Semantic Instance Segmentation for Autonomous Driving

Bert De Brabandere Davy Neven Luc Van Gool
ESAT-PSI, KU Leuven
first.last@esat.kuleuven.be

Abstract

Semantic instance segmentation remains a challenge. We propose to tackle the problem with a discriminative loss function, operating at pixel level, that encourages a convolutional network to produce a representation of the image that can easily be clustered into instances with a simple post-processing step. Our approach of combining an off-the-shelf network with a principled loss function inspired by a metric learning objective is conceptually simple and distinct from recent efforts in instance segmentation and is well-suited for real-time applications. In contrast to previous works, our method does not rely on object proposals or recurrent mechanisms and is particularly well suited for tasks with complex occlusions. A key contribution of our work is to demonstrate that such a simple setup without bells and whistles is effective and can perform on-par with more complex methods. We achieve competitive performance on the Cityscapes segmentation benchmark.

1. Introduction

Semantic instance segmentation has recently gained in popularity. As an extension of regular semantic segmentation, the task is to generate a binary segmentation mask for each individual object along with a semantic label. It is considered a fundamentally harder problem than semantic segmentation - where overlapping objects of the same class are segmented as one.

One key factor that complicates the naive application of the popular softmax cross-entropy loss function to instance segmentation, is the fact that an image can contain an arbitrary number of instances and that the labeling is permutation-invariant.

To cope with this, one popular approach is to introduce a multi-stage pipeline with object proposals [9, 4, 16, 7]. Another approach is to train a recurrent network end-to-end with a custom loss function that outputs instances sequentially [14, 18, 17]. Whereas the first approach has difficulty handling complex occlusions, the recurrent methods generally have complex network architectures making them inherently slower than standard feed-forward networks.

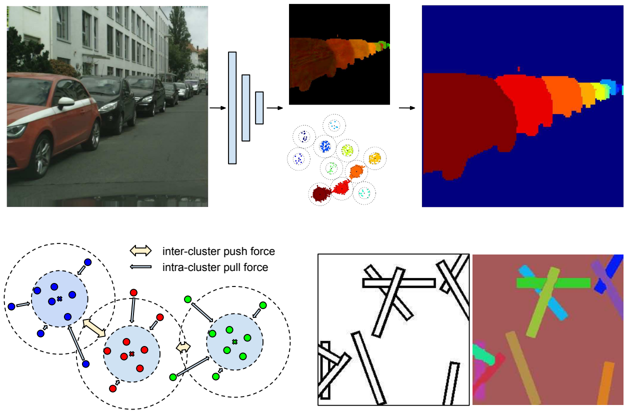

Figure 1. Top: the network maps each pixel into feature space, in which each object can be easily clustered with a fast clustering approach. Bottom left: intra-cluster pull and inter-cluster push forces. Bottom right: our method can handle complex occlusions, useful for pick and place tasks.

Another line of research is to train a network to transform the image into a representation that is clustered into individual instances with a post-processing step [24, 20, 13]. Our method belongs to this last category, but is less ad-hoc and takes a more principled approach than previous works.

We focus on the loss function, as we aim to re-use feed-forward network architectures that were designed for semantic segmentation: we plug in an off-the-shelf architecture and retrain the system with our discriminative loss function. In our framework, the tasks of semantic and instance segmentation can be treated in a consistent and similar manner and do not require changes to the architecture.

2. Method

Consider a differentiable function that maps each pixel in an input image to a point in n-dimensional feature space, referred to as the *pixel embedding*. The intuition behind our loss function is that pixel embeddings with the same label (same instance) should end up close together, while embeddings with a different label (different instance) should end

	AP	AP0.5	AP100m	AP50m
InstanceCut [11]	13.0	27.9	22.1	26.1
DWT [3]	15.6	30.0	26.2	31.8
Shape-aware [10]	17.4	36.7	29.3	34.0
Pixelwise DIN [1]	20.0	38.8	32.6	37.6
Ours	17.5	35.9	27.9	31.0

Table 1. Segmentation results of best performing entries on the test set of the Cityscapes instance segmentation benchmark.

up far apart.

Inspired by Weinberger *et al.* [21] and other distance metric learning approaches [5, 12, 19], we propose a loss function with two competing terms to achieve this objective: a *variance term* pulling embeddings towards the mean embedding of their cluster and a *distance term* pushing the clusters away from each other, see fig. 1. To relax the constraints on the network, we hinge the variance and distance terms: embeddings within a distance of δ_v from their cluster centers are no longer attracted to it and cluster centers further apart than $2\delta_v$ are no longer repulsed. We add a small regularization pull-force that draws all clusters towards the origin to keep the activations bounded. These three terms can be written as follows, with C the number of clusters in ground truth, N_c the number of elements in cluster c, x_i an embedding, μ_c the mean embedding of cluster c, $\|\cdot\|$ the L2 distance, and $[x]_+ = \max(0, x)$ the hinge:

$$\begin{cases} L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}\left[\|\mu_c - x_i\| - \delta_{\mathrm{v}}\right]_+^2 \\ \\ L_{dist} = \frac{1}{C(C-1)}\sum_{c_A=1}^{C}\sum_{c_B=1, c_A\neq c_B}^{C}\left[2\delta_{\mathrm{d}} - \|\mu_{c_A} - \mu_{c_B}\|\right]_+^2 \\ \\ L_{reg} = \frac{1}{C}\sum_{c=1}^{C}\|\mu_c\| \end{cases}$$

When the loss has converged, all pixel embeddings are within a distance of δ_v from their cluster center and all cluster centers are at least $2\delta_d$ apart. By setting $\delta_d > 2\delta_v$, each embedding is closer to all embeddings of its *own* cluster than to any embedding of a *different* cluster. During inference we can can then threshold with bandwidth $b = \delta_v$ around any embedding to select all embeddings belonging to the same cluster. Since the loss on the test set will not be zero, we apply a fast variant of the mean-shift algorithm [8] to shift to a center pixel around which to cluster.

3. Experiments

Occlusion handling A key strength of our method is its ability to handle complex occlusions. Detect-and-segment approaches like [7, 10] require an object's segmentation mask to be unambiguously extracted from its bounding box. This assumption is problematic for certain tasks. Consider a pick-and-place task where overlapping stick-like objects need to be segmented as in fig. 1. When two sticks overlap like two crossed swords, their bounding boxes are highly overlapping. Given only a detection in the form of a bounding box, it is exceedingly hard to unambiguously extract a segmentation mask of the indicated object. In contrast to methods that rely on bounding boxes, our method treats the image holistically and can learn to reason about occlusions.

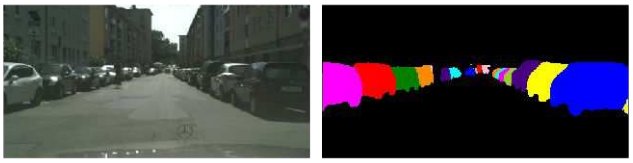

Figure 2. Visual results on Cityscapes.

	Dim	AP	AP_{gt}	fps	#p	mem
ENet [15]	512 x 256	0.19	0.21	145	0.36	1.00
	768 x 384	0.21	0.25	94		1.03
	1024 x 512	0.20	0.26	61		1.12
Segnet [2]	512 x 256	0.20	0.22	27	29.4	1.22
	768 x 384	0.22	0.26	14		1.29
	1024 x 512	0.18	0.24	8		1.47
Dilation [23]	512 x 256	0.21	0.24	15	134.3	2.20
	768 x 384	0.24	0.29	6		2.64
	1024 x 512	0.23	0.30	4		3.27
Resnet38 [22]	512 x 256	0.24	0.27	12	124	4.45
	768 x 384	0.29	0.34	5		8.83

Table 2. Average Precision (AP), AP using gt segmentation labels (AP_{gt}), speed of forward pass (fps), number of parameters ($\times 10^6$) and memory usage (GB) for different models evaluated on the car class of the Cityscapes validation set.

Scene understanding for autonomous driving We test our loss function on the challenging Cityscapes dataset [6], a multi-class semantic instance segmentation benchmark. To cope with the multi-class problem, we apply our loss function independently on each semantic class so that instances from different classes are free to occupy the same feature space. The semantic segmentation masks are obtained with the ResNet-38 network from [22]. The same architecture, pretrained on Cityscapes semantic segmentation, is also adopted for our instance segmentation network. We train the model on the 2975 training images, resized to 768x384 and use Adam with learning rate of 1e-4 on a NVIDIA Titan X. With our loss we achieve competitive results on the Cityscapes leaderboard, see table 1.

To investigate the trade-off between speed, accuracy and memory requirements, we train 4 different network models on different resolutions and evaluate them on the car class of the Cityscapes validation set. This also illustrates the benefit that our method can be used with any off-the-shelf network designed for semantic segmentation.

Table 2 shows the results. We can conclude that Resnet-38 is the best for accuracy, but requires some more memory.

Acknowledgement: The work was supported by Toyota, and was carried out at the TRACE Lab at KU Leuven (Toyota Research on Automated Cars in Europe - Leuven).

References

[1] A. Arnab and P. H. S. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *CVPR*, 2017.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[3] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. *arXiv preprint arXiv:1611.08303*, 2016.

[4] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In *CVPR*, 2015.

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[6] 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 *CVPR*, 2016.

[7] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[8] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975.

[9] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[10] Z. Hayder, X. He, and M. Salzmann. Shape-aware instance segmentation. *arXiv preprint arXiv:1612.03129*, 2016.

[11] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. Instancecut: from edges to instances with multicut. *arXiv preprint arXiv:1611.08272*, 2016.

[12] M. Koestinger, M. Hirzer, P. Wohlhart, P. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, 2012.

[13] X. Liang, Y. Wei, X. Shen, J. Yang, L. Lin, and S. Yan. Proposal-free network for instance-level object segmentation. *arXiv preprint arXiv:1509.02636*, 2015.

[14] E. Park and A. C. Berg. Learning to decompose for object detection and instance segmentation. *arXiv preprint arXiv:1511.06449*, 2015.

[15] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[16] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016.

[17] M. Ren and R. S. Zemel. End-to-end instance segmentation and counting with recurrent attention. *arXiv preprint arXiv:1605.09410*, 2016.

[18] B. Romera-Paredes and P. H. Torr. Recurrent instance segmentation. In *ECCV*, 2016.

[19] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[20] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. *GCPR*, 2016.

[21] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[22] Z. Wu, C. Shen, and A. van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.

[23] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[24] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *ICCV*, 2015.