

Distilled Semantics for Comprehensive Scene Understanding from Videos – Supplementary material

Fabio Tosi* Filippo Aleotti* Pierluigi Zama Ramirez*
Matteo Poggi Samuele Salti Luigi Di Stefano Stefano Mattoccia
Department of Computer Science and Engineering (DISI)
University of Bologna, Italy

*{fabio.tosi5, filippo.aleotti2, pierluigi.zama}@unibo.it

Supplementary material

This document provides additional material concerning CVPR 2020 paper, “Distilled Semantics for Comprehensive Scene Understanding from Videos”. In particular, we report here a more detailed description of our Ω Net architecture and the losses used to train it, alongside with more insights related to performance in the addressed tasks (depth, pose, optical flow, semantic and motion segmentation) and runtime. Moreover, we include additional qualitative results on KITTI (K) and CityScapes (CS), as well as on an arbitrary YouTube video for which the camera parameters are not known in advance, thus showing how Ω Net can provide comprehensive scene understanding in the wild.

1. Network Architecture

In this section, we provide a more detailed description of our Ω Net architecture.

Table 1 reports a detailed specification of the layers building up the DSNet and CamNet modules. For each layer, we report kernel size (K), stride (S) and number of input/output channels. As for OFNet and the proxy semantic network, a thorough description can be found in [17] and [6] respectively.

2. Losses

To train the DSNet module, we rely on a multi-task loss function based mainly on two terms. In particular, a depth term is in charge of minimize the discrepancy between the target image I_t and an image I_s , warped as \tilde{I}_t^s , from a monocular sequence while a semantic term is used to learn semantic labels from proxy label distilled by a pre-trained network.

Depth term. According to the self-supervised training paradigm proposed in [13], we adopt a photometric loss function consisting in a weighted combination between the Structural Dissimilarity Measure (DSSIM) and the standard \mathcal{L}_1 loss. In addition, a per-pixel minimum strategy [14] is used to solve occlusion/disocclusion by simply picking the minimum error between each pair I_t and I_s instead of averaging them. Thus, the photometric loss function is defined as:

$$\mathcal{L}_{ap}^D = \sum_p \min_s (\alpha \mathcal{L}_{DSSIM}(p) + (1 - \alpha) \|I_t(p) - \tilde{I}_t^s(p)\|_1) \quad (1)$$

where p denotes pixel coordinates, \tilde{I}_t^s a source image I_s warped according to estimated depth and pose and the $DSSIM$ loss function is computed as:

$$\mathcal{L}_{DSSIM}(p) = \frac{1 - SSIM(I_t(p), \tilde{I}_t^s(p))}{2} \quad (2)$$

In our experiments, we set $\alpha = 0.85$.

*Joint first authorship.

Layer	K	S	In/Out	Input
Deep feature extractor (DSE)				
conv1a	3	2	3/16	input
conv1b	3	1	16/16	conv1a
conv2a	3	2	16/32	conv1b
conv2b	3	1	32/32	conv2a
conv3a	3	2	32/64	conv2b
conv3b	3	1	64/64	conv3a
conv4a	3	2	64/128	conv3b
conv4b	3	1	128/128	conv4a
conv5a	3	2	128/256	conv4b
conv5b	3	1	256/256	conv5a
Estimator (E)				
conv1	3	1	i_channels/64	input
conv2	3	1	64/48	conv1
conv3	3	1	48/32	conv2
conv4	3	1	32/16	conv3
Context (C)				
disp1	3	1	i_channels/64	input
disp2	3	1	64/32	disp1
disp3	3	1	32/16	disp2
disp	3	1	16/1	disp2
Disparity and Semantic Tower				
conv5	3	1	i_channels/16	E(conv5b)
disp5	3	1	i_channels//1	C(conv5)
conv4	3	1	i_channels/16	E(conv4b, disp5 ↑)
disp4	3	1	i_channels//1	C(conv4, conv5 ↑) + disp5 ↑
conv3	3	1	i_channels/16	E(conv3b, disp4 ↑)
disp3	3	1	i_channels//1	C(conv3, conv4 ↑) + disp4 ↑
conv2	3	1	i_channels/16	E(conv2b, disp3 ↑)
disp2	3	1	i_channels//1	C(conv2, conv3 ↑) + disp3 ↑
conv1	3	1	i_channels/16	E(conv1b, disp2 ↑)
disp1	3	1	i_channels//1	C(conv1, conv2 ↑) + disp2 ↑
sem	3	1	i_channels//1	C(conv1, conv2 ↑) + disp2 ↑

Layer	K	S	In/Out	Input
Deep feature extractor (DFE)				
conv1a	3	2	3/16	input
conv1b	3	1	16/16	conv1a
conv2a	3	2	16/32	conv1b
conv2b	3	1	32/32	conv2a
conv3a	3	2	32/64	conv2b
conv3b	3	1	64/64	conv3a
conv4a	3	2	64/128	conv3b
conv4b	3	1	128/128	conv4a
Pose Estimator				
conv1a	3	1	i_channels/128	DFE_t, DFE_s
conv1b	3	2	128/128	conv1a
conv2a	3	1	128/256	conv1b
conv2b	3	2	256/256	conv2a
pose	1	1	256/6*N	conv2b
Intrinsic Estimator				
focals	1	1	i_channels/2	conv2b
offsets	1	1	i_channels/2	conv2b

Table 1. Detailed structure of the DSNet (left) and CamNet (right) modules in Ω Net. The symbol “;” means concatenation, while \uparrow indicates upsampling.

A smoothness term is also used to penalize large disparity differences between adjacent pixels when the former do not co-occur with strong RGB gradients:

$$\mathcal{L}_{smooth} = \sum_p |\nabla D_t(p)| \cdot \left(e^{-|\nabla I_t(p)|} \right)^T \quad (3)$$

Finally, we mask-out pixels whose appearance do not change between consecutive frames, which includes scenes with no relative motion. This has the effect of letting the network ignore pixels which move at the same velocity as the camera, and even to ignore whole frames when the camera stop moving. According to [14], this is accomplished by removing those pixels which have an unwarped photometric loss smaller than the corresponding warped photometric loss, *i.e.*

$$\mu = \min \mathcal{L}_{ap}^D(I_t, I_s) > \mathcal{L}_{ap}^D(I_t, I_t^s) \quad (4)$$

Semantic term. The standard cross-entropy loss between the predicted and proxy pixel-wise semantic labels is used as semantic term:

$$\mathcal{L}_{sem} = -(S_t \log(S_P) + (1 - S_t) \log(1 - S_P)) \quad (5)$$

where S_t is the semantics predicted by DSNet and S_P the ground-truth proxy label. Moreover, as proposed in [24] we employ a cross-task loss to tighten the link between the learning tasks dealing with depth and semantics:

$$\mathcal{L}_{cdd} = \sum_p \text{sgn}(|\nabla S_t(p)|) \cdot \left(e^{-|\nabla D_t(p)|} \right)^T \quad (6)$$

Hence, the total loss used to train DSNet is a weighted combination of the above losses:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{ap}^D + \lambda_2 \mathcal{L}_{smooth} + \lambda_3 \mathcal{L}_{sem} + \lambda_4 \mathcal{L}_{cdd} \quad (7)$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are hyper-parameters. In our experiments, we set $\lambda_1 = 1, \lambda_2 = 0.1, \lambda_3 = 1$ and $\lambda_4 = 0.1$.

As described in the paper, for the Optical Flow we rely on a peculiar training schedule based on two components in Ω Net, which are referred as OFNet and SD-OFNet.

Optical Flow term. We train a the first instance of the optical flow network (OFNet) using the same photometric loss as for DSNet:

$$\mathcal{L}_{ap}^{OF} = \sum_p \alpha \mathcal{L}_{DSSIM} + (1 - \alpha) \|I_t - \tilde{I}_t^s\|_1 \quad (8)$$

In this case, however, \tilde{I}_t^s is warped according to estimated flow. Akin to DSNet, we set $\alpha = 0.85$.

Self-Distilled Optical Flow term. The self-distilled optical flow network (SD-OFNet), instead, is trained in a quite different manner. In fact, given the optical flow $F_{t \rightarrow s}$ predicted by OFNet, the rigid flow $F_{t \rightarrow s}^{rigid}$ and the mask M , we leverage on the optical flow in the regions where $F_{t \rightarrow s}$ and $F_{t \rightarrow s}^{rigid}$ are similar as well as on moving objects, while we rely on the rigid flow for the remaining areas (*e.g.*, occlusions due to camera motion). We can distinguish the former regions from the latter ones looking at M . Moreover, we also apply a photometric term ϕ on the predicted optical flow $SF_{t \rightarrow s}$. The final loss \mathcal{L} to train SD-OFNet is given by:

$$\mathcal{L} = \sum \alpha_r \phi(SF_{t \rightarrow s}, F_{t \rightarrow s}^{rigid}) \cdot (1 - M) + \alpha_d \phi(SF_{t \rightarrow s}, F_{t \rightarrow s}) \cdot M + \psi(I_t, \tilde{I}_t^{SF}) \cdot M \quad (9)$$

During training, $F_{t \rightarrow s}, F_{t \rightarrow s}^{rigid}, M$ and the input images of SD-OFNet are randomly cropped to 416×128 before computing \mathcal{L} : in doing so, the errors at occluded areas in $F_{t \rightarrow s}$ due to camera motions, clearly visible in Figure 5, are less to appear and impact the training process. Finally, to ameliorate the photometric loss term, the image \tilde{I}_t^{SF} is obtained by padding the $SF_{t \rightarrow s}$ at first, which is predicted at 416×128 , to original resolution (*e.g.*, 640×192), then using this flow to warp the full resolution I_s at I_t coordinates and finally extracting from this image the same crop as used before. This simple strategy allows to leverage on a complete image, since otherwise the cropped image would suffer from motion occlusions near boundaries. Moreover, we highlight that SD-OFNet is initialized to the OFNet weights, *i.e.* those found during the above described OFNet training based on the standard photometric loss, and then, when training SD-OFNet, only its weights are updated, *i.e.* OFNet is kept frozen.

3. Monocular Depth Estimation

In this section, we provide more insights on Ω Net performance concerning depth estimation, in particular by reporting comparison with state-of-the-art methods trained with *stronger* supervision, a more detailed analysis about the errors computed at different depth ranges and a reproducibility study about DSNet.

3.1. Comparison with more methods on the KITTI Eigen split

In this section, we report additional comparisons on the Eigen’s KITTI test split [11]. In particular, we compare Ω Net to state-of-the-art frameworks trained with stronger forms of supervision, *i.e.* stereo pairs, stereo videos or proxy labels. Differently from these approaches, we do not apply any post-processing step to further improve predictions. As highlighted in Table 3, we can notice how our method is comparable and, in most cases performs better, wrt other self-supervised depth-from-mono architectures trained on stereo pairs/stereo videos. Moreover, we point out that we outperform frameworks running online adaptation on the testing set [3, 7] on most metrics. Only semi-supervised methods at the bottom of the table [32, 27, 30] are in general more effective, because of the much stronger supervision from traditional stereo algorithms deployed during training.

3.2. Error at different depth ranges

In Table 3, we report more data supporting the claim that DSNet produces more accurate depth estimates at long distances with respect to other strategies such as [14] or even replacing our architecture with a much more complex one [34] based on a ResNet-50 backbone. We deeply looked into this and ascribe this finding to more complex models producing *over-smoothed* depth maps. In particular, in our experiments, we noticed that our shallow network tends to produce much sharper estimates compared to models having many more parameters. *Over-smoothing* produces better qualitative predictions and higher accuracy at short ranges, but it degrades depth accuracy at long distances, as we can observe in the table.

Method	M	S	V	P	A	I	CS	Lower is better				Higher is better		
								Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou <i>et al.</i> [37]			✓				✓	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Godard <i>et al.</i> [14]			✓			✓		0.115	0.903	4.863	0.193	0.877	0.959	0.981
Godard <i>et al.</i> [14]			✓					0.132	1.044	5.142	0.210	0.845	0.948	0.977
Godard <i>et al.</i> [14] (1024 × 320)			✓			✓		0.115	0.882	4.701	0.190	0.879	0.961	0.982
Zhou <i>et al.</i> [36]			✓			✓		0.121	0.837	4.945	0.197	0.853	0.955	0.982
Mahjourian <i>et al.</i> [19]			✓				✓	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Yang <i>et al.</i> [33]			✓				✓	0.159	1.345	6.254	0.247	-	-	-
Wang <i>et al.</i> [28]			✓				✓	0.151	1.257	5.583	0.228	0.810	0.936	0.974
Bian <i>et al.</i> [2]			✓					0.128	1.047	5.234	0.208	0.846	0.947	0.970
Yin <i>et al.</i> [34]	✓		✓				✓	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Zou <i>et al.</i> [38]	✓		✓				✓	0.146	1.182	5.215	0.213	0.818	0.943	0.978
Chen <i>et al.</i> [7]	✓		✓					0.135	1.070	5.230	0.210	0.841	0.948	0.980
Luo <i>et al.</i> [18]	✓		✓					0.141	1.029	5.350	0.216	0.816	0.941	0.976
Ranjan <i>et al.</i> [1]	✓		✓				✓	0.139	1.032	5.199	0.213	0.827	0.943	0.977
Xu <i>et al.</i> [31]			✓		✓			0.138	1.016	5.352	0.217	0.823	0.943	0.976
Casser <i>et al.</i> [3]			✓		✓			0.141	1.026	5.290	0.215	0.816	0.945	0.979
Gordon <i>et al.</i> [15]	✓		✓		✓			0.128	0.959	5.230	-	-	-	-
Ω Net(416 × 128)			✓		✓			0.134	0.893	5.137	0.208	0.829	0.946	0.979
Ω Net(640 × 192)			✓		✓			0.126	0.835	4.937	0.199	0.844	0.953	0.982
Ω Net(1024 × 320)			✓		✓			0.125	0.805	4.795	0.195	0.849	0.955	0.983
Ω Net(416 × 128)			✓		✓		✓	0.126	0.862	4.963	0.199	0.846	0.952	0.981
Ω Net(640 × 192)			✓		✓		✓	0.120	0.792	4.750	0.191	0.856	0.958	0.984
Ω Net(1024 × 320)			✓		✓		✓	0.118	0.748	4.608	0.186	0.865	0.961	0.985
Ω Net(768 × 384) †			✓		✓		✓	0.184	1.565	6.456	0.243	0.742	0.920	0.974
Casser <i>et al.</i> [3] (+ Online Ref.)			✓		✓			0.109	0.825	4.750	0.187	0.874	0.958	0.983
Chen <i>et al.</i> [7] (+ Online Ref.)			✓					0.099	0.796	4.743	0.186	0.884	0.955	0.979
Poggi <i>et al.</i> [22]		✓					✓	0.146	1.291	5.907	0.245	0.801	0.926	0.967
Poggi <i>et al.</i> [23]		✓					✓	0.111	0.849	4.822	0.202	0.865	0.952	0.978
Pillai <i>et al.</i> [21]		✓						0.112	0.875	4.958	0.207	0.852	0.947	0.977
Godard <i>et al.</i> [14]		✓	✓			✓		0.106	0.806	4.630	0.193	0.876	0.958	0.980
Godard <i>et al.</i> [13]		✓					✓	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Zhang <i>et al.</i> [35]		✓	✓					0.135	1.132	5.585	0.229	0.820	0.933	0.971
Luo <i>et al.</i> [18]	✓	✓						0.127	0.936	5.008	0.209	0.841	0.946	0.979
Yang <i>et al.</i> [32]		✓	✓	✓				0.097	0.734	4.442	0.187	0.888	0.958	0.980
Watson <i>et al.</i> [30]		✓		✓		✓		0.096	0.710	4.393	0.185	0.890	0.962	0.981
Tosi <i>et al.</i> [27]		✓		✓			✓	0.096	0.673	4.351	0.184	0.890	0.961	0.981

Table 2. Quantitative evaluation on the Eigen test set of the KITTI dataset [12] for self-supervised monocular depth estimation methodologies. S: stereo pairs, V: video sequence, P: depth proxy labels, A: additional information, I: feature extractors pre-trained on ImageNet [9] or CS: Cityscapes [8]. †Trained on CS and tested on KITTI without any fine-tuning.

Method	Cap (m)	Abs Rel	Sq Rel	RMSE	RMSE log
Godard <i>et al.</i> [14]	0-8	0.059	0.062	0.503	0.082
Ours †	0-8	0.060	0.063	0.502	0.082
Ours	0-8	0.062	0.065	0.517	0.085
Godard <i>et al.</i> [14]	0-15	0.083	0.173	1.178	0.125
Ours †	0-15	0.083	0.168	1.148	0.122
Ours	0-15	0.084	0.169	1.156	0.124
Godard <i>et al.</i> [14]	0-30	0.111	0.470	2.561	0.172
Ours †	0-30	0.111	0.442	2.513	0.169
Ours	0-30	0.111	0.425	2.463	0.169
Godard <i>et al.</i> [14]	0-50	0.125	0.788	3.946	0.198
Ours †	0-50	0.127	0.762	4.020	0.199
Ours	0-50	0.124	0.702	3.836	0.195
Godard <i>et al.</i> [14]	0-80	0.132	1.044	5.142	0.210
Ours †	0-80	0.134	1.074	5.451	0.213
Ours	0-80	0.126	0.835	4.937	0.199

Table 3. Depth errors at different depth ranges. † indicates that our depth network has been replaced with the heavy-weight [34] backbone based on the ResNet50 architecture.

3.3. Reproducibility

We perform three independent training of our architecture to assess upon its reproducibility. Table 4 shows how our architecture produces the same results with negligible variance due to the randomness factors in training, *i.e.* initialization, data shuffle and augmentation.

Resolution	Abs Rel	Sq Rel	Lower is better		Higher is better		
			RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
640 × 192	0.120	0.792	4.750	0.191	0.856	0.958	0.984
640 × 192	0.122	0.799	4.749	0.191	0.856	0.958	0.984
640 × 192	0.121	0.795	4.755	0.192	0.855	0.957	0.983

Table 4. Three independent runs of our Ω Net(DSNet) result in slightly different models on the KITTI Eigen split.

4. Semantic Segmentation

In this section we report more detailed semantic segmentation results. Purposely, we use the following metrics:

1. **IoU**: Intersection over Union for pixel-wise segmentation calculated for each class or category, as defined in [8].
2. **mIoU_{class}**: mean IoU for the the 19 training classes used in CityScapes [8]: road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle and bicycle.
3. **mIoU_{category}**, mean IoU considering the 7 macro-classes defined in CityScapes [8]: flat, construction, object, nature, sky, human, vehicles.
4. **Pixel Accuracy (Acc.)**: ratio between the correct and the total pixel predictions without considering any specific class or category.

4.1. Generalization across Datasets

In Tables 5 and 6 we validate with thorough experiments the motivation behind our better generalization across datasets compared to other state-of-the-art methods for real-time semantic segmentation. In this study we train on CityScapes and test on KITTI, reporting in Table 5 the IoU for the 19 classes, the mIoU_{class} and the pixel Acc. In Table 6 we report the IoU for the 7 categories and the mIoU_{category}. We refer with CS(S) methods trained on 2975 CityScapes images and with CS(P) methods trained on 22,973 proxy labels produced by [6]. To evaluate the performance of [16, 4] we used the official code and pre-trained weights available online. Our DSNet differs from other methods by three factors: 1) the architecture, 2) the training protocol exploiting proxy labels instead of ground truths and 3) the joint reasoning about geometry and semantics.

Regarding the tests on KITTI, our architecture trained only for semantic segmentation, namely Semantic Network or SNet, achieves good performance in Acc. but turns out worse than [4] for other metrics. On the other hand, it is worth to notice that training SNet with CS(P) allows our method to achieve a great performance boost in all metrics with respect to CS(P) (rows 8 vs 9). Finally, we can notice how DSNet achieves results comparable to SNet. This confirms the findings in [24], that joint reasoning about depth and semantics is more beneficial to the former task.

Method	Train	Test	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU _{class}	Acc.
DABNet [16]	CS(S)	CS	97.05	82.86	91.01	48.20	55.56	59.30	63.12	72.76	91.58	61.24	93.50	77.96	54.70	93.28	53.06	71.01	27.77	56.00	72.91	69.62	94.62
FCHardNet [4]	CS(S)	CS	97.39	84.40	92.31	53.83	62.90	64.28	68.21	78.06	91.85	59.82	94.91	80.81	60.55	94.85	72.70	82.15	76.45	59.97	75.49	76.37	95.35
ΩNet(SNet)	CS(S)	CS	93.69	65.66	83.46	23.57	20.57	40.11	35.32	47.77	86.62	44.22	89.94	56.00	23.00	84.98	17.22	1.22	0.00	17.11	52.82	46.49	89.56
ΩNet(SNet)	CS(P)	CS	95.97	77.23	87.96	38.37	42.62	47.82	48.15	60.44	89.73	54.97	92.62	65.87	36.96	90.57	25.19	0.06	0.00	25.53	61.06	54.80	92.45
ΩNet(DSNet)	CS(P)	CS	96.00	77.46	88.30	41.84	41.68	48.74	47.80	59.24	89.61	53.89	92.57	66.29	38.61	90.61	27.39	0.37	0.00	18.01	62.78	54.80	92.50
DABNet [16]	CS(S)	K	79.02	19.07	58.38	18.04	30.73	40.61	44.24	41.67	80.87	48.76	76.61	13.39	0.17	63.30	21.32	8.21	19.81	1.29	7.04	35.40	80.50
FCHardNet [4]	CS(S)	K	75.66	32.65	78.51	13.16	28.46	51.33	57.16	55.58	81.06	45.59	91.43	23.84	12.19	58.86	24.91	34.89	68.71	4.66	11.38	44.74	72.07
ΩNet(SNet)	CS(S)	K	83.31	33.39	66.57	12.15	20.18	44.20	37.76	32.35	84.46	58.79	88.70	24.66	13.55	76.09	12.62	2.09	0.10	1.15	12.64	37.09	84.94
ΩNet(SNet)	CS(P)	K	88.73	47.85	77.01	19.72	30.65	47.34	53.63	43.16	86.65	67.97	94.49	24.81	29.39	80.68	14.88	0.53	0.00	3.05	12.30	43.31	88.76
ΩNet(DSNet)	CS(P)	K	87.89	46.64	77.48	18.55	29.65	48.73	51.12	40.52	86.66	63.54	95.06	29.79	34.74	82.03	12.77	0.63	0.00	7.60	18.82	43.80	88.31

Table 5. IoU on 19 training classes, mIoU_{class} and pixel accuracy (Acc.) results of ΩNet against state of the art method training on CS and tested on CS or K. Better generalization from CS to K thanks to our proxy labels training protocol.

Method	Train	Test	flat	construction	object	nature	sky	human	vehicle	mIoU _{category}
DABNet [16]	CS(S)	CS	97.93	91.69	65.90	92.03	93.50	79.59	92.25	87.56
FCHardNet [4]	CS(S)	CS	98.19	92.55	70.77	92.27	94.91	82.31	93.54	89.22
ΩNet(SNet)	CS(S)	CS	96.34	84.29	44.37	86.85	89.94	60.13	83.77	77.96
ΩNet(SNet)	CS(P)	CS	97.40	88.80	53.61	90.19	92.62	69.08	88.47	82.88
ΩNet(DSNet)	CS(P)	CS	97.38	88.76	53.91	89.93	92.57	69.27	88.61	82.92
DABNet [16]	CS(S)	K	83.41	59.07	46.41	84.30	76.61	17.05	63.61	61.49
FCHardNet [4]	CS(S)	K	80.89	75.35	58.68	88.11	91.43	24.62	58.33	68.20
ΩNet(SNet)	CS(S)	K	87.93	63.92	45.79	85.47	88.70	31.02	69.95	67.54
ΩNet(SNet)	CS(P)	K	91.97	74.95	52.29	89.80	94.49	29.28	81.83	73.52
ΩNet(DSNet)	CS(P)	K	91.42	74.84	53.35	89.36	95.06	35.45	80.69	74.31

Table 6. IoU on 7 training categories and, mIoU_{category} results of ΩNet against state of the art method training on CS and tested on CS or K. Better generalization from CS to K thanks to our proxy labels training protocol.

4.2. Proxy Semantic Network

We evaluate the performance of the proxy semantic network. We employ DPC [6], pre-trained on CityScapes with the 2975 training ground truths. We report in Table 7 the testing results on the 500 and 200 images belonging to CityScapes validation set and the KITTI training datasets, respectively. Even though DPC [6] achieves impressive performance both on CityScapes as well as in generalizing to KITTI, it is a huge network unable to run in real-time (*i.e.*, it approximately delivers 3.5 fps on 768×384 images).

Method	Train	Test	mIoU _{class}	mIoU _{category}	Acc.
DPC[5] - Proxy	CS(S)	CS	80.22	90.73	95.99
DPC[5] - Proxy	CS(S)	K	58.75	81.30	90.21

Table 7. Semantic segmentation performances of the proxy semantic network [6] on CS and K datasets.

4.3. Priors Evaluation on KITTI

When we produce the priors used during training and, at prediction time, to create the M_t^{mot} , we split the 19 classes in static and potentially dynamic ones according to the following scheme:

1. **Static:** road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky
2. **Potentially dynamic:** person, rider, car, truck, bus, train

As among our objectives is to obtain a good motion segmentation mask, we are interested in evaluating the quality of our semantic segmentation predictions in terms of how they are amenable to producing good estimated priors according to the mapping defined above. We evaluate our DSNet trained on CityScapes+KITTI in the 200 KITTI images which provides semantic labels. We obtain a pixel accuracy of 98.50% while a 98.40% IoU for the static classes and a 80.99% for the potentially dynamic classes for a global 89.64% mIoU. It is worth noticing that, even though our segmentation is not able to perform a precise class segmentation, it yields excellent binary priors that turns out key to performance for motion segmentation.

5. Optical Flow Estimation

5.1. Comparison with more methods on the KITTI 2015 split

In Table 8 we include additional results from our main competitors to allow for a more comprehensive analysis. In particular, we report additional experiments from [1], in which the authors exploit a different combination of depth and optical flow networks, and from [29], that demonstrate how using stereo pairs at training time allows to obtain much better results on rigid regions. Nonetheless, it can be noticed that our network still outperforms existing monocular multi-task strategies by a large margin.

Method	Dataset	train			test
		Noc	All	F1	F1
Meister <i>et al.</i> [20] - C	SYN + K	-	8.80	28.94%	29.46%
Meister <i>et al.</i> [20] - CSS	SYN + K	-	8.10	23.27%	23.30%
Zou <i>et al.</i> [38]	SYN + K	-	8.98	26.0%	25.70%
Ranjan <i>et al.</i> [1] - DispResNet + PWC	SYN + K	-	5.66	20.93%	25.27%
Wang <i>et al.</i> [29] (Ego-motion) **	K	-	10.69	-	32.34%
Wang <i>et al.</i> [29] (Full) **	K	-	5.58	-	18.00%
Ren <i>et al.</i> [25]	K	-	16.79	36.00%	39.00%
Yin <i>et al.</i> [34]	K	8.05	10.81	-	-
Chen <i>et al.</i> [7] †	K	5.40	8.95	-	-
Chen <i>et al.</i> [7] (online) †	K	4.86	8.35	-	-
Ranjan <i>et al.</i> [1] - DispNet + FlowNetC	K	-	7.76	-	-
Ranjan <i>et al.</i> [1] - DispResNet + PWC	K	-	6.21	26.41%	-
Luo <i>et al.</i> [18]	K	5.84	-	21.56%	-
Luo <i>et al.</i> [18] *	K	5.43	-	20.61%	-
Ω Net (Ego-motion)	K	11.72	13.50	51.22%	-
Ω Net(OFFNet)	K	3.48	11.61	25.78%	-
Ω Net(SD-OFFNet)	K	3.29	5.39	20.00%	19.47%

Table 8. We report percentage of erroneous pixels (F1 score) and average end-point error over all pixels (All) and non-occluded pixels (Noc) on the KITTI 2015 flow dataset. We indicate with †feature extractors pre-trained on ImageNet, SYN as the SYNTHIA [26] dataset, CS for the Cityscapes dataset, multi-task methods *trained on stereo pair and ** using stereo at testing time.

6. Pose Estimation

We validate the performance of our framework on pose estimation on the KITTI odometry split, which provides ground-truth camera poses obtained with IMU/GPS readings for 11 driving sequences, indexed from 00 to 08 for training and 09-10 for testing purposes. As in [14], we have not changed our architecture for this specific task but simply trained it from scratch on new training sequences without known intrinsic parameters. We compare our model with learned camera intrinsic parameters with several monocular self-supervised methods on the two sequences of KITTI odometry test split. All of the results, summarized in 9, are evaluated by optimizing the scaling factor to align with the ground-truth to address the inherent scale ambiguity.

Method	Frames	Sequence 09	Sequence 10
Zhou <i>et al.</i> [37]	5	0.021 ± 0.017	0.020 ± 0.015
Ranjan <i>et al.</i> [1]	5	0.012 ± 0.007	0.012 ± 0.008
Yin <i>et al.</i> [34]	5	0.012 ± 0.007	0.012 ± 0.009
ORB-Slam	3	0.014 ± 0.008	0.012 ± 0.011
Casser <i>et al.</i> [3]	3	0.011 ± 0.006	0.011 ± 0.010
Zou <i>et al.</i> [3]	3	0.017 ± 0.007	0.015 ± 0.009
Luo <i>et al.</i> [18]	3	0.013 ± 0.007	0.012 ± 0.008
Godard <i>et al.</i> [14]	2	0.017 ± 0.008	0.015 ± 0.010
Ours †	2	0.020 ± 0.013	0.017 ± 0.011

Table 9. Absolute Trajectory Error (ATE) of pose estimation evaluated on the KITTI odometry split sequences 09-10. †indicates strategies trained with unknown camera intrinsics.

7. Motion Segmentation

7.1. Threshold analysis

In Figure 1, we present an ablation study dealing with the motion segmentation task. In the main paper, to be consistent with other methodologies, we set the threshold τ used for the evaluation to 0.5. However, we point out that a careful tuning of such threshold can improve the overall motion segmentation accuracy. In particular, we can notice how the best configuration for our predictions is obtained using a larger threshold. Indeed, we found out that the best trade-off between the mean accuracy and the mean IoU is achieved by setting the threshold value to 0.7 (in this case the Mean Acc is 0.91 while Mean IoU is 0.77).

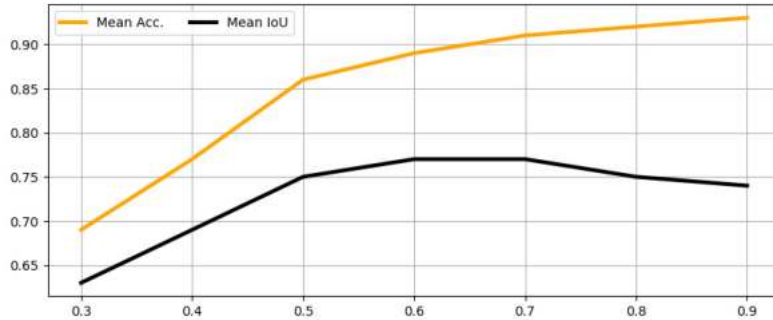


Figure 1. Mean Acc. and mIoU varying the threshold used to compute the motion segmentation M_t^{mot} .

7.2. Evaluation for KITTI only on Cars

We conduct an additional study to evaluate our motion segmentation masks only on pixels belonging to Cars, as proposed in [1]. In Table 10 we evaluate the IoU for static and dynamic cars yielded by Ω Net and [1] on the 200 KITTI images endowed with ground truth for the motion segmentation task. We notice that our M_t^{mot} outperforms [1] in all metrics (rows 1 vs 2 and 3) for all thresholds. Moreover, we point out that in this test configuration the contribution given to the motion segmentation by our estimated semantics is almost negligible as car regions are already extracted by using KITTI ground truths. Therefore, we test also our motion probability P_t alone, showing that it is superior to [1] even without the help provided by the estimated semantics.

Method	Threshold	Overall	Static Cars	Moving Cars
Ranjan [1]	-	56.94	55.77	58.11
Ω Net M_t^{mot}	0.5	63.98	64.16	63.79
Ω Net M_t^{mot}	0.7	63.97	64.15	63.79
Ω Net P_t	0.5	63.67	62.58	64.77
Ω Net P_t	0.7	62.66	58.42	66.89

Table 10. Motion Segmentation Results. IoU scores on KITTI 2015 training dataset images computed only over car pixels.

8. Runtime

In this section we report additional runtime results on the three different devices used in the main paper, that is: an NVIDIA Titan Xp GPU, an Intel i7-7700K CPU and an NVIDIA Jetson TX2 GPU. In Table 11, we show further timings by varying the input image resolution of our architecture. It can be noticed how Ω Net achieves real-time results (*i.e.* 27.9) on the Titan Xp GPU even with the largest image size 1024×320 , reaching about 2 FPS on the Jetson Tx2 embedded device with the same input configuration.

	416 × 128						640 × 192					1024 × 320				
	W	D	DS	OF	Cam	O	D	DS	OF	Cam	O	D	DS	OF	Cam	O
Jetson TX2	15	20.2	17.9	8.9	54.1	7.1	12.5	10.3	6.5	49.2	4.5	6.4	5.3	3.2	26.31	2.0
i7-7700K	91	10.9	9.1	11.0	60.1	5.5	5.0	4.2	4.9	31.4	2.4	1.9	1.6	1.8	13.2	0.9
Titan XP	250	250.7	212.4	152.6	550.7	90.5	170.2	134.1	94.1	446.7	57.4	86.0	64.5	44.5	251.0	27.9

Table 11. Runtime analysis on different hardware devices. For each device we report the power consumption in Watt and the FPS by varying input resolution. D: Depth, S: Semantic, OF: Optical Flow, Cam: camera pose, O: Overall architecture.

9. Qualitative results

In Figures 2, 3, 4, 5, 6, 7, 8, we provide qualitative results of our architecture on the standard datasets used in the main paper, such as KITTI and CityScapes. We refer the reader to the captions for description and comments related to each example.

9.1. Results on a YouTube Video

Furthermore, to prove that our network can be trained on unconstrained monocular sequences with unknown camera parameters and without semantic ground-truth labels, we downloaded from YouTube an online video captured by a moving camera consisting of 130K images depicting an urban scenario. Then, we generated proxy semantic labels using [6] and trained Ω Net(DSNet) to learn depth, pose, semantics and camera intrinsics. Figure 9, show qualitative results yielded by Ω Net on this unconstrained monocular video.

References

- [1] Ranjan Anurag, Varun Jampani, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4, 7, 8
- [2] Jia-Wang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *Thirty-third Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 4
- [3] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019. 3, 4, 7
- [4] Ping Chao, Chao-Yang Kao, Yu-Shan Ruan, Chien-Hsiang Huang, and Youn-Long Lin. Hardnet: A low memory traffic network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3552–3561, 2019. 5, 6
- [5] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *Advances in Neural Information Processing Systems*, pages 8699–8710, 2018. 6
- [6] Liang-Chieh Chen, Maxwell D. Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jonathon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *NIPS*, 2018. 1, 5, 6, 9
- [7] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *ICCV*, 2019. 3, 4, 7
- [8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4, 5
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 4
- [10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 12
- [11] 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. 3
- [12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 4
- [13] Clément Godard, Oisín Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1, 4
- [14] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 1, 2, 3, 4, 5, 7
- [15] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *ICCV*, 2019. 4

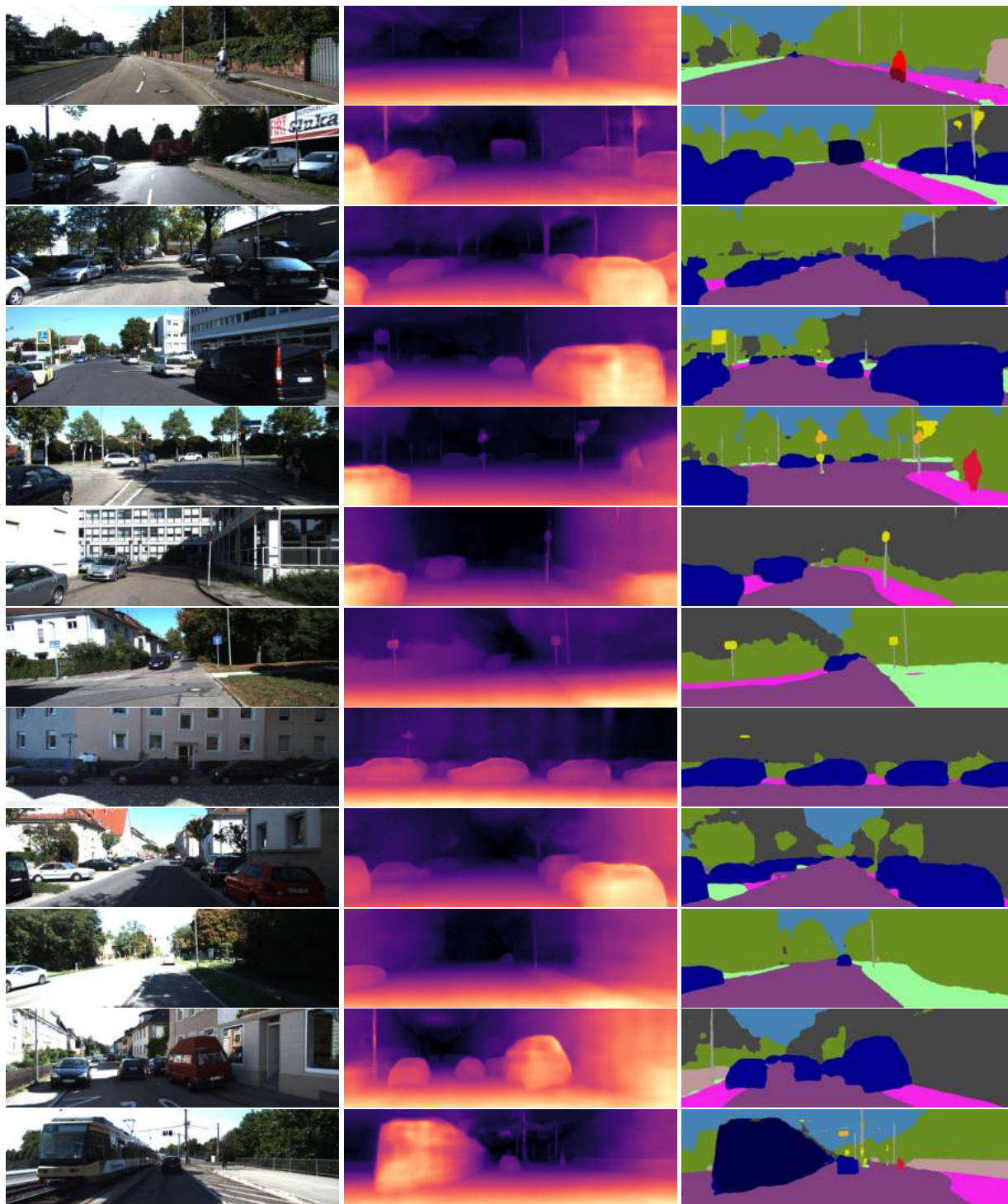


Figure 2. Qualitative results of our Ω Net(DSNet) on the KITTI Eigen split. From left to right we show image, the predicted single-image depth map and the predicted semantic segmentation of the scene.

[16] Gen Li and Joongkyu Kim. Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. In *British Machine Vision Conference*, 2019. 5, 6

[17] Pengpeng Liu, Michael R. Lyu, Irwin King, and Jia Xu. Selfflow: Self-supervised learning of optical flow. In *CVPR*, 2019. 1

[18] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning

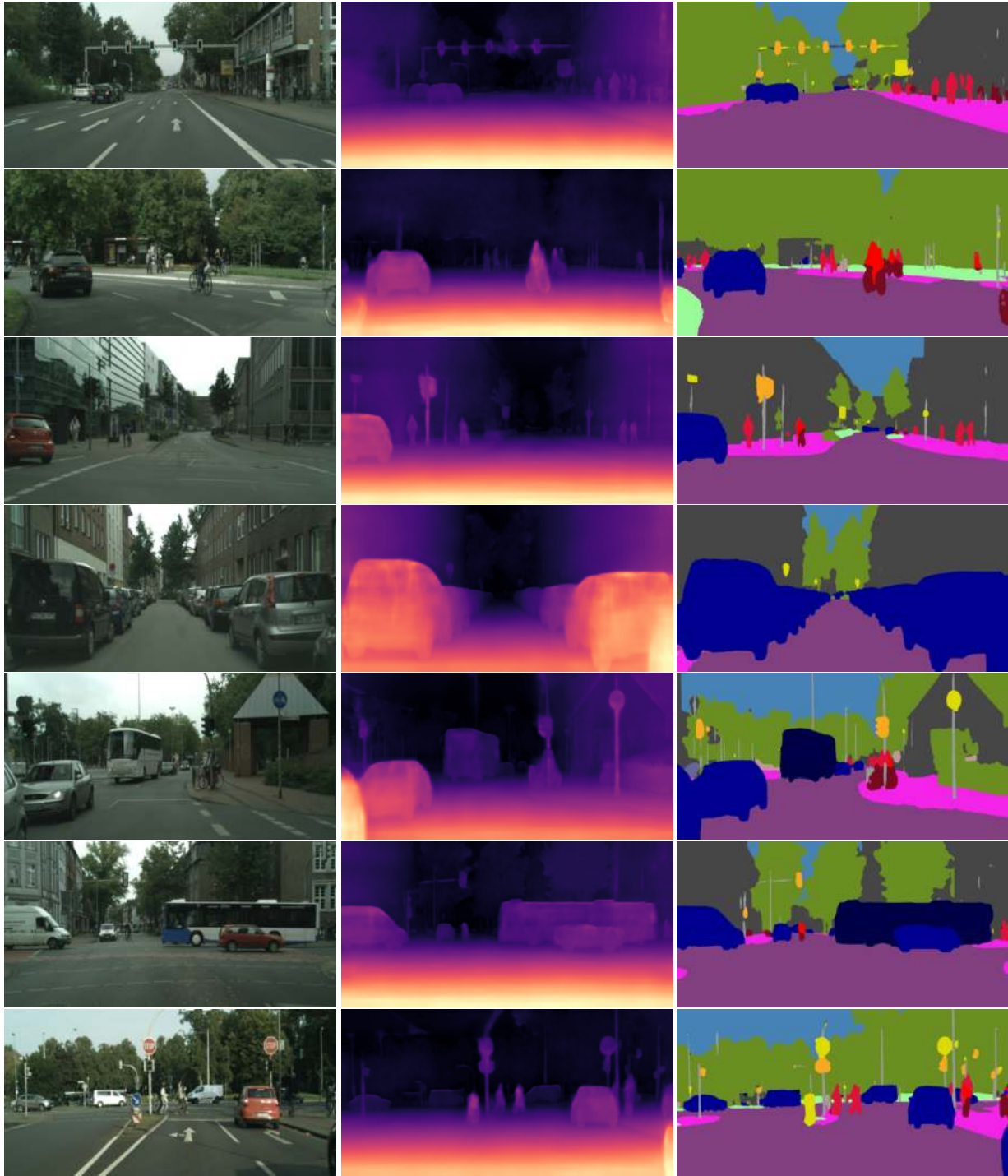


Figure 3. Qualitative results of our network Ω Net(DSNet) on the Cityscapes dataset. From left to right, the input image, single-view depth and the semantic prediction of our model.

of geometry and motion with 3d holistic understanding. *PAMI*, 2019. 4, 7

- [19] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4
- [20] Simon Meister, Junhwa Hur, and Stefan Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, New Orleans, Louisiana, Feb. 2018. 7, 12

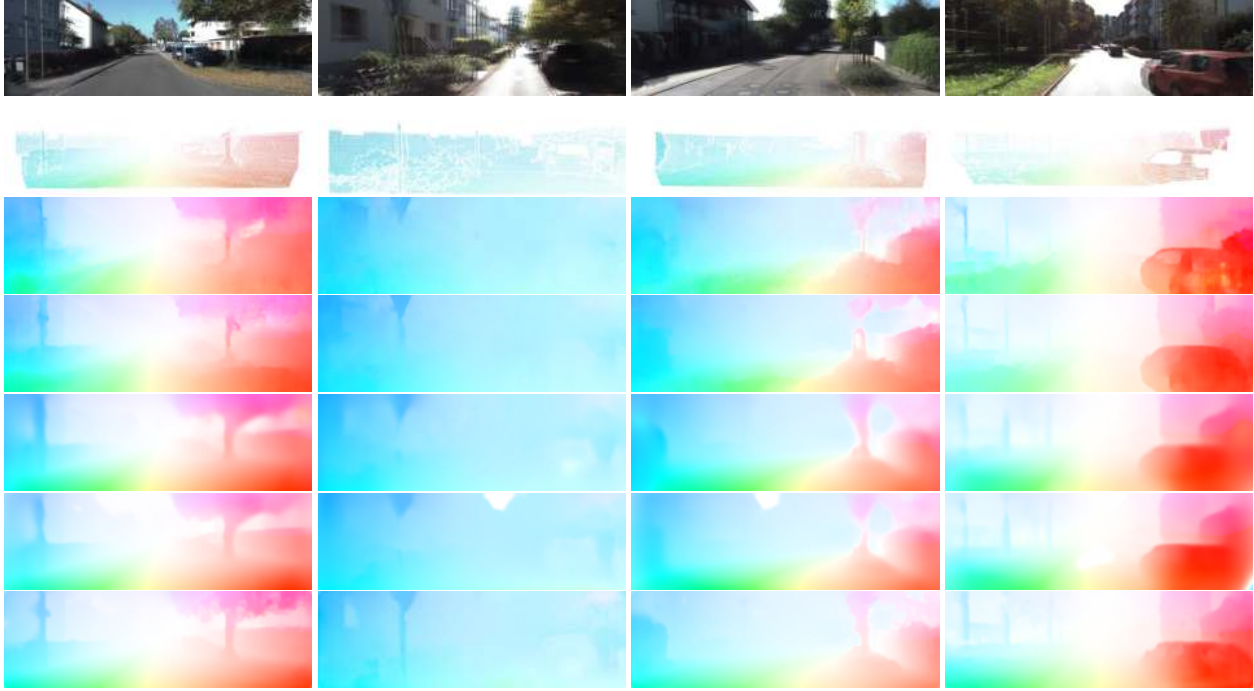


Figure 4. Qualitative results on optical flow estimation using the KITTI 2015 dataset. From top to bottom we show image, ground-truth labels, FlowNetS [10], FlowNetC [10], Unflow [20], DF-Net [38] and our Ω Net(SD-OFNet).



Figure 5. Qualitative comparison between the initial optical flow network Ω Net(OFNet) and the self-distilled one Ω Net(SD-OFNet) obtained with our strategy on the KITTI 2015 dataset. From left to right we show image, Ω Net(OFNet) and Ω Net(SD-OFNet) results respectively. It can be noticed that the proposed self-distilled paradigm greatly alleviates motion boundaries occlusions and improves details in the final optical flow predictions.



Figure 6. Visualization of the flow error map of our optical flow network on the KITTI 2015 testing benchmark. Larger errors are encoded in red, while blue pixels represents good optical flow estimates with respect to the ground-truth.



Figure 7. Qualitatives on motion segmentation. From left to right, the input images, the motion objects detected in the scene by our method (highlighted in red), the optical flow and the rigid flow.

- [21] Sudeep Pillai, Rares Ambrus, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *ICRA*, 2019. 4
- [22] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. Towards real-time unsupervised monocular depth estimation on CPU. In *IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2018. 4
- [23] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *6th International Conference on 3D Vision (3DV)*, 2018. 4
- [24] Pierluigi Zama Ramirez, Matteo Poggi, Fabio Tosi, Stefano Mattoccia, and Luigi Di Stefano. Geometry meets semantics for semi-supervised monocular depth estimation. In *Asian Conference on Computer Vision*, pages 298–313. Springer, 2018. 2, 5
- [25] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Bin, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017. 7
- [26] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 7

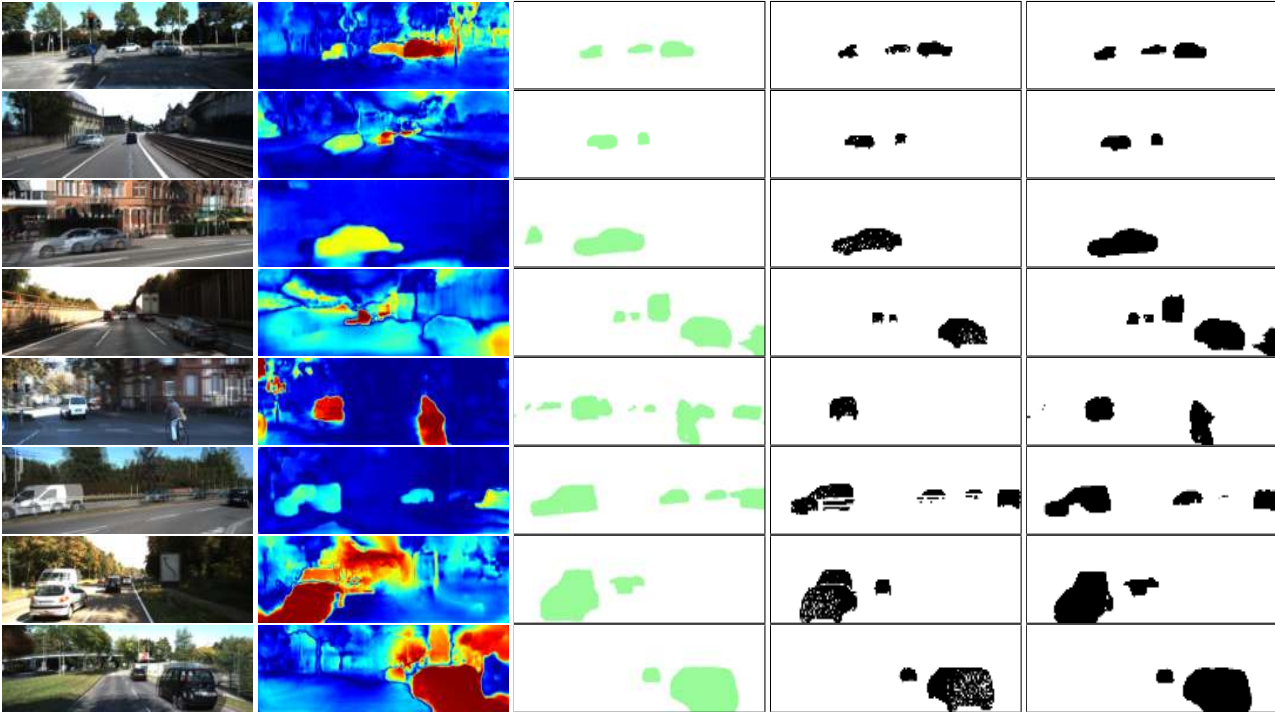


Figure 8. Motion segmentation results on the KITTI 2015 dataset. From left to right, we show the monocular sequence, the outcome of the proposed motion probability strategy (high probability of motion is encoded in red), semantic priors extracted from our semantic predictions, ground-truth motion masks and ours estimated motion segmentation masks.

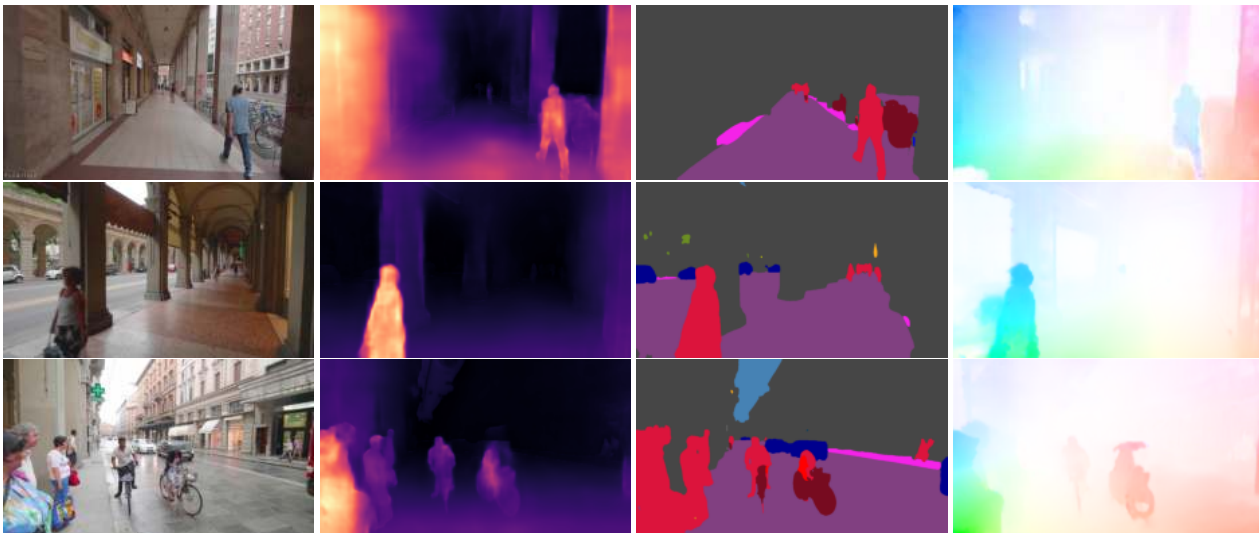


Figure 9. Qualitative results of Ω Net on a raw YouTube video. From left to right, we show the input images of a monocular sequence, the single-view depth and semantic predictions and, finally, the optical flow estimate.

- [27] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 4
- [28] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4
- [29] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8071–8081, 2019. 7

- [30] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *ICCV*, 2019. 3, 4
- [31] Haofei Xu, Jianmin Zheng, Jianfei Cai, and Juyong Zhang. Region deformer networks for unsupervised depth estimation from unconstrained monocular videos. In *IJCAI*, 2019. 4
- [32] Nan Yang, Rui Wang, Jörg Stückler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *European Conference on Computer Vision*, pages 835–852. Springer, 2018. 3, 4
- [33] Zhenheng Yang, Peng Wang, Wang Yang, Wei Xu, and Nevatia Ram. Lego: Learning edge with geometry all at once by watching videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4
- [34] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 4, 5, 7
- [35] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4
- [36] Junsheng Zhou, Yuwang Wang, Naiyan Wang, and Wenjun Zeng. Unsupervised high-resolution depth learning from videos with dual networks. In *Inter. Conf. on Computer Vision*. IEEE, IEEE, October 2019. 4
- [37] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 4, 7
- [38] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision (ECCV)*, 2018. 4, 7, 12