Method	Frames	Car (IOU = 0.7)			Pedestrian (IOU = 0.5)			Cyclist (IOU = 0.5)		
		Easy	**Mod.**	Hard	Easy	**Mod.**	Hard	Easy	**Mod.**	Hard
Kinematic3D [4]	4	26.69	17.52	13.10	–	–	–	–	–	–
ROI-10D [38]	1	9.78	4.91	3.74	–	–	–	–	–	–
UR3D [64]	1	21.85	12.51	9.20	–	–	–	–	–	–
MonoPSR [22]	1	18.33	12.58	9.91	7.24	4.56	4.11	**9.87**	**5.78**	**4.57**
MonoDIS [52]	1	17.23	13.19	11.12	–	–	–	–	–	–
M3D-RPN [3]	1	21.02	13.67	10.23	5.65	4.05	3.29	1.25	0.81	0.78
Mono3D-PLiDAR [61]	1	21.27	13.92	11.25	–	–	–	–	–	–
RTM3D [29]	1	19.17	14.20	11.99	–	–	–	–	–	–
SMOKE [33]	1	20.83	14.49	12.75	–	–	–	–	–	–
MonoPair [12]	1	19.28	14.83	12.89	**10.99**	**7.04**	**6.29**	4.76	2.87	2.42
RAR-Net [32]	1	22.45	15.02	12.93	–	–	–	–	–	–
D4LCN [13]	1	22.51	16.02	12.55	5.06	3.86	3.59	2.72	1.82	1.79
PatchNet [36]	1	22.97	16.86	**14.97**	–	–	–	–	–	–
MoVi-3D [53]	1	22.76	17.03	14.85	10.08	6.29	5.37	1.45	0.91	0.93
AM3D [37]	1	**25.03**	**17.32**	14.91	–	–	–	–	–	–
CaDDN (ours)	1	**27.94**	**18.91**	**17.19**	**14.72**	**9.41**	**8.17**	**9.67**	**5.38**	**4.75**
Improvement	–	*+2.91*	*+1.59*	*+2.22*	*+3.73*	*+2.37*	*+1.88*	*-0.20*	*-0.40*	*+0.18*

Table 5. BEV detection results on the KITTI [16] *test* set. Results are shown using the $\text{AP}|_{R_{40}}$ metric only for results that are readily available. We indicate the highest result with **red** and the second highest with **blue**.

Exp.	Disc. Method	Car (IOU = 0.7)		
		Easy	Mod.	Hard
1	UD	20.40	15.10	12.75
2	SID	22.00	15.65	13.26
3	LID	**23.57**	**16.31**	**13.84**

Table 6. CaDDN Depth Discretization Ablation Experiments on the KITTI *val* set using $\text{AP}|_{R_{40}}$. See Section 3.3 for description of depth discretization methods.

Exp.	Block	$W_F \times H_F$	Car (IOU = 0.7)		
			Easy	Mod.	Hard
1	*Block1*	$\frac{W_I}{4} \times \frac{H_I}{4}$	**23.57**	**16.31**	**13.84**
2	*Block2*	$\frac{W_I}{8} \times \frac{H_I}{8}$	20.57	14.86	12.93
3	*Block3*	$\frac{W_I}{8} \times \frac{H_I}{8}$	19.60	14.65	12.73
4	*Block4*	$\frac{W_I}{8} \times \frac{H_I}{8}$	20.71	14.94	12.91

Table 7. CaDDN Feature Resolution Ablation Experiments on the KITTI *val* set using $\text{AP}|_{R_{40}}$. Block indicates the Image Backbone block where image features $\tilde{\mathbf{F}}$ are extracted from (See Figure 2). $W_F \times H_F$ indicates the width and height of image features $\tilde{\mathbf{F}}$.

A. Additional Results

A.1. KITTI Dataset Results

Table 5 shows the results of CaDDN on the KITTI [16] *test* set for BEV detection. Our method outpeforms previous single frame methods by large margins on $\text{AP}|_{R_{40}}$ of +2.91%, +1.59%, and +2.22% on the Car class on the Easy, Moderate, and Hard difficulties respectively. Our method outperforms the previous state-of-the art method on the Pedestrian class MonoPair [12] with margins on $\text{AP}|_{R_{40}}$ of +3.73%, +2.37%, and +1.88%. Our method achieves first or second place on the Cyclist class with margins on $\text{AP}|_{R_{40}}$ of -1.37%, -1.33%, and +0.18% relative to MonoPSR [22].

A.2. Ablation Studies

Depth Discretization. Table 6 shows the detection performance of CaDDN with each of the depth discretization methods outlined in Section 3.3. We observe that LID offers the highest performance, leading to its use for our method.

Feature Resolution. Table 7 shows the detection peformance of CaDDN when modifying the image feature extraction layer in the Image Backbone (See Figure 2). Experiment 1 shows the performance when image features are extracted from *Block1*. Experiments 2, 3, and 4 shows reduced performance when smaller resolution image features are extracted. Smaller spatial resolutions in the image features cause oversampling in the frustum to voxel grid transformation, leading to many voxel features $\mathbf{V}$ with similar features (See Section **??**).

B. Additional Details

Depth Distributions. We add an depth bin to our depth distributions $\mathbf{D}$ that represents any depth outside the range $[d_{min}, d_{max}]$. The added bin is included in the depth distribution loss L_{depth}, but is removed when generating frustum features $\mathbf{G}$.

Training Details. We initialize the Image Backbone and Depth Distribution Network (See Figure 2) using the DeepLabV3 [6] model pretrained on the MS-COCO dataset [31]. All other components are randomly initialized.