The Cityscapes Dataset for Semantic Urban Scene Understanding

– SUPPLEMENTAL MATERIAL –

Marius Cordts[1,2] Mohamed Omran[3] Sebastian Ramos[1,4] Timo Rehfeld[1,2]
Markus Enzweiler[1] Rodrigo Benenson[3] Uwe Franke[1] Stefan Roth[2] Bernt Schiele[3]

[1]Daimler AG R&D, [2]TU Darmstadt, [3]MPI Informatics, [4]TU Dresden
www.cityscapes-dataset.net

A. Related Datasets

In Tab. 7 we provide a comparison to other related datasets in terms of the type of annotations, the meta information provided, the camera perspective, the type of scenes, and their size. The selected datasets are either of large scale or focus on street scenes.

B. Class Definitions

Table 8 provides precise definitions of our annotated classes. These definitions were used to guide our labeling process, as well as quality control. In addition, we include a typical example for each class.

The annotators were instructed to make use of the depth ordering and occlusions of the scene to accelerate labeling, analogously to LabelMe [60]; see Fig. 6 for an example. In doing so, distant objects are annotated first, while occluded parts are annotated with a coarser, conservative boundary (possibly larger than the actual object). Subsequently, the occluder is annotated with a polygon that lies in front of the occluded part. Thus, the boundary between these objects is shared and consistent.

Holes in an object through which a background region can be seen are considered to be part of the object. This allows keeping the labeling effort within reasonable bounds such that objects can be described via simple polygons forming simply-connected sets.

C. Example Annotations

Figure 7 presents several examples of annotated frames from our dataset that exemplify its diversity and difficulty. All examples are taken from the *train* and *val* splits and were chosen by searching for the extremes in terms of the number of traffic participant instances in the scene; see Fig. 7 for details.

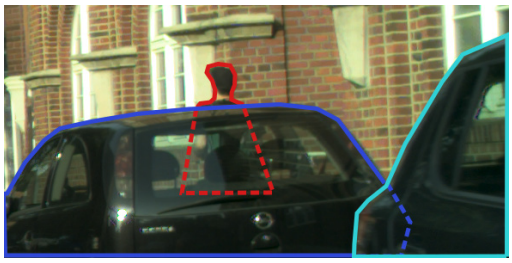

Figure 6. Exemplary labeling process. Distant objects are annotated first and subsequently their occluders. This ensures the boundary between these objects to be shared and consistent.

D. Detailed Results

In this section, we present additional details regarding our control experiments and baselines. Specifically, we give individual class scores that complement the aggregated scores in the main paper. Moreover, we provide details on the training procedure for all baselines. Finally, we show additional qualitative results of all methods.

D.1. Semantic labeling

Tables 9 and 11 list all individual class-level IoU scores for all control experiments and baselines. Tables 10 and 12 give the corresponding instance-normalized iIoU scores. In addition, Figs. 8 and 9 contain qualitative examples of these methods.

Basic setup. All baselines relied on single frame, monocular LDR images and were pretrained on ImageNet [59], *i.e.* their underlying CNN was generally initialized with ImageNet VGG weights [68]. Subsequently, the CNNs were finetuned on Cityscapes using the respective portions listed in Tab. 4. In our own FCN [41] experiments, we additionally investigated first pretraining on PASCAL-Context [45], but found this to not influence performance given a sufficiently large number of training iterations. Most baselines applied a subsampling of the input image, *c.f.* Tab. 4, proba-

Dataset	Labels	Color	Video	Depth	Camera	Scene	#images	#classes
[59]	B	✓	×	×	Mixed	Mixed	150 k	1000
[14]	B, C	✓	×	×	Mixed	Mixed	20 k (B), 10 k (C)	20
[45]	D	✓	×	×	Mixed	Mixed	20 k	400
[38]	C	✓	×	×	Mixed	Mixed	300 k	80
[69]	D, C	✓	×	Kinect	Pedestrian	Indoor	10 k	37
[19]	B, D[a]	✓	✓	Laser, Stereo	Car	Suburban	15 k (B), 700 (D)	3 (B), 8 (D)
[7]	D	✓	✓	×	Car	Urban	701	32
[35]	D	✓	✓	Stereo, Manual	Car	Urban	70	7
[61]	D	×	✓	Stereo	Car	Urban	500	5
[2]	D	✓	×	×	Pedestrian	Urban	200	2
[65]	C	✓	×	Stereo	Car	Facades	86	13
[56]	D	✓	×	3D mesh	Pedestrian	Urban	428	8
[75]	D	✓	✓	Laser	Car	Suburban	400 k	27
Ours	D, C	✓	✓	Stereo	Car	Urban	5 k (D), 20 k (C)	30

[a] Including the annotations of 3rd party groups [22, 29, 32, 33, 58, 64, 77, 80]

Table 7. Comparison to related datasets. We list the type of labels provided, *i.e*. object bounding boxes (B), dense pixel-level semantic labels (D), coarse labels (C) that do not aim to label the whole image. Further, we mark if color, video, and depth information are available. We list the camera perspective, the scene type, the number of images, and the number of semantic classes.

bly due to time or memory constraints. Only Adelaide [37], Dilated10 [79], and our FCN experiments were conducted on the full-resolution images. In the first case, a new random patch of size 614×614 pixels was drawn at each iteration. In our FCN training, we split each image into two halves (left and right) with an overlap that is sufficiently large considering the network's receptive field.

Own baselines. The training procedure of all our FCN experiments follows [41]. We use three-stage training with subsequently smaller strides, *i.e*. first FCN-32s, then FCN-16s, and then FCN-8s, always initializing with the parameters from the previous stage. We add a 4th stage for which we reduce the learning rate by a factor of 10. The training parameters are identical to those publicly available for training on PASCAL-Context [45], except that we reduce the learning rate to account for the increased image resolution. Each stage is trained until convergence on the validation set; pixels with *void* ground truth are ignored such that they do not induce any gradient. Eventually, we retrain on *train* and *val* together with the same number of epochs, yielding 243 250, 69 500, 62 550, and 5950 iterations for stages 1 through 4. Note that each iteration corresponds to half of an image (see above). For the variant with factor 2 downsampling, no image splitting is necessary, yielding 80 325, 68 425, 35 700, and 5950 iterations in the respective stages. The variant only trained on *val* (full resolution) uses *train* for validation, leading to 130 000, 35 700, 47 600, and 0 iterations in the 4 stages. Our last FCN variant is trained using the coarse annotations only, with 386 750, 113 050, 35 700, and 0 iterations in the respective stage; pixels with *void* ground truth are ignored here as well.

3rd-party baselines. *Note that for the following descriptions of the 3rd-party baselines, we have to rely on author-provided information.*

SegNet [4] training for both the *basic* and *extended* variant was performed until convergence, yielding approximately 50 epochs. Inference takes 0.12 s per image.

DPN [40] was trained using the original procedure, while using all available Cityscapes annotations.

For training *CRF as RNN* [81], an FCN-32s model was trained for 3 days on *train* using a GPU. Subsequently an FCN-8s model was trained for 2 days, and eventually the model was further finetuned including the CRF-RNN layers. Testing takes 0.7 s on half-resolution images.

For training DeepLab on the fine annotations, denoted *DeepLab-LargeFOV-Strong*, the authors applied the training procedure from [9]. The model was trained on *train* for 40 000 iterations until convergence on *val*. Then *val* was included in the training set for another 40 000 iterations. In both cases, a mini-batch size of 10 was applied. Each training iteration lasts 0.5 s, while inference including the dense CRF takes 4 s per image. The DeepLab variant including our coarse annotations, termed *DeepLab-LargeFOV-StrongWeak*, followed the protocol in [48] and is initialized from the *DeepLab-LargeFOV-Strong* model. Each mini-batch consists of 5 finely and 5 coarsely annotated images and training is performed for 20 000 iterations until convergence on *val*. Then, training was continued for another 20 000 iterations on *train* and *val*.

Adelaide [37] was trained for 8 days using random crops of the input image as described above. Inference on a single image takes 35 s.

The best performing baseline, Dilated10 [79], is a convolutional network that consists of a front-end prediction module and a context aggregation module. The front-end module is an adaptation of the VGG-16 network based on dilated convolutions. The context module uses dilated convolutions

to systematically expand the receptive field and aggregate contextual information. This module is derived from the "Basic" network, where each layer has $C = 19$ feature maps. The total number of layers in the context module is 10, hence the name Dilation10. The increased number of layers in the context module (10 for Cityscapes versus 8 for PASCAL VOC) is due to the higher input resolution. The complete Dilation10 model is a pure convolutional network: there is no CRF and no structured prediction. The Dilation10 network was trained in three stages. First, the front-end prediction module was trained for $40\,000$ iterations on randomly sampled crops of size 628×628, with learning rate 10^{-4}, momentum 0.99, and batch size 8. Second, the context module was trained for $24\,000$ iterations on whole (uncropped) images, with learning rate 10^{-4}, momentum 0.99, and batch size 100. Third, the complete model (front-end + context) was jointly trained for $60\,000$ iterations on halves of images (input size 1396×1396, including padding), with learning rate 10^{-5}, momentum 0.99, and batch size 1.

D.2. Instance-level semantic labeling

For our instance-level semantic labeling baselines and control experiments, we rely on Fast R-CNN [20] and proposal regions from either MCG (Multiscale Combinatorial Grouping [1]) or from the ground truth annotations.

We use the standard training and testing parameters for Fast R-CNN. Training starts with a model pre-trained on ImageNet [59]. We use a learning rate of 0.001 and stop when the validation error plateaus after $120\,000$ iterations.

At test time, one score per class is assigned to each object proposal. Subsequently, thresholding and non-maximum suppression is applied and either the bounding boxes, the original proposal regions or their convex hull are used to generate the predicted masks of each instance. Quantitative results of all classes can be found in Tables 13 to 16 and qualitative results in Fig. 12.

Category	Class	Definition	Examples
human	person[1]	All humans that would primarily rely on their legs to move if necessary. Consequently, this label includes people who are standing/sitting, or otherwise stationary. This class also includes babies, people pushing a bicycle, or standing next to it with both legs on the same side of the bicycle.	
	rider[1]	Humans relying on some device for movement. This includes drivers, passengers, or riders of bicycles, motorcycles, scooters, skateboards, horses, Segways, (inline) skates, wheelchairs, road cleaning cars, or convertibles. Note that a visible driver of a closed car can only be seen through the window. Since holes are considered part of the surrounding object, the human is included in the *car* label.	
vehicle	car[1]	This includes cars, jeeps, SUVs, vans with a continuous body shape (*i.e.* the driver's cabin and cargo compartment are one). Does not include trailers, which have their own separate class.	
	truck[1]	This includes trucks, vans with a body that is separate from the driver's cabin, pickup trucks, as well as their trailers.	
	bus[1]	This includes buses that are intended for 9+ persons for public or long-distance transport.	
	train[1]	All vehicles that move on rails, *e.g.* trams, trains.	

[1] Single instance annotation available.
[2] Not included in challenges.

Table 8. List of annotated classes including their definition and typical example.

Category	Class	Definition	Examples
vehicle	motorcycle[1]	This includes motorcycles, mopeds, and scooters without the driver or other passengers. The latter receive the label *rider*.	
	bicycle[1]	This includes bicycles without the cyclist or other passengers. The latter receive the label *rider*.	
	caravan[1,2]	Vehicles that (appear to) contain living quarters. This also includes trailers that are used for living and has priority over the *trailer* class.	
	trailer[1,2]	Includes trailers that can be attached to any vehicle, but excludes trailers attached to trucks. The latter are included in the *truck* label.	
nature	vegetation	Trees, hedges, and all kinds of vertically growing vegetation. Plants attached to buildings/walls/fences are not annotated separately, and receive the same label as the surface they are supported by.	
	terrain	Grass, all kinds of horizontally spreading vegetation, soil, or sand. These are areas that are not meant to be driven on. This label may also include a possibly adjacent curb. Single grass stalks or very small patches of grass are not annotated separately and thus are assigned to the label of the region they are growing on.	

[1] Single instance annotation available.
[2] Not included in challenges.

Table 8. List of annotated classes including their definition and typical example. (continued)

Category	Class	Definition	Examples
construction	building	Includes structures that house/shelter humans, *e.g.* low-rises, skyscrapers, bus stops, car ports. Translucent buildings made of glass still receive the label *building*. Also includes scaffolding attached to buildings.	
	wall	Individually standing walls that separate two (or more) outdoor areas, and do not provide support for a building.	
	fence	Structures with holes that separate two (or more) outdoor areas, sometimes temporary.	
	guard rail[2]	Metal structure located on the side of the road to prevent serious accidents. Rare in inner cities, but occur sometimes in curves. Includes the bars holding the rails.	
	bridge[2]	Bridges (on which the ego-vehicle is not driving) including everything (fences, guard rails) permanently attached to them.	
	tunnel[2]	Tunnel walls and the (typically dark) space encased by the tunnel, but excluding vehicles.	

[1] Single instance annotation available.
[2] Not included in challenges.

Table 8. List of annotated classes including their definition and typical example. (continued)

Category	Class	Definition	Examples
object	traffic sign	Front part of signs installed by the state/city authority with the purpose of conveying information to drivers/cyclists/pedestrians, *e.g.* traffic signs, parking signs, direction signs, or warning reflector posts.	
	traffic light	The traffic light box without its poles in all orientations and for all types of traffic participants, *e.g.* regular traffic light, bus traffic light, train traffic light.	
	pole	Small, mainly vertically oriented poles, *e.g.* sign poles or traffic light poles. This does not include objects mounted on the pole, which have a larger diameter than the pole itself (*e.g.* most street lights).	
	pole group[2]	Multiple poles that are cumbersome to label individually, but where the background can be seen in their gaps.	
sky	sky	Open sky (without tree branches/leaves)	

[1] Single instance annotation available.
[2] Not included in challenges.

Table 8. List of annotated classes including their definition and typical example. (continued)

Category	Class	Definition	Examples
flat	road	Horizontal surfaces on which cars usually drive, including road markings. Typically delimited by curbs, rail tracks, or parking areas. However, *road* is not delimited by road markings and thus may include bicycle lanes or roundabouts.	
	sidewalk	Horizontal surfaces designated for pedestrians or cyclists. Delimited from the road by some obstacle, *e.g*. curbs or poles (might be small), but not only by markings. Often elevated compared to the road and often located at the side of a road. The curbs are included in the *sidewalk* label. Also includes the walkable part of traffic islands, as well as pedestrian-only zones, where cars are not allowed to drive during regular business hours. If it's an all-day mixed pedestrian/car area, the correct label is *ground*.	
	parking[2]	Horizontal surfaces that are intended for parking and separated from the road, either via elevation or via a different texture/material, but not separated merely by markings.	
	rail track[2]	Horizontal surfaces on which only rail cars can normally drive. If rail tracks for trams are embedded in a standard road, they are included in the *road* label.	

[1] Single instance annotation available.
[2] Not included in challenges.

Table 8. List of annotated classes including their definition and typical example. (continued)

Category	Class	Definition	Examples
void	ground[2]	All other forms of horizontal ground-level structures that do not match any of the above, for example mixed zones (cars and pedestrians), roundabouts that are flat but delimited from the road by a curb, or in general a fallback label for horizontal surfaces that are difficult to classify, *e.g.* due to having a dual purpose.	
	dynamic[2]	Movable objects that do not correspond to any of the other non-void categories and might not be in the same position in the next day/hour/minute, *e.g.* movable trash bins, buggies, luggage, animals, chairs, or tables.	
	static[2]	This includes areas of the image that are difficult to identify/label due to occlusion/distance, as well as non-movable objects that do not match any of the non-void categories, *e.g.* mountains, street lights, reverse sides of traffic signs, or permanently mounted commercial signs.	
	ego vehicle[2]	Since a part of the vehicle from which our data was recorded is visible in all frames, it is assigned to this special label. This label is also available at test time.	
	unlabeled[2]	Pixels that were not explicitly assigned to a label.	
	out of roi[2]	Narrow strip of 5 pixels along the image borders that is not considered for training or evaluation. This label is also available at test-time.	
	rectification border[2]	Areas close to the image border that contain artifacts resulting from the stereo pair rectification. This label is also available at test time.	

[1] Single instance annotation available.
[2] Not included in challenges.

Table 8. List of annotated classes including their definition and typical example. (continued)

Figure 7. Examples of our annotations on various images of our *train* and *val* sets. The images were selected based on criteria overlayed on each image.

	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**mean IoU**
static fine (SF)	80.0	13.2	40.3	0.0	0.0	0.0	0.0	0.0	12.5	0.0	22.1	0.0	0.0	23.4	0.0	0.0	0.0	0.0	0.0	10.1
static coarse (SC)	80.1	9.5	39.5	0.0	0.0	0.0	0.0	0.0	16.4	0.0	24.3	0.0	0.0	26.2	0.0	0.0	0.0	0.0	0.0	10.3
GT segmentation with SF	80.8	11.1	44.5	0.0	0.0	0.0	0.0	0.0	4.2	0.0	17.9	0.0	0.0	32.9	0.0	0.0	0.0	0.0	0.0	10.1
GT segmentation with SC	79.6	5.1	46.6	0.0	0.0	0.0	0.0	0.0	11.8	0.0	29.2	0.0	0.0	34.1	0.0	0.0	0.0	0.0	0.0	10.9
GT segmentation with [41]	99.3	91.9	94.8	44.9	62.0	66.1	81.2	84.3	96.5	80.1	99.1	90.6	69.2	98.0	59.0	66.9	71.6	66.8	85.8	79.4
GT subsampled by 2	99.6	98.1	98.6	97.8	97.4	90.4	94.1	95.2	98.7	97.6	98.3	96.5	95.7	98.9	98.9	99.1	98.9	96.5	95.8	97.2
GT subsampled by 4	99.4	96.8	98.0	96.1	95.5	83.1	89.7	91.6	98.0	96.0	97.9	94.1	92.5	98.2	98.1	98.5	98.1	94.1	93.0	95.2
GT subsampled by 8	98.6	93.4	95.4	92.3	91.1	69.5	80.9	84.2	95.5	92.1	94.5	88.9	86.1	96.2	95.9	96.7	96.1	88.7	86.8	90.7
GT subsampled by 16	97.8	88.8	93.1	86.9	84.9	50.9	68.4	73.0	93.4	86.5	93.1	81.0	76.0	93.5	93.0	94.4	93.4	80.8	78.0	84.6
GT subsampled by 32	96.0	80.9	88.7	77.6	75.2	30.9	51.6	56.8	89.2	77.3	88.7	69.4	62.3	88.0	87.4	89.8	88.5	68.6	65.6	75.4
GT subsampled by 64	92.1	69.6	83.0	65.5	61.0	14.8	32.1	37.6	83.3	65.2	81.6	55.1	46.4	78.8	78.9	82.4	80.2	54.2	50.7	63.8
GT subsampled by 128	86.2	55.0	75.2	51.3	45.9	5.7	13.6	17.9	75.2	51.6	69.9	41.1	31.5	67.3	66.3	70.1	68.3	36.0	33.3	50.6
nearest training neighbor	85.3	35.6	56.7	15.6	6.2	1.3	0.5	1.0	54.2	23.3	36.5	4.0	0.4	42.0	9.7	18.3	12.9	0.3	1.7	21.3

Table 9. Detailed results of our control experiments for the pixel-level semantic labeling task in terms of the IoU score on the class level. All numbers are given in percent. See the main paper for details on the listed methods.

	person	rider	car	truck	bus	train	motorcycle	bicycle	**mean iIoU**
static fine (SF)	0.0	0.0	38.0	0.0	0.0	0.0	0.0	0.0	4.7
static coarse (SC)	0.0	0.0	39.8	0.0	0.0	0.0	0.0	0.0	5.0
GT segmentation with SF	0.0	0.0	50.3	0.0	0.0	0.0	0.0	0.0	6.3
GT segmentation with SC	0.0	0.0	50.8	0.0	0.0	0.0	0.0	0.0	6.3
GT segmentation with [41]	68.3	44.4	92.8	32.3	38.7	41.5	39.5	63.1	52.6
GT subsampled by 2	91.4	91.9	95.1	93.3	94.1	94.3	91.4	89.6	92.6
GT subsampled by 4	88.1	86.4	94.4	91.8	93.1	93.0	88.9	87.2	90.4
GT subsampled by 8	78.4	75.6	89.7	85.7	87.8	88.8	79.4	76.8	82.8
GT subsampled by 16	63.5	58.5	82.6	73.4	78.2	81.5	66.4	62.3	70.8
GT subsampled by 32	45.5	38.0	71.0	57.7	62.1	66.0	46.2	43.5	53.7
GT subsampled by 64	28.4	19.1	51.0	37.0	42.0	51.4	27.6	24.4	35.1
GT subsampled by 128	19.1	10.5	41.9	18.9	24.5	30.7	11.0	11.8	21.1
nearest training neighbor	3.6	0.5	32.7	1.9	4.0	2.8	0.3	1.5	5.9

Table 10. Detailed results of our control experiments for the pixel-level semantic labeling task in terms of the instance-normalized iIoU score on the class level. All numbers are given in percent. See the main paper for details on the listed methods.

	train	val	coarse	sub	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**mean IoU**
FCN-32s	✓	✓			97.1	76.0	87.6	33.1	36.3	35.2	53.2	58.1	89.5	66.7	91.6	71.1	46.7	91.0	33.3	46.6	43.8	48.2	59.1	61.3
FCN-16s	✓	✓			97.3	77.6	88.7	34.7	44.0	43.0	57.7	62.0	90.9	68.6	92.9	75.4	50.5	91.9	35.3	49.1	45.9	50.7	65.2	64.3
FCN-8s	✓	✓			97.4	78.4	89.2	34.9	44.2	47.4	**60.1**	65.0	91.4	69.3	**93.9**	77.1	51.4	92.6	35.3	48.6	46.5	51.6	**66.8**	65.3
FCN-8s	✓	✓		2	97.0	75.4	87.3	37.4	39.0	35.1	47.7	53.3	89.3	66.1	92.5	69.5	46.0	90.8	41.9	52.9	50.1	46.5	58.4	61.9
FCN-8s		✓			95.9	69.7	86.9	23.1	32.6	44.3	52.1	56.8	90.2	60.9	92.9	73.3	42.7	89.9	22.8	39.2	29.6	42.5	63.1	58.3
FCN-8s			✓		95.3	67.7	84.6	35.9	41.0	36.0	44.9	52.7	86.6	60.2	90.2	59.6	37.2	86.1	35.4	53.1	39.7	42.6	52.6	58.0
[4] ext.	✓			4	95.6	70.1	82.8	29.9	31.9	38.1	43.1	44.6	87.3	62.3	91.7	67.3	50.7	87.9	21.7	29.0	34.7	40.5	56.6	56.1
[4] basic	✓			4	96.4	73.2	84.0	28.5	29.0	35.7	39.8	45.2	87.0	63.8	91.8	62.8	42.8	89.3	38.1	43.1	44.2	35.8	51.9	57.0
[40]	✓	✓	✓	3	96.3	71.7	86.7	43.7	31.7	29.2	35.8	47.4	88.4	63.1	**93.9**	64.7	38.7	88.8	48.0	56.4	49.4	38.3	50.0	59.1
[81]	✓			2	96.3	73.9	88.2	**47.6**	41.3	35.2	49.5	59.7	90.6	66.1	93.5	70.4	34.7	90.1	39.2	57.5	55.4	43.9	54.6	62.5
[9]	✓	✓		2	97.3	77.7	87.7	43.6	40.5	29.7	44.5	55.4	89.4	67.0	92.7	71.2	49.4	91.4	48.7	56.7	49.1	47.9	58.6	63.1
[48]	✓	✓	✓	2	97.4	78.3	88.1	47.5	44.2	29.5	44.4	55.4	89.4	67.3	92.8	71.0	49.3	91.4	**55.9**	**66.6**	**56.7**	48.1	58.1	64.8
[37]	✓				97.3	78.5	88.4	44.5	**48.3**	34.1	55.5	61.7	90.1	**69.5**	92.2	72.5	52.3	91.0	54.6	61.6	51.6	**55.0**	63.1	66.4
[79]	✓				**97.6**	**79.2**	**89.9**	37.3	47.6	**53.2**	58.6	**65.2**	**91.8**	69.4	93.7	**78.9**	**55.0**	**93.3**	45.5	53.4	47.7	52.2	66.0	**67.1**

Table 11. Detailed results of our baseline experiments for the pixel-level semantic labeling task in terms of the IoU score on the class level. All numbers are given in percent and we indicate the used training data for each method, *i.e. train* fine, *val* fine, *coarse* extra, as well as a potential downscaling factor (*sub*) of the input image. See the main paper and Appendix D.1 for details on the listed methods.

	train	val	coarse	sub	person	rider	car	truck	bus	train	motorcycle	bicycle	**mean iIoU**
FCN-32s	✓	✓			46.9	32.0	82.1	21.2	28.8	21.9	26.0	47.1	38.2
FCN-16s	✓	✓			53.6	33.5	84.2	21.3	32.8	25.8	28.9	48.6	41.1
FCN-8s	✓	✓			55.9	33.4	83.9	22.2	30.8	26.7	31.1	49.6	41.7
FCN-8s	✓	✓		2	42.8	22.3	79.3	16.6	27.3	22.2	20.0	38.5	33.6
FCN-8s		✓			51.8	31.0	80.6	17.0	23.9	24.5	23.7	47.3	37.4
FCN-8s			✓		43.2	18.9	72.5	18.2	24.2	20.1	20.9	36.2	31.8
[4] extended	✓			4	49.9	27.1	81.1	15.3	23.7	18.5	19.6	38.4	34.2
[4] basic	✓			4	44.3	22.7	78.4	16.1	24.3	20.7	15.8	33.6	32.0
[40]	✓	✓	✓	3	38.9	12.8	78.6	13.4	24.0	19.2	10.7	27.2	28.1
[81]	✓			2	50.6	17.8	81.1	18.0	25.0	30.3	22.3	30.1	34.4
[9]	✓	✓		2	40.5	23.3	78.8	20.3	31.9	24.8	21.1	35.2	34.5
[48]	✓	✓	✓	2	40.7	23.1	78.6	21.4	32.4	27.6	20.8	34.6	34.9
[37]	✓				56.2	**38.0**	77.1	**34.0**	**47.0**	**33.4**	**38.1**	**49.9**	**46.7**
[79]	✓				**56.3**	34.5	**85.8**	21.8	32.7	27.6	28.0	49.1	42.0

Table 12. Detailed results of our baseline experiments for the pixel-level semantic labeling task in terms of the instance-normalized iIoU score on the class level. All numbers are given in percent and we indicate the used training data for each method, *i.e. train* fine, *val* fine, *coarse* extra, as well as a potential downscaling factor (*sub*) of the input image. See the main paper and Appendix D.1 for details on the listed methods.

Proposals	Classifier	person	rider	car	truck	bus	train	motorcycle	bicycle	**mean AP**
MCG regions	FRCN	**1.9**	**1.0**	6.2	4.0	3.1	2.8	1.5	**0.6**	2.6
MCG bboxes	FRCN	0.5	0.1	7.8	**6.4**	**10.3**	4.5	0.9	0.2	3.8
MCG hulls	FRCN	1.3	0.6	**10.5**	6.1	9.7	**5.9**	**1.7**	0.5	**4.6**
GT bboxes	FRCN	7.6	0.5	17.5	10.7	15.7	8.4	2.6	2.9	8.2
GT regions	FRCN	65.5	40.6	65.9	21.1	31.9	30.2	28.8	46.4	41.3
MCG regions	GT	3.7	4.4	11.9	19.9	21.5	12.4	7.8	2.6	10.5
MCG bboxes	GT	2.0	2.0	10.9	18.2	22.1	15.9	6.0	2.2	9.9
MCG hulls	GT	3.4	4.1	13.4	20.4	24.1	16.0	8.3	2.8	11.6

Table 13. Detailed results of our baseline experiments for the instance-level semantic labeling task in terms of the region-level average precision scores AP on the class level. All numbers are given in percent. See the main paper and Appendix D.2 for details on the listed methods.

Proposals	Classifier	person	rider	car	truck	bus	train	motorcycle	bicycle	**mean AP$^{50\%}$**
MCG regions	FRCN	**6.7**	**5.4**	19.3	10.3	11.9	7.6	7.8	3.0	9.0
MCG bboxes	FRCN	2.7	0.6	23.3	**15.4**	**27.2**	15.2	4.8	1.4	11.3
MCG hulls	FRCN	5.6	3.9	**26.0**	13.8	26.3	**15.8**	**8.6**	**3.1**	**12.9**
GT bboxes	FRCN	35.4	4.3	44.9	19.3	29.9	26.7	11.9	16.7	23.7
GT regions	FRCN	65.5	40.6	65.9	21.1	31.9	30.2	28.8	46.4	41.3
MCG regions	GT	12.3	18.1	29.6	43.9	44.6	31.4	25.9	10.0	27.0
MCG bboxes	GT	9.2	11.5	29.0	41.8	46.0	36.0	23.3	9.6	25.8
MCG hulls	GT	12.0	18.4	31.4	46.1	46.3	40.7	27.7	10.7	29.1

Table 14. Detailed results of our baseline experiments for the instance-level semantic labeling task in terms of the region-level average precision scores AP$^{50\%}$ for an overlap value of 50 %. All numbers are given in percent. See the main paper and Appendix D.2 for details on the listed methods.

Proposals	Classifier	person	rider	car	truck	bus	train	motorcycle	bicycle	**mean AP$^{100\mathrm{m}}$**
MCG regions	FRCN	**3.7**	**1.6**	10.2	6.8	5.4	4.2	2.2	**1.1**	4.4
MCG bboxes	FRCN	0.9	0.1	12.9	**11.3**	**18.5**	6.9	1.3	0.3	6.5
MCG hulls	FRCN	2.6	1.1	**17.5**	10.6	17.4	**9.2**	**2.6**	0.9	**7.7**
GT bboxes	FRCN	8.8	0.8	25.3	18.4	27.1	13.0	3.9	3.6	12.6
GT regions	FRCN	79.1	66.0	78.9	33.6	53.9	47.1	42.6	63.5	58.1
MCG regions	GT	6.8	6.8	18.9	28.7	32.7	19.0	10.5	4.3	16.0
MCG bboxes	GT	3.5	2.9	17.3	27.3	34.5	24.9	8.2	3.7	15.3
MCG hulls	GT	6.1	6.2	21.4	29.9	37.2	24.7	11.4	4.7	17.7

Table 15. Detailed results of our baseline experiments for the instance-level semantic labeling task in terms of the region-level average precision scores AP$^{100\mathrm{m}}$ for objects within 100 m. All numbers are given in percent. See the main paper and Appendix D.2 for details on the listed methods.

Proposals	Classifier	person	rider	car	truck	bus	train	motorcycle	bicycle	**mean AP$^{50\mathrm{m}}$**
MCG regions	FRCN	**4.0**	**1.7**	12.0	9.0	7.8	6.4	2.4	**1.1**	5.5
MCG bboxes	FRCN	1.0	0.1	15.5	**14.9**	**27.7**	10.0	1.4	0.4	8.9
MCG hulls	FRCN	2.7	1.1	**21.2**	14.0	25.2	**14.2**	**2.7**	1.0	**10.3**
GT bboxes	FRCN	8.5	0.8	26.6	23.2	37.2	17.7	4.1	3.6	15.2
GT regions	FRCN	79.1	68.3	80.5	42.9	69.4	67.9	46.2	64.7	64.9
MCG regions	GT	7.2	7.0	21.7	32.4	42.4	23.6	11.1	4.5	18.7
MCG bboxes	GT	3.7	3.0	19.9	33.0	46.0	32.9	8.6	3.8	18.9
MCG hulls	GT	6.5	6.4	24.8	35.4	49.6	31.8	12.2	4.9	21.4

Table 16. Detailed results of our baseline experiments for the instance-level semantic labeling task in terms of the region-level average precision scores AP$^{50\mathrm{m}}$ for objects within 50 m. All numbers are given in percent. See the main paper and Appendix D.2 for details on the listed methods.

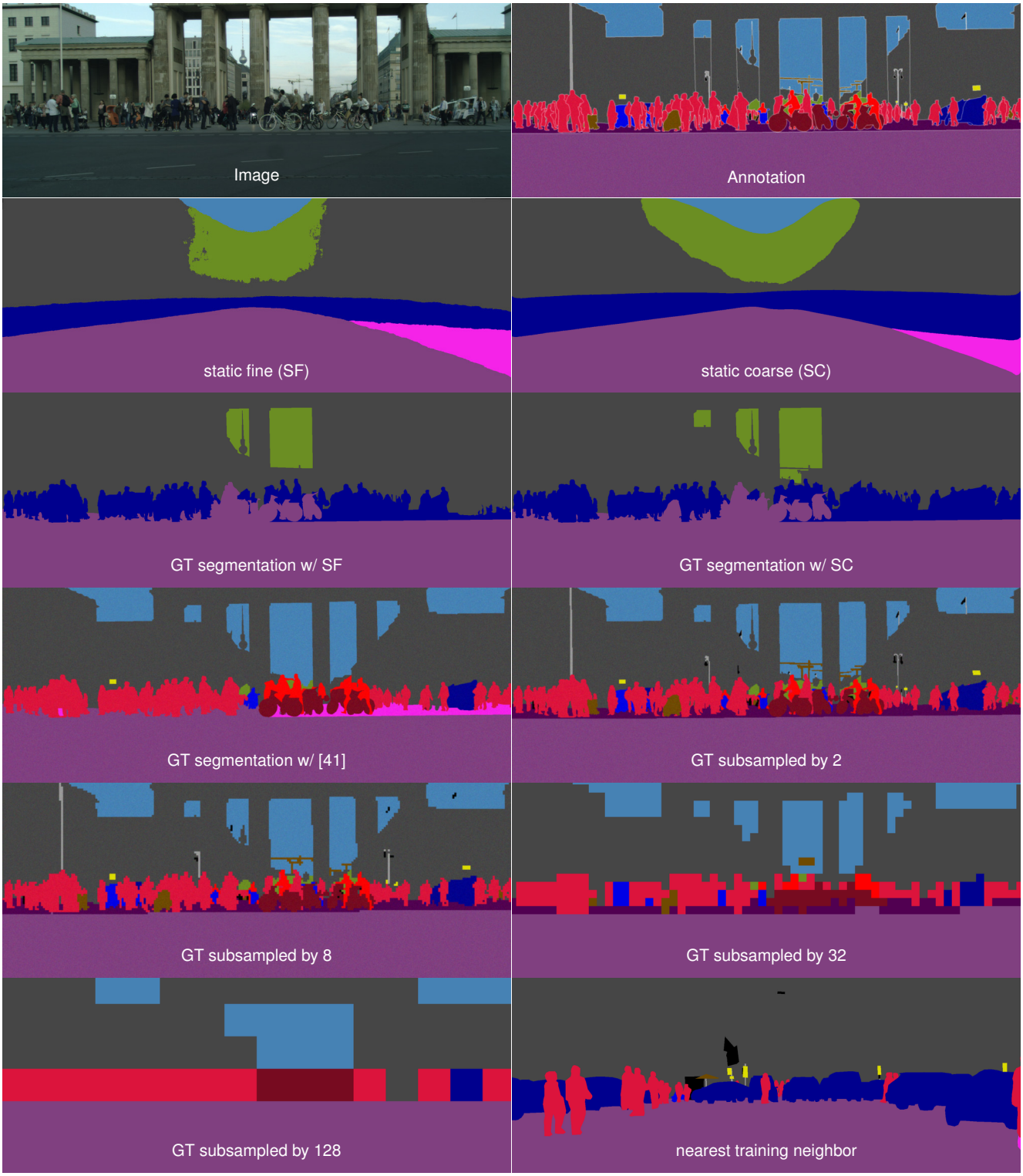

Figure 8. Exemplary output of our control experiments for the pixel-level semantic labeling task, see the main paper for details. The image is part of our *test* set and has both, the largest number of instances and persons.

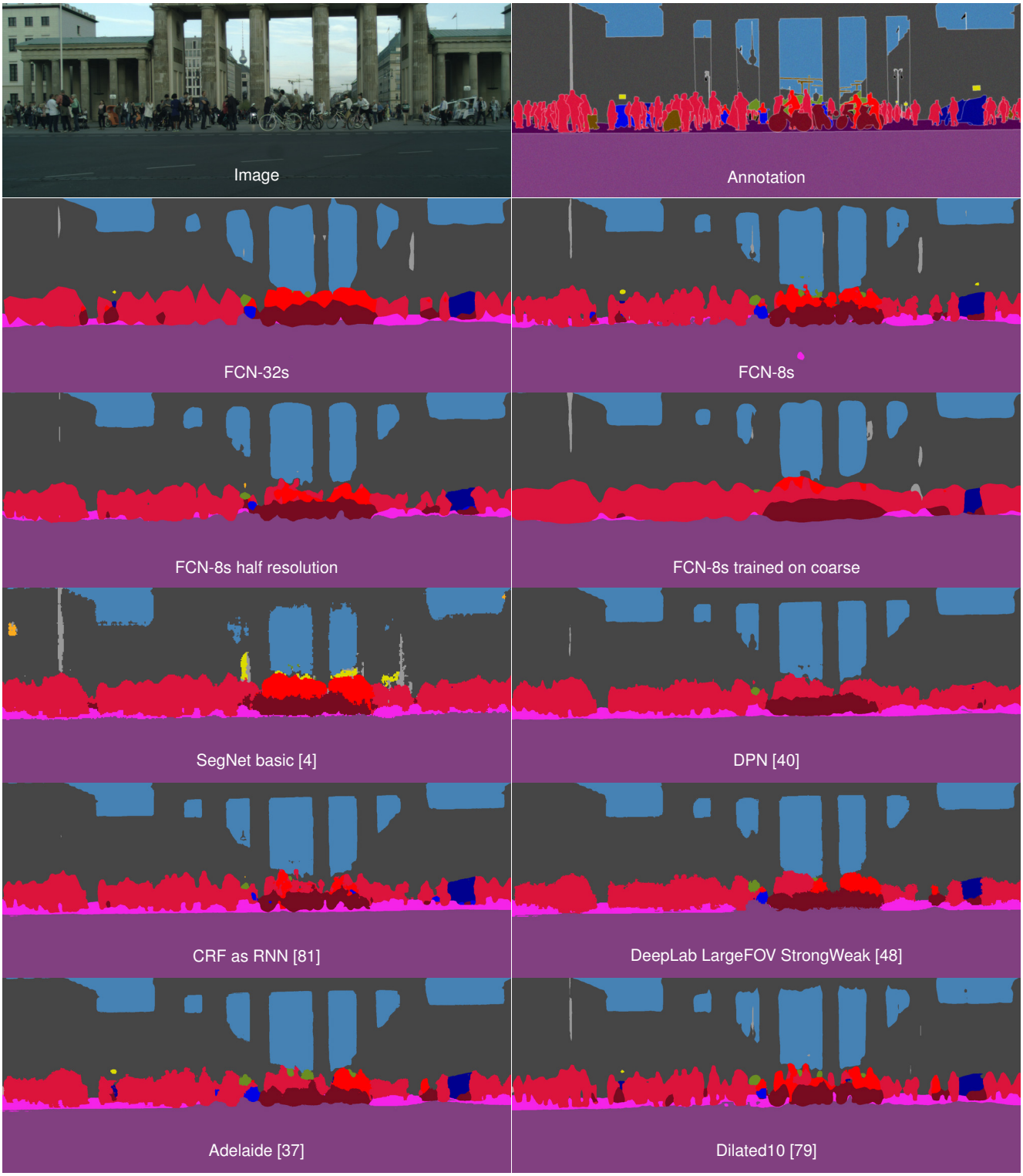

Figure 9. Exemplary output of our baselines for the pixel-level semantic labeling task, see the main paper for details. The image is part of our *test* set and has both, the largest number of instances and persons.

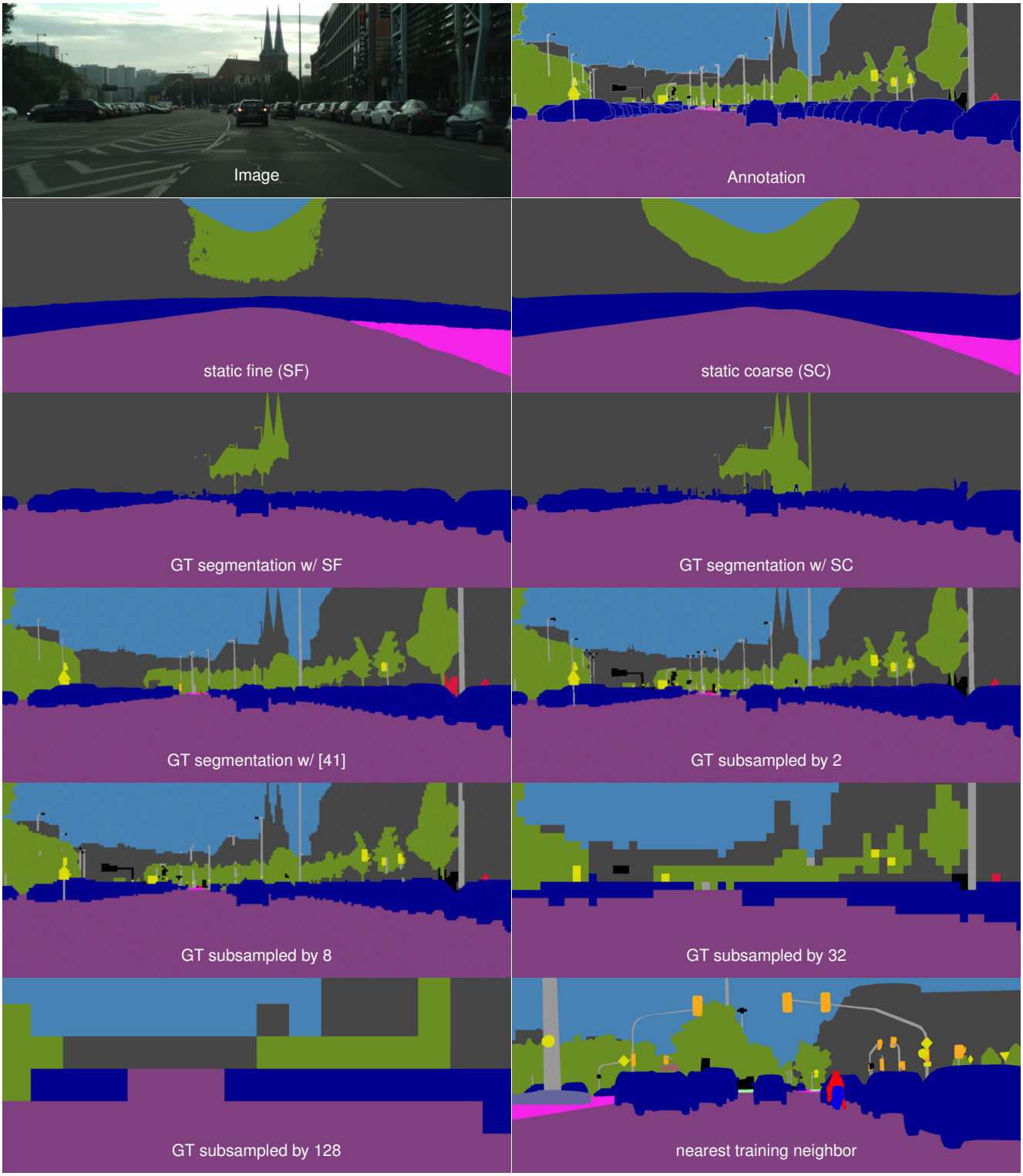

Figure 10. Exemplary output of our control experiments for the pixel-level semantic labeling task, see the main paper for details. The image is part of our *test* set and has the largest number of car instances.

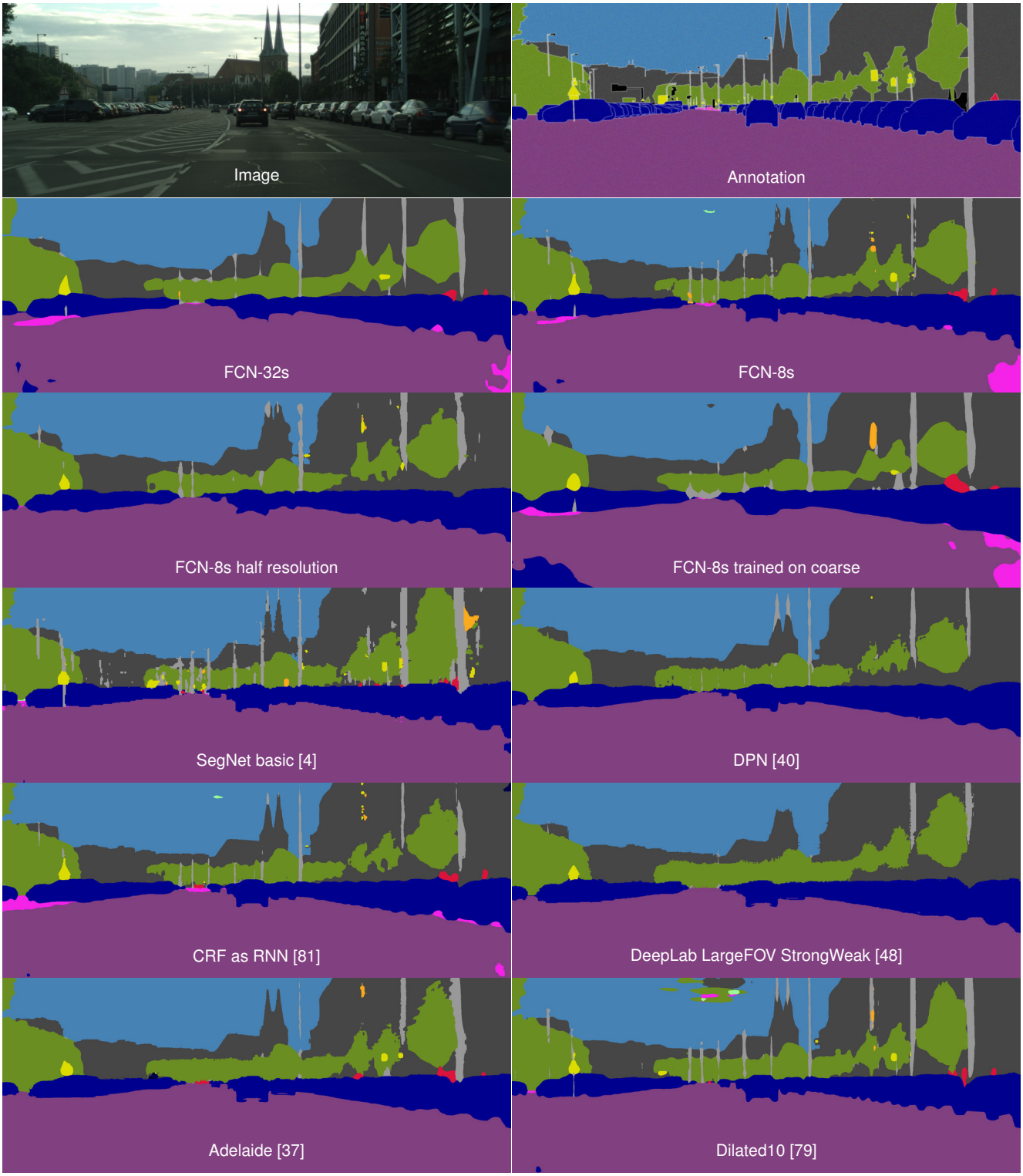

Figure 11. Exemplary output of our baseline experiments for the pixel-level semantic labeling task, see the main paper for details. The image is part of our *test* set and has the largest number of car instances.

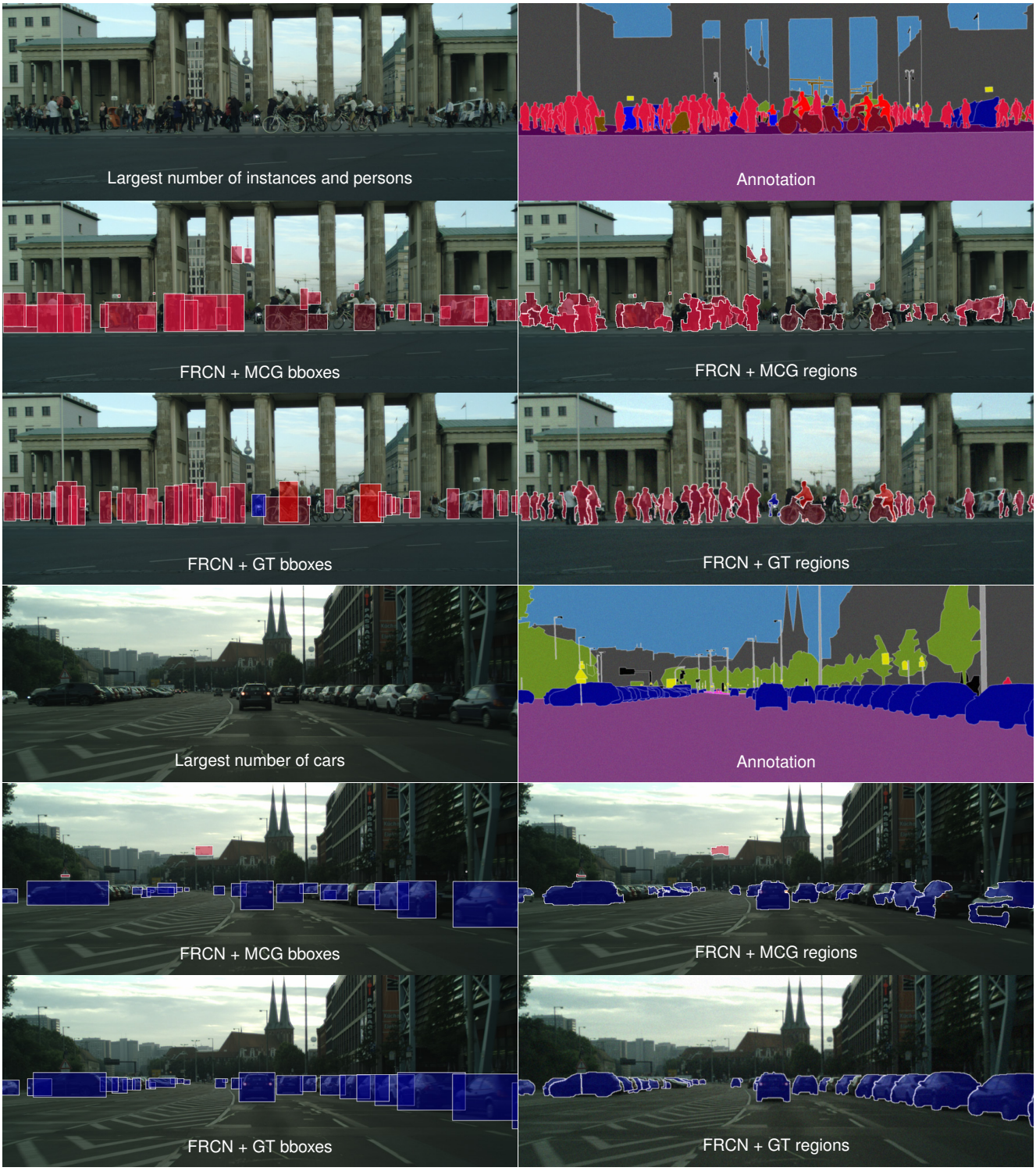

Figure 12. Exemplary output of our control experiments and baselines for the instance-level semantic labeling task, see the main paper for details.