LevelSet R-CNN: A Deep Variational Method for Instance Segmentation

Namdar Homayounfar[1,2(✉)], Yuwen Xiong[1,2(✉)], Justin Liang[1(✉)],
Wei-Chiu Ma[1,3], and Raquel Urtasun[1,2]

[1] Uber Advanced Technologies Group, Pittsburgh, USA
{namdar,yuwen,justin.liang,weichiu,urtasun}@uber.com
[2] University of Toronto, Toronto, Canada
[3] MIT, Cambridge, USA

Abstract. Obtaining precise instance segmentation masks is of high importance in many modern applications such as robotic manipulation and autonomous driving. Currently, many state of the art models are based on the Mask R-CNN framework which, while very powerful, outputs masks at low resolutions which could result in imprecise boundaries. On the other hand, classic variational methods for segmentation impose desirable global and local data and geometry constraints on the masks by optimizing an energy functional. While mathematically elegant, their direct dependence on good initialization, non-robust image cues and manual setting of hyperparameters renders them unsuitable for modern applications. We propose LevelSet R-CNN, which combines the best of both worlds by obtaining powerful feature representations that are combined in an end-to-end manner with a variational segmentation framework. We demonstrate the effectiveness of our approach on COCO and Cityscapes datasets.

1 Introduction

Instance segmentation, the task of detecting and categorizing the pixels of unique countable objects in an image, is of paramount interest in many computer vision applications such as medical imaging [67], photo editing [68], pose estimation [50], robotic manipulation [21] and autonomous driving [69]. With the advent of deep learning [38] and its tremendous success in object classification and detection tasks [24,58,59], the computer vision community has made great strides in instance segmentation [2,3,10,30,47,64,66].

Currently, the prevailing instance segmentation approaches are based on the Mask R-CNN [27] framework which detects and classifies objects in the image and further processes each instance to produce a binary segmentation mask. While achieving impressive results in many benchmarks, the predicted masks

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58592-1_33) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12368, pp. 555–571, 2020.
https://doi.org/10.1007/978-3-030-58592-1_33

are produced at a low resolution and label predictions are independent per pixel, which could result in imprecise boundaries and irregular object discontinuities.

In contrast, traditional variational segmentation methods [6,7,31] are explicitly designed to delineate the boundaries of objects and handle complicated topologies. They first encode desired geometric properties into an energy functional and then evolve an initial contour according to the minimization landscape of the energy functional. One seminal work in this direction is the *Chan-Vese* [7] level set method, which formulates the segmentation problem as a partitioning task where the goal is to divide the image into two regions, each of which has similar intensity values. Through an energy formulation, Chan-Vese can produce good results even from a coarse initialization. However, in the real world, the photometric values may not be consistent, for example due to illumination changes and varying textures, rendering this method impractical for modern challenging applications.

With these problems in mind, we propose *LevelSet R-CNN*, a novel deep structured model that combines the strengths of modern deep learning with the energy based Chan-Vese segmentation framework. Specifically, we build our model in a multi-task setting following the Mask R-CNN framework: four different heads are utilized based on Feature Pyramid Network (FPN) [43] to output object localization and classification, a truncated signed distance function (TSDF) as the mask initialization, a set of instance-aware energy hyperparameters, and a deep object feature embedding, as shown in Fig. 1. These intermediate outputs are then passed into a differentiable unrolled optimization module to refine the predicted TSDF mask of each detected object by minimizing the Chan-Vese energy functional. This results in more precise object masks at higher resolutions.

We evaluate the effectiveness of our method on the challenging Cityscapes [15] instance segmentation task, where we achieve state-of-the-art results. We show also improvements over the baseline on the COCO [44] and the higher quality LVIS [25] datasets. Finally, we evaluate our model choices through extensive ablation studies.

2 Related Work

Instance Segmentation: Current modern instance segmentation methods can be classified as being either a top down or a bottom up approach. In a top down approach [5,8,9,22,33], region proposals for each instance are generated and a voting process is used to determine which ones to keep. Masks are predicted from these proposals to obtain the final instance segmentation output. For example, [17] uses a cascade of networks to predict boxes, estimate masks and categorize objects in a sequential manner so that the convolutional features are shared between the tasks. In [40], the authors use position sensitive inside/outside score maps to jointly perform detection and object segmentation. Recently, Mask R-CNN [27] augments Faster R-CNN [59] to achieve very strong instance segmentation performance across benchmarks. Following this paper, the authors in [30] optimize the scores of the bounding boxes to match the mask IoU, [47] adds a bottom to top aggregation path to allow for better information flow to improve the performance and [35,66] extend it to panoptic segmentation. In [39] the authors improve an initial segmentation by fine-tuning it using a recurrent unit [14] that mimics level set evolution. Our approach is also top down. Here we add structure to the output space of Mask R-CNN by optimizing an explicit energy functional that incorporates geometrical constraints.

The bottom up approaches [4,20,36,54,64] typically perform segmentation by grouping the feature embeddings of individual instances without any early stage object proposals. In [42], the authors develop a model that predicts the category confidence, instance number and instance location and use a normalized spectral clustering algorithm [55] to group the instances together. In [69,70], a CNN outputs instance labels followed by a Markov Random Field to achieve a coherent and consistent labeling of the global image. In [3], the authors exploit a CNN to output a deep watershed energy which can be thresholded to obtain the instance components. [46] use a sequence of neural networks to solve a subgrouping problem that gradually increase in complexity to group pixels of the same instance. In [32], the authors propose a multi task framework that as a subtask groups pixels by regressing a vector pointing towards the object's center. [53] are able to achieve real time instance segmentation by introducing and using a new clustering loss that encourages pixels to point towards an optimal region around the instance center. While bottom up approaches have a much simpler design than top down methods, they usually underperform in standard metrics such as average precision and recall. In our work, we cluster feature embeddings of an instance by differentiable optimization of an energy functional embedded within a state of the art top down approach.

Variational Methods: The classic pioneering active contour models (ACM) of [31] formulate the segmentation task as the minimization of an energy functional w.r.t. an explicit contour parametrization of the boundaries. This energy functional is comprised of a data term that moves the contours to areas of high gradient in the image. Furthermore, it regularizes the contour in terms of its smoothness and curvature. The shortcomings of ACM is that it is sensitive to initialization and requires heuristics such as re-sampling of points to handle changes of topology of the contour. The Level Set frameworks of [19,56] overcome these challenges by formulating the segmentation task as finding the zero-level crossing of a higher dimensional function. In this framework, the contours of an object are implicitly defined as the zero crossing of an embedding function such as the TSDF. This eliminates the need for heuristics to handle complicated object topologies [16]. In this work, we build upon the level set framework put forward by Chan and Vese [7] where we exploit neural networks to learn robust features and optimization schedules from data.

In recent years, several works have explored combining these classical variational methods with neural networks. In the context of building segmentation from aerial images, CNNs have been deployed to output the energy terms used to evolve an active contour and develop a deep structured model that can be learned end-to-end [13,51]. In [26,45], the authors predict the offset to an initial circle to obtain object polygons and use a differentiable renderer to compare with the ground truth mask in the presence of a ground truth bounding box. However, they are not minimizing an explicit energy functional. These works focus on a simpler setting than us where detection is eschewed in favor of using ground truth boxes and a dedicated neural network for segmentation. Moreover, they parameterize the output space with explicit polygons which are not able to handle multi component objects without heuristics. In our work, we tackle the full instance segmentation setting with a single backbone and also use implicit level sets that can naturally handle complicated topologies without heuristics.

In the context of implicit contours, certain works have explored leveraging level sets in neural networks either as a post processing step to obtain ground truth data, or as a loss function for deep neural networks. In a semi-supervised setting, initial masks have been predicted for unlabeled data then further refined with level set evolution to create a quasi ground truth label [63]. The authors in [11,29,34] employ level set energies as a loss function for saliency estimation and semantic segmentation. In contrast, we employ level set optimization as a differentiable module within a deep neural network. In the experimental section, we evaluate the efficacy of using a level set loss function for the task of instance segmentation. The closest work to ours is [65], where the authors embed a different level set optimization framework within a neural network for the task of annotator in the loop foreground segmentation. There are several key differences: (i) The energy formulation is different, whereas their work is built upon the edge based method of [6] to push the contour to the boundaries, we exploit the region based approach of [7] which imposes uniformity of object masks. (ii) their setting requires ground truth object bounding boxes to output the features used in the level set optimization, while we embed the optimization within Mask R-CNN to build on top of shared features. Note that our setting is much more challenging. In the experimental section, we extend their method to the setting of instance segmentation and compare to our proposed model.

3 Overview of Chan-Vese Segmentation

In this section we provide a brief overview of the classic Chan-Vese level set segmentation method [7], which we later combine in a differentiable manner with Mask R-CNN. Chan-Vese is a region based segmentation approach which is capable of segmenting objects with complex topologies, e.g., holes and multiple components. This method operates globally on image intensities and is not dependent on local well-defined edge information. At a high level, Chan-Vese partitions an image to foreground and background segments by minimizing an energy functional that encourages regions to have uniform intensity values.

Let I be an image defined on the image plane $\Omega \subset \mathbb{R}^2$. Suppose I contains only one object that we wish to segment. Let $\phi : \Omega \to \mathbb{R}$ be the truncated signed distance function (TSDF) to the boundaries of this object taking positive values inside the object and negative outside. Let the curve C correspond to, possibly multi-component, boundaries of this object. The curve C can implicitly be defined as the zero crossing of ϕ, i.e., $C = \{x \in \mathbb{R}^2 \mid \phi(x) = 0\}$.

The core idea is to evolve an initial TSDF ϕ_0 by minimizing an energy functional E such that the zero crossing C of the minimizer coincides with the object boundaries. In the Chan-Vese [7] framework, the energy functional is defined as:

$$\begin{aligned} E(\phi, c_1, c_2) = {} & \lambda_1 \int_\Omega \|I(x) - c_1\|^2 \; H(\phi(x)) dx \\ & + \lambda_2 \int_\Omega \|I(x) - c_2\|^2 (1 - H(\phi(x))) dx + \mu \int_\Omega \delta(\phi(x)) \|\nabla \phi(x)\| \, dx \end{aligned} \tag{1}$$

where H and δ are Heaviside and Dirac delta functions respectively. The first two terms encourage the image intensity values inside and outside of the object to be close to constants c_1 and c_2 respectively. These terms impose a partitioning of the image to two regions of similar intensity values. The last term regularizes the length of the zero level set C. The parameters μ, λ_1 and λ_2 are positive global hyperparameters that regulate the contribution of each energy term.

The minimization of Eq. (1) is achieved by alternatively optimizing the function ϕ and the constants c_1 and c_2. In particular, by holding ϕ fixed, the minimizer of Eq. (1) w.r.t. c_1 and c_2 is given by:

$$c_1(\phi) = \frac{\int_\Omega I(x) H(\phi(x)) dx}{\int_\Omega H(\phi(x)) dx} \quad , \quad c_2(\phi) = \frac{\int_\Omega I(x)(1 - H(\phi(x))) dx}{\int_\Omega (1 - H(\phi(x))) dx} \tag{2}$$

We thus observe that c_1 and c_2 correspond to the average of the intensity values inside and outside of the object respectively.

Next, by holding c_1 and c_2 fixed and introducing an artificial time constant $t \geq 0$, we compute the functional derivative of E w.r.t. ϕ:

$$\frac{\partial \phi(\varepsilon)}{\partial t} = \delta_\varepsilon(\phi) \Big(\mu \mathrm{div}(\frac{\nabla \phi}{\|\nabla \phi\|}) - \lambda_1 \|I - c_1\|^2 + \lambda_2 \|I - c_2\|^2 \Big) \tag{3}$$

where *div* is the divergence operator, ∇ is the spatial derivative and we have used a soft version of H and δ defined as:

$$H_\varepsilon(z) = \frac{1}{2}\Big(1 + \frac{2}{\pi} \arctan(\frac{z}{\varepsilon})\Big) \quad , \quad \delta_\varepsilon(z) = \frac{1}{\pi} \cdot \frac{\varepsilon^2}{\varepsilon^2 + z^2} \tag{4}$$

Finally, the update step of ϕ is given by:

$$\phi_n = \phi_{n-1} + \Delta t \frac{\partial \phi(\varepsilon)}{\partial t} \tag{5}$$

The alternating optimization is repeated for N iterations. This procedure draws similarities to clustering techniques such as K-Means, where the optimization involves alternating assignments and cluster center computations.

While Chan-Vese segmentation is mathematically elegant and powerful, working directly on image intensities is not robust due to factors such as lighting, different textures, motion blur or backgrounds that have similar intensities to the foreground. Moreover, the energy and optimization hyperparameters such as μ, λ_1 and λ_2, that balance the energy terms, and ε and Δt that regulate the gradient descent have to be manually adjusted depending on the image and domain. Furthermore, different objects have different optimal hyperparameters as their appearance and resolution might be very different. As a consequence, this method is not used in modern segmentation algorithms. In this paper, we leverage the power of deep learning to learn high dimensional object representations where the representations of pixels of the same object instance cluster together. We also learn complex inference schedules via data dependent adaptive hyperparameters for the energy terms and the optimization.

4 LevelSet R-CNN

In this section, we develop a deep structured model for the task of instance segmentation by combining the strengths of modern deep neural networks with the classical continuous energy based Chan-Vese [7] segmentation framework. In particular, we build on top of Mask R-CNN [27], which has been widely adopted for object localization and segmentation. However, the masks it produces suffer from low resolution resulting in segmentations that roughly have the right shape but are not precise. Moreover, pixel predictions are independent and there is no explicit mechanism encouraging neighboring pixels to have the same label. On the other hand, the Chan-Vese segmentation framework provides an elegant mathematical approach for global region based segmentation which encourages the pixels within the object to have the same label. However, it suffers in the presence of objects with different appearances within the instance, as it relies on non-robust intensity cues. In this paper we take the best of both worlds by combining these two paradigms.

We build on top of Mask R-CNN to first locate the objects in the image from the detection branch. Next, for each detected RoI corresponding to that object, we predict an initial TSDF ϕ_0, the set of hyperparameters $\{\mu, \lambda_1, \lambda_2\}$ for the energy terms and $\{\varepsilon, \Delta t\}$ for the optimization, and finally a deep feature embedding F that will replace the image intensities in (1). These predictions in turn will be fed into the Chan-Vese module where the costs are created and the optimization is unrolled for N steps as layers of a feedforward neural network. This module will output an evolved TSDF ϕ_N for each object such that its zero crossing corresponds to the boundaries of this object.

In what follows, we first describe how we build on top of Mask R-CNN, and then discuss how inference is performed in the deep Chan-Vese module. Finally, we will describe how learning is done in an end-to-end manner.

4.1 LevelSet R-CNN Architecture

Here we describe the specifics of the backbone and the additional heads of our model that provide the necessary components for the Chan-Vese optimization. The model architecture is presented in Fig. 1.

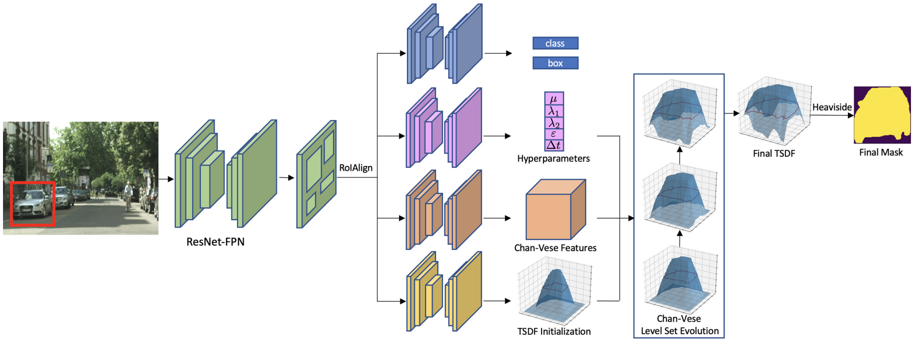

Fig. 1. LevelSet R-CNN for Instance Segmentation: We build on top of Mask R-CNN to first detect and classify all objects in the image. Then for each detection, the corresponding RoI is fed to a series of convolutions to obtain a truncated signed distance function (TSDF) initialization, a deep feature tensor, and a set of instance aware adaptive hyperparameters. These in turn are inputted into an unrolled Chan-Vese level set optimization procedure which outputs a final TSDF. We obtain a mask by applying the Heaviside function to the TSDF.

Backbone, Object Localization and Classification: As our shared backbone we employ a Residual Network [28] augmented with an FPN [43] and an RPN [59] that provides object region proposals. For object localization and classification, we maintain the original head structure of Mask R-CNN where RoIs are passed through a series of fully connected layers to output bounding box coordinates and object classification scores. Next, using *RoIAlign* [27] we extract features from the backbone that are further processed by the *initial TSDF head*, *hyperparameter head*, and the *Chan-Vese features head*. We denote the features corresponding to a RoI by r_m for $m \in \{1, \ldots, M\}$. We refer the reader to the supplementary material for the exact architectural details.

Initial TSDF Head: We replace the binary output of the mask head of Mask R-CNN to produce a TSDF output instead. Specifically, each pixel of the 28×28 output provides the signed ℓ_2 distance to the closest point on the object boundary. Furthermore, we threshold the values to a fixed symmetric range and normalize to $[-1, 1]$. This output is upsampled to 112×112 and used as the initial TSDF $\phi_0(r_m)$ in Eq. (1).

Hyperparameter Head: Each object instance could benefit from an adaptive set of hyperparameters for the energy terms and optimization steps. To achieve this, we output $\lambda_1(r_m)$ and $\lambda_2(r_m)$ to adaptively balance the influence of the foreground and background pixels in Eq. 1 for the object. We also predict $\mu(r_m)$ to regulate the length of its boundary. For the optimization hyperparameters, we output a separate $\varepsilon_n(r_m)$ for each of the N iterations. As shown in Eqs. 3 and 5, larger values of $\varepsilon_n(r_m)$ update the TSDF ϕ_n more globally and smaller values focus the evolution on the boundaries. Similarly, we output N step sizes $\Delta t_n(r_m)$ for each gradient descent step.

To predict the above hyperparameters, we add an additional head to the RoI r_m that applies a series of convolutions followed by average pooling and two fully connected layers to output a vector of dimension $2N+3$. To ensure that these hyperparameters are positive, we found that applying a *sigmoid* layer and multiplying by 2 works well.

Chan-Vese Features Head: The energy in Eq. (1) encourages partitioning of the image based on the uniformity of image intensities I inside and outside of the object. However, image intensity values can be non-regular due to many factors such as lighting, different textures, motion blur, etc. Hence, we map the image intensities to a higher dimensional feature embedding space which is learned such that pixels of the same instance are close together in embedding space. We achieve this by passing the RoI r_m through a sequence of convolutions and upsampling layers to output a feature embedding $F(r_m)$ of dimension $C \times H \times W$. In our experiments we found $C = 64$ and $H = W = 112$ to be the most efficient in terms of memory for training and inference. The feature embedding $F(r_m)$ will replace the image intensities I in Eq. 1.

Chan-Vese Optimization as a Recurrent Net: After obtaining the initial TSDF, the set of hyperparameters, and the Chan-Vese feature map, we optimize the following deep energy functional E_m for each RoI r_m:

$$\begin{aligned} E_m(\phi, c_1, c_2) = \lambda_1(r_m) \int_{\Omega_m} \left\| F(r_m)(x) - c_1 \right\|^2 H(\phi(x)) dx \\ + \lambda_2(r_m) \int_{\Omega_m} \left\| F(r_m)(x) - c_2 \right\|^2 (1 - H(\phi(x))) dx \\ + \mu(r_m) \int_{\Omega_m} \delta_\varepsilon(\phi(x)) \left\| \nabla \phi(x) \right\| dx \end{aligned} \tag{6}$$

Note that the integration is over the image subset $\Omega_m \subset \Omega$ corresponding to r_m. We perform alternating optimization of ϕ and c_1, c_2. We implement the ϕ update step:

$$\phi_n = \phi_{n-1} + \Delta t_n(r_m) \frac{\partial \phi(\varepsilon_n(r_m))}{\partial t} \tag{7}$$

for $n = 1, \ldots, N$ as a set of feedforward layers with

$$\begin{aligned} \frac{\partial \phi(\varepsilon_n(r_m))}{\partial t} = \delta_{\varepsilon_n(r_m)}(\phi) \Big(\mu(r_m) \mathrm{div}\big(\frac{\nabla \phi}{\|\nabla \phi\|}\big) - \lambda_1(r_m) \left\| F(r_m) - c_1 \right\|^2 \\ + \lambda_2(r_m) \left\| F(r_m) - c_2 \right\|^2 \Big) \end{aligned} \tag{8}$$

In practice, we implement the gradient and the divergence term by using the Sobel operator [61] and the integration as a sum on the discrete image grid. At each update step, the constants c_1 and c_2 have closed-form updates as:

$$c_1(\phi) = \frac{\int_{\Omega_m} F(r_m)(x) H(\phi(x)) dx}{\int_{\Omega_m} H(\phi(x)) dx}, c_2(\phi) = \frac{\int_{\Omega_m} F(r_m)(x)(1 - H(\phi(x))) dx}{\int_{\Omega_m} (1 - H(\phi(x))) dx} \tag{9}$$

Here c_1 and c_2 are vectors where each element is the average of the corresponding feature embedding channel inside or outside of the object in the ROI respectively.

4.2 Learning

We train our model jointly in an end-to-end manner, as the Mask R-CNN backbone, the three extra heads, and the deep Chan-Vese recurrent network are all fully differentiable. We employ the standard regression and cross-entropy losses for the bounding box and classification components of both the RPN and the detection/classification heads of the backbone. For training the weights of the initial TSDF head, the hyperparameter head and the Chan-Vese features head, we apply the following loss, which is a mix of l_1 and binary cross-entropy BCE, to the initial and final TSDFs ϕ_0 and ϕ_N:

$$\ell_{TSDF}(\phi_{\{0,N\}}, \phi_{GT}, M_{GT}) = \left\|\phi_{\{0,N\}} - \phi_{GT}\right\|_1 + BCE(H_\varepsilon(\phi_{\{0,N\}}), M_{GT})$$

Here M_{GT} and ϕ_{GT} are the ground truth mask and TSDF targets. In order to apply BCE on ϕ_0 and ϕ_N, similar to [65] we map them to $[0, 1]$ with the soft Heaviside function and $\varepsilon = 0.1$. During backpropagation, the loss gradient from ϕ_N flows through the unrolled level set optimization and then through the Chan-Vese features head, the hyperparameter head, and the initial TSDF head.

5 Experimental Evaluation

In this section, we describe the datasets, implementation details and the metrics and compare our approach with the state-of-the-art. Next, we study the various aspects of our proposed approach through ablations.

Datasets: We evaluate our model on Cityscapes [15] and COCO [44] datasets. Cityscapes contains very precise annotations for 8 categories split into 2975 train, 500 validation and 1525 test images of resolution 1024 × 2048. The COCO dataset has 80 categories with 118k images in the `train2017` set for training and 5k images in the `val2017` set for evaluation. However, as demonstrated quantitatively by [25], COCO does not consistently provide accurate object annotations rendering mask quality evaluation of a method not indicative. As such, we follow the approach of [37] and also evaluate our model on the COCO sub-categories of the validation set of the LVIS dataset [25] with our model trained only on COCO. Note that LVIS re-annotates all the COCO validation images with high quality masks which makes it suitable for evaluating mask improvements. We follow this protocol since the LVIS dataset has more than 1000 categories and is designed for large vocabulary instance segmentation which is still in its infancy and an exciting topic for future research.

Table 1. Instance segmentation on Cityscapes val and test sets: This table shows our instance segmentation results on Cityscape on val and test. We report models trained on Cityscapes with and without COCO/Mapillary pre-training as well the methods that use horizontal flipping (F) or multiscale (MS) inference at test time.

	Training data	AP_{val}	AP_{test}	AP^{50}_{test}	person	rider	car	truck	bus	train	mcycle	bcycle
DWT [3]	fine	21.2	19.4	35.3	15.5	14.1	31.5	22.5	27.0	22.9	13.9	8.0
Kendall et al. [32]	fine	–	21.6	39.0	19.2	21.4	36.6	18.8	26.8	15.9	19.4	14.5
Arnab et al. [2]	fine	–	23.4	45.2	21.0	18.4	31.7	22.8	31.1	**31.0**	19.6	11.7
SGN [46]	fine+coarse	29.2	25.0	44.9	21.8	20.1	39.4	24.8	33.2	30.8	17.7	12.4
PolyRNN++ [1]	fine	–	25.5	45.5	29.4	21.8	48.3	21.2	32.3	23.7	13.6	13.6
Mask R-CNN [27]	fine	31.5	26.2	49.9	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0
BShapeNet+ [33]	fine	–	27.3	50.4	29.7	23.4	46.7	26.1	33.3	24.8	20.3	14.1
GMIS [48]	fine+coarse	–	27.3	45.6	31.5	25.2	42.3	21.8	37.2	28.9	18.8	12.8
Neven et al. [53]	fine	–	27.6	50.9	34.5	26.1	52.4	21.7	31.2	16.4	20.1	18.9
PANet [47]	fine	36.5	31.8	57.1	36.8	**30.4**	**54.8**	27.0	36.3	25.5	22.6	**20.8**
Ours	fine	**37.9**	**33.3**	**58.2**	**37.0**	29.2	54.6	**30.4**	**39.4**	30.2	**25.5**	20.3
AdaptIS [62] (F)	fine	36.3	32.5	52.5	31.4	29.1	50.0	31.6	41.7	**39.4**	24.7	12.1
SSAP [23] (MS+F)	fine	37.3	32.7	51.8	35.4	25.5	55.9	**33.2**	**43.9**	31.9	19.5	16.2
Pan-DL [12] (MS+F)	fine	38.5	34.6	57.3	34.3	28.9	55.1	32.8	41.5	36.6	26.3	21.6
Ours (MS+F)	fine	**40.0**	**35.8**	**61.2**	**40.5**	**31.7**	**56.9**	31.4	42.4	32.5	**28.6**	**22.2**
Mask R-CNN [27]	fine+COCO	36.4	32.0	58.1	34.8	27.0	49.1	30.1	40.9	30.9	24.1	18.7
BShapeNet+ [33]	fine+COCO	–	32.9	58.8	36.6	24.8	50.4	33.7	41.0	33.7	25.4	17.8
UPSNet [66]	fine+COCO	37.8	33.0	59.7	35.9	27.4	51.9	31.8	43.1	31.4	23.8	19.1
PANet [47]	fine+COCO	41.4	36.4	63.1	41.5	33.6	58.2	31.8	45.3	28.7	28.2	24.1
Pan-DL [12]	fine+MV	42.5	39.0	64.0	36.0	30.2	56.7	**41.5**	**50.8**	**42.5**	30.4	23.7
Polytransform [41]	fine+COCO	**44.6**	**40.1**	**65.9**	42.4	**34.8**	58.5	39.8	50.0	41.3	30.9	23.4
Ours (COCO)	fine+COCO	43.3	40.0	65.7	**43.4**	33.9	**59.0**	37.6	49.4	39.4	**32.5**	**24.9**

Implementation Details: For Cityscapes, we follow [27] and adopt multi-scale training where we resize the input image in a way that the length of the shorter edge is randomly sampled from [800, 1024]. We train the model on 8 GPUs for 24 K iterations with a learning rate of 0.01, decayed to 0.001 at 18 K iterations. We set the loss weights for the initial and final TSDF output to 1 and 5 in the multitask objective. For COCO, following [27], we train the model without multi-scaling on 16 GPUs for 90 K iterations with a learning rate of 0.02 decayed by a factor of 10 at 60 K and 80 K iterations. We set the loss weights for the initial and final TSDF output to 0.2 and 1 in the multitask objective. For both datasets, we set the weight decay as 0.0001, with mini-batch size of 8. We employ WideResNet-38 [60] on Cityscapes test set and ResNet-50 [28] in all the other experiments. For the level set optimization, we unroll the optimization for 3 steps. Finally, we simply apply the Heaviside function to the TSDF output to obtain a mask. Note that if we apply marching squares [49] to the final TSDF instead, we could obtain sub-pixel accuracy for the boundaries. However for simplicity and since the AP metric of COCO and Cityscapes requires binary masks for evaluation, we simply threshold our TSDFs using the heaviside function.

Evaluation Metrics: We report the standard AP metric of [44] on both Cityscapes and COCO. For LVIS, we report the federated average precision metric denoted by AP* [25] on the COCO subcategories.

Table 2. LevelSet R-CNN vs. Mask R-CNN: We report mask AP for both COCO and Cityscapes on the val set with `Resnet-50` backbone. We also report the federated AP, i.e. AP*, of the LVIS dataset with COCO subcategories trained only COCO. For our model we report both the initial and final mask results after optimization.

	Cityscapes AP	COCO AP	LVIS AP*
Mask R-CNN	32.3	33.8	35.6
Ours (Initial Mask)	35.4	33.7	35.8
Ours	**36.2**	**34.3**	**36.4**

Table 3. LevelSet R-CNN vs. deep level set methods on the val set of Cityscapes: Using level sets as a loss function (LS Loss) or using the geodesic level sets (DELSE).

	LS Loss [29]	DELSE [65]	DELSE [65] + HP head	Ours
Initial Mask AP	–	34.6	34.9	**35.4**
Final Mask AP	34.3	34.6	35	**36.2**

Cityscapes Test: We compare LevelSet R-CNN against published state-of-the-art (SOTA) methods on Cityscapes in Table 1. LevelSet R-CNN outperforms all previous methods that are trained on Cityscapes data without test time augmentations achieving a new state-of-the-art performance by 1.5 AP over PANet [47]. We also compare against models that adopt multiscale (MS) and horizontal flipping (F) at test time. We improve upon the state-of-the-art, Panoptic-Deeplab [12], by 1.2 AP. Next, we evaluate against models that pretrain on external datasets such as COCO [44] or Mapillary Vistas [52]. For a fair comparison, we follow the exact setting of Polytransform [41] which was the state-of-the-art at the time of submission. In particular, we use a `WideResNet-38` backbone with deformable convolutions [18] and PANet modifications [47]. We train on COCO for 270000 iterations with a learning rate of 0.02 decayed by a factor of 10 at 210000 and 250000 iterations on 16 GPUs. On Cityscapes, we finetuned for 6000 iterations on 8 GPUs with a learning rate of 0.01 decayed to 0.001 at 4000 iterations. As shown in Table 1, our performance is comparable with Polytransform.

AP Improvements Across Datasets: In Table 2, we compare Mask R-CNN with our initial and final mask outputs on the validation sets of Cityscapes, COCO and LVIS. All models employ the `Resnet-50` backbone. LevelSet R-CNN outperforms Mask R-CNN on all datasets. Note that while Levelset R-CNN was only trained on COCO and not with the precise boundaries of LVIS, it improves upon the baseline by 0.8 AP*. We also see an improvement of about 4 AP on Cityscapes.

Different Deep Level Set Formulations: To further justify our deep region based level set formulation, we compare with two different variations of level sets applied to the task of instance segmentation. [29] use the Chan-Vese energy as a

Table 4. Backproping through the initial mask from the final TSDF loss ℓ_{TSDF} on the val set of Cityscapes

	Detach ϕ_0	Backprop Thru ϕ_0
Initial Mask AP	33	**35.4**
Final Mask AP	34.2	**36.2**

Table 5. Boundary metric: on val set of Cityscapes: We evaluate the AF at thresholds of 1 and 2 pixels against Mask R-CNN across two backbones.

	Backbone	AF_1	AF_2
Mask R-CNN [27]	`Resnet-50`	40.2	57.6
Ours	`Resnet-50`	**45.8**	**63.1**
Mask R-CNN [27]	`WideResnet-38`	42.7	59.9
Ours	`WideResnet-38`	**46.8**	**64.6**

loss function for salient object detection. Here, we employ their loss for instance segmentation. In particular, we shift the mask output of Mask R-CNN by −0.5, apply the soft Heaviside and pass to the Chan-Vese energy loss function. In Table 3 we observe that LevelSet R-CNN improves the level set loss by about 2 AP. Next, we combine the deep edge based level set of [65], referred to as DELSE, with Mask R-CNN by changing the mask head to a TSDF head and adding two more heads: the velocity head that predicts the direction to the object boundaries and the modulation head which regulates the effect of the curvature term on the object boundaries. We evaluate in two settings: 1) Similar to their work, we use hand-tuned hyperpameters and use their exact loss functions, i.e., L_2 for the initial TSDF, class balanced cross entropy for the final TSDF and L_2 on angular domain. 2) To remove the effect of loss functions and hyperparameters choices and provide the most fair comparison, we add our hyperparameter head to their model and use our loss functions with the exception of having an extra loss function for the velocity head. In Table 3, we observe that LevelSet R-CNN has a 1.2 AP improvement over the DELSE formulation. Moreover, we obtain 0.8 AP improvement over the initial mask whereas DELSE gain is 0.1 AP.

Passing Gradients Through the Initial TSDF: Table 4 shows that by passing the gradient from ϕ_N through the initial TSDF head, we improve the AP of both the initial TSDF ϕ_0 and the final TSDF ϕ_N. As an alternative we could have detached ϕ_0 from the computation graph so that it does not take supervision from the final TSDF ϕ_N; Passing the gradient improves ϕ_0 by 2.4 AP and ϕ_N by 2 AP. Finally, we see that the AP of the initial TSDF when passing gradients is higher than the AP of the final TSDF when not passing the gradient by 1.2 AP. This suggests that the hyperparameter head, the Chan-Vese features head and the unrolled optimization, can also be used during training for improving the performance of the mask head and discarded during inference.

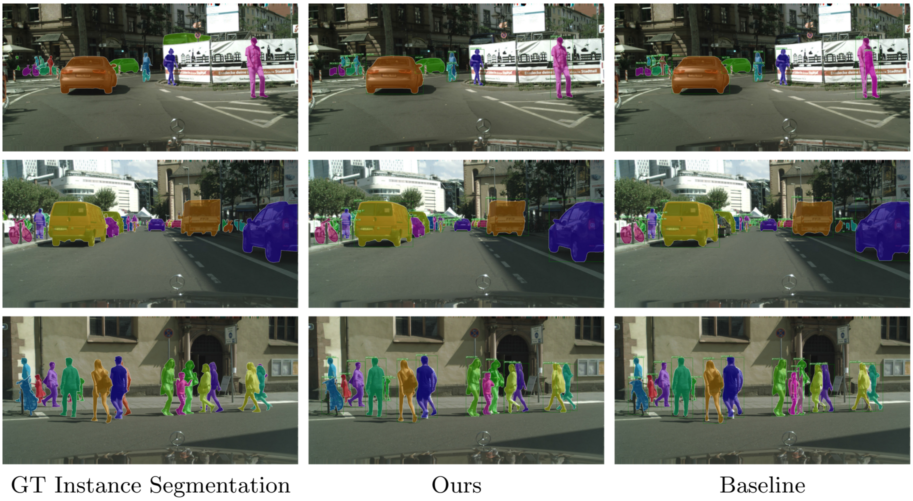

Fig. 2. We showcase qualitative instance segmentation results of our model on the Cityscapes validation set. We can see that our method produces masks with higher quality when bounding box results are similar.

Boundary Metric: In addition, to evaluate the capacity of our model in improving the boundaries of objects, we adapt the boundary metric of DAVIS [57] to our task. In particular, for a True Positive detection, we compute F1 between the prediction and ground truth boundary pixels at thresholds of 1 and 2 pixels. Similar to AP, we obtain the True Positives at IoUs in range [0.5, 0.95] at 0.05 increments. The F1s are averaged over all the classes and thresholds and are denoted by (AF_1) and (AF_2) for thresholds of 1 and 2 pixels away. In Table 5, we observe that our method is able to improve the boundaries of the objects by at least 4 AF at each threshold compared to the baseline across the two backbones `Resnet-50` and `WideResnet-38`.

Mask R-CNN with Different Training Targets: We modify the mask head of Mask R-CNN to output a TSDF instead of a binary mask and we train with ℓ_{TSDF} rather than BCE as loss function. To understand the dependence of the Mask R-CNN performance on this TSDF target ℓ_{TSDF}, we trained a model with only the mask head modification and without the other Chan-Vese components (i.e., the adaptive hyperparameter head and the deep Chan-Vese module and unrolled optimization). We obtain the same AP of 32.3 for model with ℓ_{TSDF} as the original Mask R-CNN. This indicates that the model improvements do not simply come from changing the loss of the mask head.

Effect of the Hyperparameter Head: To verify the importance of learning adaptive hyperparameters per object instance, we perform an ablation where we remove the hyperparameter head and just learn a set of global hyperparameters for the whole dataset. The adaptive hyper parameter head achieves 36.2 AP vs. the 35.4 AP of a global set giving a boost of 0.8 AP.

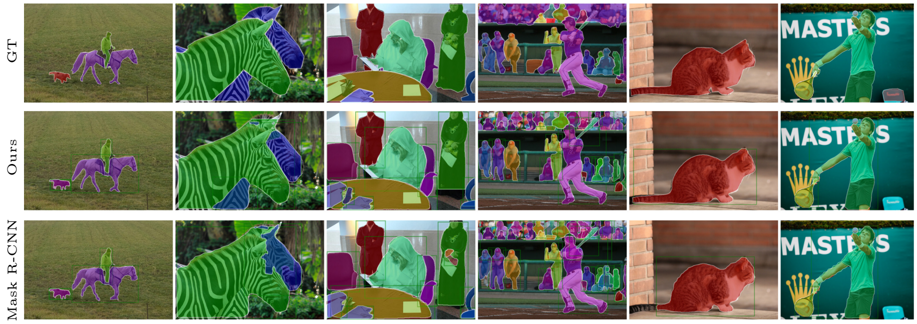

Fig. 3. We showcase qualitative instance segmentation results of our model on the COCO validation set.

Higher Resolution Mask R-CNN: We evaluate whether we could improve the performance of Mask R-CNN by just increasing the resolution of the mask head. We train Mask R-CNN on Cityscapes with `ResNet-50` backbone at 112×112 resolution which is the same resolution of the Chan-Vese features and our final TSDF ϕ_N. Interestingly the performance drops by 2.2 AP from 32.3. We hypothesize that by increasing the resolution, the ratio of non-boundary pixels vs. boundary pixels will become higher and they dominate the loss function gradients leading to worse masks. In our proposed method however, there is a global competition between the foreground/background regions to minimize the energy and hence we are able to increase the resolution.

Inference Time: LevelSet R-CNN with a `ResNet-50` runs on average at 182 ms vs. Mask R-CNN at 145 ms on GTX 1080 ti on images of dimension 1024×2048.

Qualitative Results: As shown in Figs. 2 and 3 we observe mask boundary and region improvements compared to the baseline.

6 Conclusion

In this paper, we proposed *LevelSet R-CNN* which combines the strengths of modern deep learning based Mask R-CNN and classical energy based Chan-Vese level set segmentation framework in an end-to-end manner. In particular, we utilize four heads based on FPN to obtain each detected object, an initial level set, deep robust feature representation for the Chan-Vese energy data terms and a set of instance dependent hyperparameters that balance the energy terms and schedule the optimization procedure. We demonstrated the effectiveness of our method on COCO and Cityscapes showing improvements on both datasets.

References

1. Acuna, D., Ling, H., Kar, A., Fidler, S.: Efficient interactive annotation of segmentation datasets with polygon-RNN++. In: CVPR (2018)

2. Arnab, A., Torr, P.H.S.: Pixelwise instance segmentation with a dynamically instantiated network. In: CVPR (2017)
3. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: CVPR (2017)
4. Brabandere, B.D., Neven, D., Gool, L.V.: Semantic instance segmentation with a discriminative loss function. In: CVPR (2017)
5. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: CVPR (2018)
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**(1), 61–79 (1997)
7. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Trans. Image Process. **10**(2), 266–277 (2001)
8. Chen, K., et al.: Hybrid task cascade for instance segmentation. In: CVPR (2019)
9. Chen, L.C., Hermans, A., Papandreou, G., Schroff, F., Wang, P., Adam, H.: Masklab: instance segmentation by refining object detection with semantic and direction features. In: CVPR (2018)
10. Chen, X., Girshick, R.B., He, K., Dollár, P.: TensorMask: a foundation for dense object segmentation. In: ICCV (2019)
11. Chen, X., Williams, B.M., Vallabhaneni, S.R., Czanner, G., Williams, R.S., Zheng, Y.: Learning active contour models for medical image segmentation. In: CVPR (2019)
12. Cheng, B., et al.: Panoptic-DeepLab: a simple, strong, and fast baseline for bottom-up panoptic segmentation. In: CVPR (2020)
13. Cheng, D., Liao, R., Fidler, S., Urtasun, R.: DARNet: deep active ray network for building segmentation. In: CVPR (2019)
14. Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP 2014 (2014)
15. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
16. Cremers, D., Rousson, M., Deriche, R.: A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. Int. J. Comput. Vis. **72**(2), 195–215 (2007)
17. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: CVPR (2015)
18. Dai, J., et al.: Deformable convolutional networks. In: ICCV (2017)
19. Dervieux, A., Thomasset, F.: A finite element method for the simulation of a Rayleigh-Taylor instability. In: Rautmann, R. (ed.) Approximation Methods for Navier-Stokes Problems. LNM, vol. 771, pp. 145–158. Springer, Heidelberg (1980). https://doi.org/10.1007/BFb0086904
20. Fathi, A., et al.: Semantic instance segmentation via deep metric learning. ArXiv (2017)
21. Fazeli, N., Oller, M., Wu, J., Wu, Z., Tenenbaum, J.B., Rodriguez, A.: See, feel, act: hierarchical learning for complex manipulation skills with multisensory fusion. Sci. Robot. (2019)
22. Fu, C.Y., Shvets, M., Berg, A.C.: RetinaMask: learning to predict masks improves state-of-the-art single-shot detection for free. ArXiv (2019)
23. Gao, N., et al.: SSAP: single-shot instance segmentation with affinity pyramid. In: ICCV (2019)
24. Girshick, R.B.: Fast R-CNN. In: ICCV (2015)

25. Gupta, A., Dollar, P., Girshick, R.: LVIS: a dataset for large vocabulary instance segmentation. In: CVPR (2019)
26. Gur, S., Shaharabany, T., Wolf, L.: End to end trainable active contours via differentiable rendering. In: ICLR (2020)
27. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: CVPR (2017)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2015)
29. Hu, P., Shuai, B., Liu, J., Wang, G.: Deep level sets for salient object detection. In: CVPR (2017)
30. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring R-CNN. In: CVPR (2019)
31. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vis. **1**(4), 321–331 (1988)
32. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: CVPR (2018)
33. Kim, H.Y., Kang, B.R.: BshapeNet: object detection and instance segmentation with bounding shape masks. Pattern Recogn. Lett. **131**, 449–455 (2020)
34. Kim, Y., Kim, S., Kim, T., Kim, C.: CNN-based semantic segmentation using level set loss. In: WACV (2019)
35. Kirillov, A., He, K., Girshick, R., Rother, C., Dollar, P.: Panoptic segmentation. In: CVPR (2019)
36. Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., Rother, C.: InstanceCut: from edges to instances with multicut. In: CVPR (2017)
37. Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: image segmentation as rendering. In: ECCV (2020)
38. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NeurIPS (2012)
39. Le, T.H.N., Quach, K.G., Luu, K., Duong, C.N., Savvides, M.: Reformulating level sets as deep recurrent neural network approach to semantic segmentation. IEEE Trans. Image Process. **27**(5), 2393–2407 (2018)
40. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: CVPR (2017)
41. Liang, J., Homayounfar, N., Ma, W.C., Xiong, Y., Hu, R., Urtasun, R.: PolyTransform: deep polygon transformer for instance segmentation. In: CVPR (2020)
42. Liang, X., Lin, L., Wei, Y., Shen, X., Yang, J., Yan, S.: Proposal-free network for instance-level object segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **40**(12), 2978–2991 (2018)
43. Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR (2016)
44. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
45. Ling, H., Gao, J., Kar, A., Chen, W., Fidler, S.: Fast interactive object annotation with curve-GCN. In: CVPR (2019)
46. Liu, S., Jia, J., Fidler, S., Urtasun, R.: SGN: sequential grouping networks for instance segmentation. In: ICCV (2017)
47. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR (2018)
48. Liu, Y., et al.: Affinity derivation and graph merge for instance segmentation. In: ECCV (2018)

49. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. In: SIGGRAPH (1987)
50. Ma, W.C., Wang, S., Hu, R., Xiong, Y., Urtasun, R.: Deep rigid instance scene flow. In: CVPR (2019)
51. Marcos, D., et al.: Learning deep structured active contours end-to-end. In: CVPR (2018)
52. Neuhold, G., Ollmann, T., Rota Bulò, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV (2017)
53. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In: CVPR, June 2019
54. Newell, A., Huang, Z., Deng, J.: Associative embedding: end-to-end learning for joint detection and grouping. In: NeurIPS (2017)
55. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: NeurIPS (2001)
56. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**(1), 12–49 (1988)
57. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
58. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR (2015)
59. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NeurIPS (2015)
60. Rota Bulò, S., Porzi, L., Kontschieder, P.: In-place activated batchnorm for memory-optimized training of DNNs. In: CVPR (2018)
61. Sobel, I.: An isotropic 3×3 image gradient operator. Presentation at Stanford A.I. Project 1968, February 2014
62. Sofiiuk, K., Barinova, O., Konushin, A.: Adaptis: adaptive instance selection network. In: ICCV (2019)
63. Tang, M., Valipour, S., Zhang, Z., Cobzas, D., Jagersand, M.: A deep level set method for image segmentation. In: Cardoso, M.J., et al. (eds.) DLMIA/ML-CDS -2017. LNCS, vol. 10553, pp. 126–134. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67558-9_15
64. Uhrig, J., Cordts, M., Franke, U., Brox, T.: Pixel-level encoding and depth layering for instance-level semantic labeling. In: GCPR (2016)
65. Wang, Z., Acuna, D., Ling, H., Kar, A., Fidler, S.: Object instance annotation with deep extreme level set evolution. In: CVPR (2019)
66. Xiong, Y., et al.: UPSNet: a unified panoptic segmentation network. In: CVPR (2019)
67. Xu, Y., et al.: Gland instance segmentation by deep multichannel neural networks. In: MICCAI (2016)
68. Yao, S., et al.: 3D-aware scene manipulation via inverse graphics. In: NeurIPS (2018)
69. Zhang, Z., Fidler, S., Urtasun, R.: Instance-level segmentation for autonomous driving with deep densely connected MRFs. In: CVPR (2016)
70. Zhang, Z., Schwing, A.G., Fidler, S., Urtasun, R.: Monocular object instance segmentation and depth ordering with CNNs. In: ICCV (2015)