation as well. We also compare our work with other state-of-the-art real time detectors like SqueezeDet [29] and empirically show our methods consistently outperform such single task approaches.

Monocular Depth Estimation Results: We evaluate our monocular depth estimation using the threshold metric from [11] ($\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$); δ is defined as $\max(\frac{y^*}{y}, \frac{y}{y^*})$ where y is the predicted depth and y^* is the ground truth depth) and summarize the results for our experiments in Table 4. We focus on this metric for two reasons: first, it is less sensitive to outlier depths than metrics that are computed via an average, e.g. RMSE. Secondly, it provides a bound on the number of "functional" depths, or depths that meet a sufficiently rigorous quality metric. This a met-

Method	Dataset	All			Car			Person		
		$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
2D Detection + SS + SAIS + MD	ADD	87.84	93.17	95.46	83.81	89.33	92.33	74.05	87.52	93.10
2D Detection + SS + DAIS + MD		**88.68**	93.42	95.57	84.36	89.45	92.39	74.77	87.46	92.99
2D Detection + SS + SDAIS + MD		88.67	**93.50**	**95.63**	**84.48**	**89.58**	**92.49**	**75.38**	**87.73**	**93.15**

Table 4. Comparison of monocular depth metrics for our methods. Both DAIS and SDAIS show significant improvements over naively training monocular depth as a joint task with SAIS. Values are percentages.

Method	All		Car		Person		Road	
	$\delta < 1.01$	$\delta < 1.25$	$\delta < 1.01$	$\delta < 1.25$	$\delta < 1.01$	$\delta < 1.25$	$\delta < 1.01$	$\delta < 1.25$
W/o residuals	16.90	88.65	14.92	84.45	8.50	75.30	22.60	98.45
With residuals	**19.74**	**88.67**	**15.04**	**84.48**	**8.53**	**75.38**	**30.27**	**98.46**

Table 5. Ablative analysis of residuals on monocular depth accuracy on the SDAIS formulation. To better study high accuracy monocular depth estimates, we introduce $\delta < 1.01$ in addition to the threshold metric from [11]; values are percentages. Under this metric we can see significant improvements, especially "road".

ric that can be incorporated in estimating the overall safety of a safety-critical system, while a metric that is an average could have a high (and perhaps unknowable) proportion of depths that do not meet a particular safety requirement. Upon comparing the SAIS formulation versus the SDAIS formulation, we see an increase in $\delta < 1.25$ of +0.83% (Table 4, "All" category; from 87.84% to 88.67%) as well as increases in metrics in other semantic categories. We postulate that this is in part because our unique DAIS and SDAIS formulations allow for dense supervision (by predicting a dense pixel-wise instance mask using the monocular depth logits) for the task of monocular depth estimation, which is not possible using only the sparse lidar supervision.

We also study the effect of our MultiBin output approach over simply outputting discrete depth. To study high-precision depth details, we introduce an additional metric, the percentage of pixels for which $\delta < 1.01$. As an example, this provides a tolerance of 20cm at a distance of 20m, allowing the system to perceive important details such as potholes in the road surface. We find the addition of continuous depth residuals significantly improves the proportion of pixels with highly accurate depths, as seen in Table 5, particularly for drivable surface pixels, which saw an increase from 22.60% to 30.27%.

Qualitative Results: We present qualitative outputs of our model in Figure 4. Despite its limited computational budget, our model produces high quality 2D object detections, instance segmentations, semantic segmentations and depths (shown as both an egocentric depth map and as 3D point clouds).

Figure 5 shows examples of the improvement in monocular depth from the SDAIS formulation over the SAIS formulation. As expected, depths near the boundary edges of the object are improved. From the pedestrian leg in Figure 5a, we can see how conditioning the instance mask on the depth estimate results in depth consistent with instance boundaries. For the case of the bus in Figure 5b, we see how

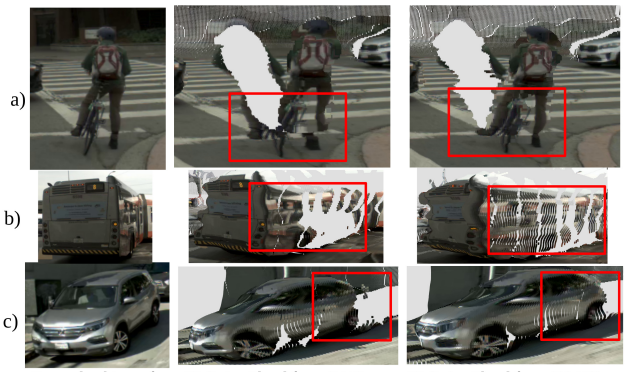

Figure 5. Examples of improvement in monocular depth estimates due to SDAIS formulation over SAIS formulation. We have highlighted the differences between the two approaches with a red box.

the instance mask acts as a dense structural prior in contrast to sparse lidar supervision for longitudinally oriented objects, resulting in much more geometrically coherent depth estimates for objects.

Inference: All experiments are run on a Tesla-V100 GPU, which has 16GB of DRAM and 112 theoretical fp16 TFLOPS. We run QuadroNet on the NVIDIA TensorRT [27] framework, with fp16 precision for all results. We also run FC-HarDNet-70 [3] and DeepLabv3+ [7] on the same GPU when reporting timing and accuracy in Table 2.

6. Conclusion

We demonstrated the efficacy of a single network trained to jointly perform a set of relevant scene-understanding tasks: 2D object detection, semantic segmentation, instance segmentation and monocular depth estimation. These tasks cover a broad range of semantic and geometric use cases essential for important robotics applications such as drones and autonomous driving. We also introduced a network architecture that allows for real-time performance and facilitates task co-learning. Finally, we showed that training benefits from combining these tasks via our novel consistency prior for instance segmentation using monocular depth, improving the performance of both tasks. Interesting directions for future work could include the addition of 3D boxes for objects as an output of the network, or using depth self-supervision using stereo images could reduce or eliminate the need for lidar data for ground truth depth data.

References

[1] 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. *arXiv preprint arXiv:1903.11027*, 2019.

[2] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] Ping Chao, Chao-Yang Kao, Yu-Shan Ruan, Chien-Hsiang Huang, and Youn-Long Lin. Hardnet: A low memory traffic network. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[4] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4013–4022, 2018.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 833–851, 2018.

[8] Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir A. Rawashdeh. Auxnet: Auxiliary tasks enhanced semantic segmentation for automated driving. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2019, Volume 5: VISAPP, Prague, Czech Republic, February 25-27, 2019.*, pages 645–652, 2019.

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] Raul Diaz and Amit Marathe. Soft labels for ordinal regression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4738–4747, 2019.

[11] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2366–2374, 2014.

[12] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2002–2011, 2018.

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

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, Oct 2017.

[15] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7482–7491, 2018.

[16] Alexander Kirillov, Ross B. Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6399–6408, 2019.

[17] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *CoRR*, abs/1907.10326, 2019.

[18] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007, 2017.

[19] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. *CoRR*, abs/1803.10704, 2018.

[20] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. ROI-10D: monocular lifting of 2d detection to 6d pose and metric shape. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2069–2078, 2019.

[21] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. *CoRR*, abs/1904.08918, 2019.

[22] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5632–5640, 2017.

[23] Sudeep Pillai, Rares Ambrus, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 9250–9256, 2019.

[24] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. In *The Thirty-Third AAAI Conference on Artificial*

Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019., pages 8851–8858, 2019.

[25] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 1161–1168, 2005.

[26] Trevor Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *CoRR*, abs/1905.07553, 2019.

[27] Nvidia TensorRT. https://developer.nvidia.com/tensorrt.

[28] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[29] Bichen Wu, Forrest N. Iandola, Peter H. Jin, and Kurt Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. *CoRR*, abs/1612.01051, 2016.

[30] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Bridging category-level and instance-level semantic image segmentation. *CoRR*, abs/1605.06885, 2016.

[31] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2345–2353, 2018.

[32] Amir Zamir, Alexander Sax, Teresa Yeo, Oğuzhan Kar, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas Guibas. Robust learning through cross-task consistency. *arXiv*, 2020.

[33] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6230–6239, 2017.