Boundary-aware Instance Segmentation

Zeeshan Hayder[1,2], Xuming He[2,1]
[1]Australian National University & [2]Data61/CSIRO *

Mathieu Salzmann[3]
[3]CVLab, EPFL, Switzerland

Abstract

We address the problem of instance-level semantic segmentation, which aims at jointly detecting, segmenting and classifying every individual object in an image. In this context, existing methods typically propose candidate objects, usually as bounding boxes, and directly predict a binary mask within each such proposal. As a consequence, they cannot recover from errors in the object candidate generation process, such as too small or shifted boxes.

In this paper, we introduce a novel object segment representation based on the distance transform of the object masks. We then design an object mask network (OMN) with a new residual-deconvolution architecture that infers such a representation and decodes it into the final binary object mask. This allows us to predict masks that go beyond the scope of the bounding boxes and are thus robust to inaccurate object candidates. We integrate our OMN into a Multitask Network Cascade framework, and learn the resulting boundary-aware instance segmentation (BAIS) network in an end-to-end manner. Our experiments on the PASCAL VOC 2012 and the Cityscapes datasets demonstrate the benefits of our approach, which outperforms the state-of-the-art in both object proposal generation and instance segmentation.

1. Introduction

Instance-level semantic segmentation, which aims at jointly detecting, segmenting and classifying every individual object in an image, has recently become a core challenge in scene understanding [5, 23, 9]. Unlike its category-level counterpart, instance segmentation provides detailed information about the location, shape and number of individual objects. As such, it has many applications in diverse areas, such as autonomous driving [35], personal robotics [12] and plant analytics [30].

Existing approaches to multiclass instance segmentation typically rely on generic object proposals in the form of bounding boxes. These proposals can be learned [14, 21, 8] or sampled by sliding windows [25, 6], and greatly facilitate the task of identifying the different instances, may they be from the same category or different ones. Object segmentation is then achieved by predicting a binary mask within each box proposal, which can then be classified into a semantic category. This approach to segmentation, however, makes these methods sensitive to the quality of the bounding boxes; they cannot recover from errors in the object proposal generation process, such as too small or shifted boxes.

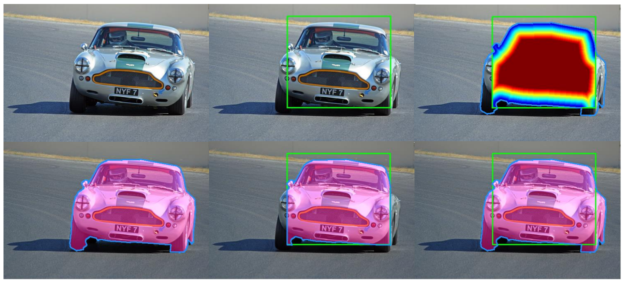

Figure 1. **Traditional instance segmentation vs our boundary based representation. Left:** Original image and ground-truth segmentation. **Middle:** Given a bounding box, traditional methods directly predict a binary mask, whose extent is therefore limited to that of the box and thus suffers from box inaccuracies. **Right:** We represent the object segment with a multi-valued map encoding the truncated minimum distance to the object boundary. This can be converted into a mask that goes beyond the bounding box, which makes our approach robust to box errors.

In this paper, we introduce a novel representation of object segments that is robust to errors in the bounding box proposals. To this end, we propose to model the shape of an object with a dense multi-valued map encoding, for every pixel in a box, its (truncated) minimum distance to the object boundary, or the fact that the pixel is outside the object. Object segmentation can then be achieved by converting this multi-valued map into a binary mask via the inverse distance transform [3, 18]. In contrast to existing methods discussed above, and as illustrated in Fig. 1, the resulting mask is *not* restricted to lie inside the bounding box; even when the box covers only part of the object, the distances to the boundary in our representation may correspond to an object segment that goes beyond the box's spatial extent.

*Data61/CSIRO is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the ARC through the ICT Centre of Excellence program.

To exploit our new object representation, we design an object mask network (OMN) that, for each box proposal, first predicts the corresponding pixel-wise multi-valued map, and then decodes it into the final binary mask, potentially going beyond the box itself. In particular, we discretize the truncated distances and encode them using a binary vector. This translates the prediction of the multi-valued map to a pixel-wise labeling task, for which deep networks are highly effective, and facilitates decoding the map into a mask. The first module of our network then produces multiple probability maps, each of which indicates the activation of one particular bit in this vector. We then pass these probability maps into a new residual-deconvolution network module that generates the final binary mask. Thanks to the deconvolution layers, our output is not restricted to lie inside the box, and our OMN is fully differentiable.

To tackle instance-level semantic segmentation, we integrate our OMN into the Multitask Network Cascade framework of [8], by replacing the original binary mask prediction module. As our OMN is fully differentiable, we can learn the resulting instance-level semantic segmentation network in an end-to-end manner. Altogether, this yields a *boundary-aware* instance segmentation (BAIS) network that is robust to noisy object proposals.

We demonstrate the effectiveness of our approach on PASCAL VOC 2012 [9] and the challenging Cityscapes [5] dataset. Our BAIS framework outperforms all the state-of-the-art methods on both datasets, by a considerable margin in the regime of high IOUs. Furthermore, an evaluation of our OMN on the task of object proposal generation on the PASCAL VOC 2012 dataset reveals that it achieves performance comparable to or even better than state-of-the-art methods, such as DeepMask [25] and SharpMask [26].

2. Related Work

Over the years, much progress has been made on the task of category-level semantic segmentation, particularly since the advent of Deep Convolutional Neural Networks (CNNs) [10, 24, 4]. Categorical labeling, however, fails to provide detailed annotations of individual objects, from which many applications could benefit. By contrast, instance-level semantic segmentation produces information about the identity, location, shape and class label of each individual object.

To simplify this challenging task, most existing methods first rely on detecting individual objects, for which a detailed segmentation is then produced. The early instances of this approach [32, 16] typically used pre-trained class-specific object detectors. More recently, however, many methods have proposed to exploit generic object proposals [1, 28], and postpone the classification problem to later stages. In this context, [14] makes use of Fast-RCNN boxes [11] and builds a multi-stage pipeline to extract features, classify and segment the object. This framework was improved by the development of Hypercolumn features [15] and the use of a fully convolutional network (FCN) to encode category-specific shape priors [21]. In [8], the Region Proposal Network of [28] was integrated into a multi-task network cascade (MNC) for instance semantic segmentation. Ultimately, all these methods suffer from the fact that they predict a binary mask within the bounding box proposals, which are typically inaccurate. By contrast, here, we introduce a boundary-aware OMN that lets us predict instance segmentations that go beyond the box's spatial extent. We show that integrating this OMN in the MNC framework outperforms the state-of-the-art instance-level semantic segmentation techniques.

Other methods have nonetheless proposed to bypass the object proposal step for instance-level segmentation. For example, the Proposal-free Network (PFN) of [22] predicts the number of instances and, at each pixel, a semantic label and the location of its enclosing bounding box. The results of this approach, however, strongly depend on the accuracy of the predicted number of instances. By contrast, [36] proposed to identify the individual instances based on their depth ordering. This was further extended in [35] via a deep densely connected Markov Random Field. It is unclear, however, how this approach handles the case where multiple instances are at roughly identical depths. To overcome this, the recent work of [33] uses an FCN to jointly predict depth, semantics and an instance-based direction encoding. This information is then used to generate instances via a template matching procedure. Unfortunately, this process involves a series of independent modules, which cannot be optimized jointly, thus yielding a potentially suboptimal solution. Finally, in [29], a recurrent neural network was proposed to segment an image instance-by-instance. This approach, however, essentially assumes that all the instances observed in the image belong to the same class.

Beyond instance-level semantic segmentation, many methods have been proposed to generate class-agnostic region proposals [1, 34, 20]. The most recent such approaches rely on deep architectures [25, 26]. In particular, the method of [6], in which an FCN computes a small set of instance-sensitive score maps that are assembled into object segment proposals, was shown to effectively improve instance-level semantic segmentation when incorporated in the MNC framework. Our experiments demonstrate that our OMN produces segments of a quality comparable to or even higher than these state-of-the-art methods. Furthermore, by integrating it in a complete instance-level semantic segmentation network, we also outperform the state-of-the-art on this task.

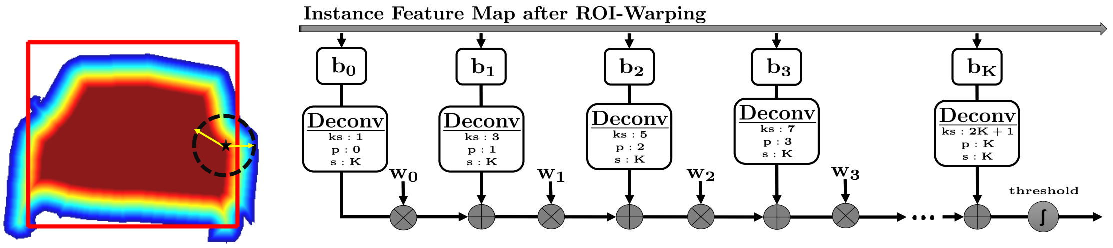

Figure 2. **Left:** Truncated distance transform. **Right:** Our deconvolution-based shape-decoding network. Each deconvolution has a specific kernel size (ks), padding (p) and stride (s). Here, K represents the number of binary maps.

3. Boundary-aware Segment Prediction

Our goal is to design an instance-level semantic segmentation method that is robust to the misalignment of initial bounding box proposals. To this end, we first introduce a novel object mask representation capable of capturing the overall shape or exact boundaries of an object. This representation, based on the distance transform, allows us to infer the complete shape of an object segment even when only partial information is available. We then construct a deep network that, given an input image, uses this representation to generate generic object segments that can go beyond the boundaries of initial bounding boxes.

Below, we first describe our object mask representation and object mask network (OMN). In Section 4, we show how our network can be integrated in a Multistage Network Cascade [8] to learn an instance-level semantic segmentation network in an end-to-end manner.

3.1. Boundary-aware Mask Representation

Given a window depicting a potentially partially-observed object, obtained from an image and a bounding box, we aim to produce a mask of the entire object. To this end, instead of directly inferring a binary mask, which would only represent the visible portion of the object, we propose to construct a pixel-wise, multi-valued map encoding the boundaries of the complete object by relying on the concept of distance transform [3]. In other words, the value at each pixel in our map represents either the distance to the nearest object boundary if the pixel is inside the object, or the fact that the pixel belongs to the background.

With varying window sizes and object shapes, the distance transform can produce a large range of different values, which would lead to a less invariant shape representation and complicate the training of our OMN in Section 3.2. Therefore, we normalize the windows to a common size and truncate the distance transform to obtain a restricted range of values. Specifically, let Q denote the set of pixels on the object boundary and outside the object. For every pixel p in the normalized window, we compute a truncated distance $D(p)$ to Q as

$$D(p) = \min\left(\min_{\forall q \in Q} \lceil d(p,q) \rceil, R\right) , \qquad (1)$$

where $d(p,q)$ is the spatial, Euclidean distance between pixel p and q, $\lceil x \rceil$ returns the integer nearest to but larger than x, and R is the truncation threshold, i.e., the largest distance we want to represent. We then directly use D as our dense object representation. Fig. 2 (Left) illustrates such a dense map for one object.

As an object representation, the pixel-wise map described above as several advantages over a binary mask that specifies the presence or absence of an object of interest at each pixel. First, the value at a pixel gives us information about the location of the object boundary, even if the pixel belongs to the interior of the object. As such, our representation is robust to partial occlusions arising from inaccurate bounding boxes. Second, since we have a distance value for every pixel, this representation is redundant, and thus robust to some degree of noise in the pixel-wise map. Importantly, predicting such a representation can be formulated as a pixel-wise labeling task, for which deep networks have proven highly effective.

To further facilitate this labeling task, we quantize the values in the pixel-wise map into K uniform bins. In other words, we encode the truncated distance for pixel p using a K-dimensional binary vector $b(p)$ as

$$D(p) = \sum_{n=1}^{K} r_n \cdot b_n(p), \quad \sum_{n=1}^{K} b_n(p) = 1 , \qquad (2)$$

where r_n is the distance value corresponding to the n-th bin. By this one-hot encoding, we have now converted the multi-value pixel-wise map into a set of K binary pixel-wise maps. This allows us to translate the problem of predicting the dense map to a set of pixel-wise binary classification tasks, which are commonly, and often successfully, carried out by deep networks.

Given the dense pixel-wise map of an object segment (or truly K binary maps), we can recover the complete object mask approximately by applying an inverse distance transform. Specifically, we construct the object mask by associating each pixel with a binary disk of radius $D(p)$. We then compute the object mask M by taking the union of all the disks. Let $T(p,r)$ denote the disk of radius r at pixel p. The object mask can then be expressed as

$$
\begin{aligned}
M &= \bigcup_p T(p, D(p)) = \bigcup_p T(p, \sum_{n=1}^{K} r_n \cdot b_n(p)) \\
&= \bigcup_{n=1}^{K} \bigcup_p T(p, r_n \cdot b_n(p)) = \bigcup_{n=1}^{K} T(\cdot, r_n) * B_n \ , \quad (3)
\end{aligned}
$$

where $*$ denotes the convolution operator, and B_n is the binary pixel-wise map for the n-th bin. Note that we make use of the property of the one-hot encoding in the derivation. Interestingly, the resulting operation consists of a series of convolutions, which will again become convenient when working with deep networks.

The rightmost column in Fig. 1 illustrates the behavior of our representation. In the top image, the value at each pixel represents the truncated distance to the instance boundary inside the bounding box. Although it does not cover the entire object, converting this dense map into a binary mask, yields the complete instance mask shown at the bottom.

3.2. Object Mask Network

We now turn to the problem of exploiting our boundary-aware representation to produce a mask for every object instance in an input image. To this end, we design a deep neural network that predicts K boundary-aware dense binary maps for every box in a set of bounding box proposals and decodes them into a full object mask via Eq. 3. In practice, we use the Region Proposal Network (RPN) [28] to generate the initial bounding box proposals. For each one of them, we perform a Region-of-Interest (RoI) warping of its features and pass the result to our network. This network consists of two modules described below.

Given the RoI warped features of one bounding box as input, the first module in our network predicts the K binary masks encoding our (approximate) truncated distance transform. Specifically, for the n-th binary mask , we use a fully connected layer with a sigmoid activation function to predict a pixel-wise probability map that approximates B_n.

Given the K probability maps, we design a new residual deconvolution network module to decode them into a binary object mask. Our network structure is based on the observation that the morphology operator in Eq. 3 can be implemented as a series of deconvolutions with fixed weights but different kernel and padding sizes, as illustrated in Fig. 2 (Right). We then approximate the union operator with a series of weighted summation layers followed by a sigmoid activation function. The weights in the summation layers are learned during training. To accommodate for the different sizes of the deconvolution filters, we upsample the output of the deconvolution corresponding to a smaller value of r_n in the network before each weighted summation. We use a fixed stride value of K for this purpose.

Our OMN is fully differentiable, and the output of the decoding module can be directly compared to the ground truth at a high resolution using a cross-entropy loss. This allows us to train our OMN in an end-to-end fashion, including the initial RPN, or, as discussed in Section 4, to integrate it with a classification module to perform instance-level semantic segmentation.

4. Learning Instance Segmentation

We now introduce our approach to tackling instance-level semantic segmentation with our OMN. To this end, we construct a Boundary-Aware Instance Segmentation (BAIS) network by integrating our object mask network into a Multistage Network Cascade (MNC) [8]. Since our OMN module is differentiable, we can train the entire instance segmentation network in an end-to-end manner. Below, we first describe the overall network architecture, and then discuss our end-to-end training procedure and inference at test time.

4.1. BAIS Network

Our boundary-aware instance segmentation network follows a structure similar to that of the MNC. Specifically, our segmentation network consists of three sub-networks, corresponding to the tasks of bounding box proposal generation, object mask prediction and object classification. The first module consists of a deep CNN (in practice, the VGG16 [31] architecture) to extract a feature representation from an input image, followed by an RPN [28], which generates a set of bounding box proposals. After RoI warping, we pass each proposal through our OMN to produce a segment mask. Finally, as in the original MNC network, mask features are computed by using the predicted mask in a feature masking layer and concatenated with bounding box features. The resulting representation is then fed into the third sub-network, which consists of a single fully-connected layer for classification and bounding-box regression. The overall architecture of our BAIS network is illustrated in Fig. 3.

Multi-stage Boundary-aware Segmentation Network. Following the strategy of [8], we extend the BAIS network described above, which can be thought of as a 3-stage cascade, to a 5-stage cascade. The idea, here, is to refine the initial set of bounding box proposals, and thus the predicted segments, based on the output of our OMN. As illustrated in Fig. 3 (Right), the first three stages consist of the model described above, that is the VGG16 convolutional layers, RPN, OMN, classification module and bounding-box prediction. We then make use of the prediction offset generated by the bounding-box regression part of the third stage to refine the initial boxes. These new boxes act as input, via RoI warping, to the fourth-stage, which corresponds to a second OMN. Its output is then used in the last stage in conjunction

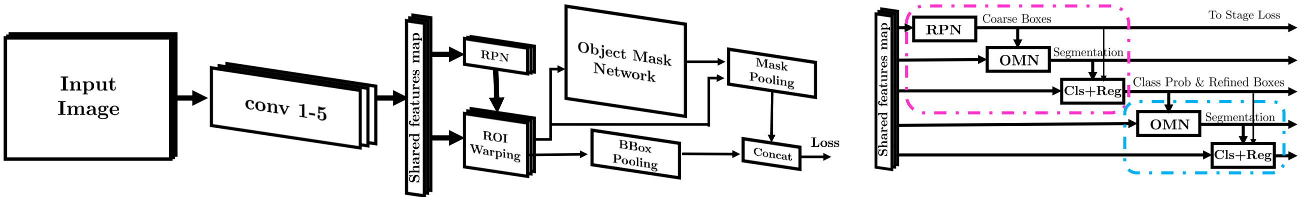

Figure 3. **Left:** Detailed architecture of our boundary-aware instance segmentation network. An input image first goes through a series of convolutional layers, followed by an RPN to generate bounding box proposals. After RoI warping, each proposal passes through our OMN to obtained a binary mask that can go beyond the box's spatial extent. Mask features are then extracted and used in conjunction with bounding-box features for classification purpose. During training, our model makes use of a multi-task loss encoding bounding box, segmentation and classification errors. **Right:** 5-stage BAIS network. The first three stages correspond to the model on the left. The five-stage model then concatenates an additional OMN and classification module to these three stages. The second OMN takes as input the classification score and refined box from the previous stage, and outputs a new segmentation with a new score obtained via the second classification module. The weights of the OMN and classification modules in both stages are shared.

with the refined boxes for classification purpose. In this 5-stage cascade, the weights of the two OMN and of the two classification modules are shared.

4.2. Network Learning and Inference

Our BAIS network is fully differentiable, and we therefore train it in an end-to-end manner. To this end, we use a multi-task loss function to account for bounding box, object mask and classification errors. Specifically, we use the softmax loss for the RPN and for classification, and the binary cross-entropy loss for the OMN. In our five-stage cascade, the bounding box and mask losses are computed after the third and fifth stages, and we use the smooth L_1 loss for bounding-box regression.

We minimize the resulting multi-task, multi-stage loss over all parameters jointly using stochastic gradient descent (SGD). Following [8, 6, 11], we rely on min-batches of 8 images. As in [8, 28, 11], we resize the images such that the shorter side has 600 pixels. The VGG16 network in our first module was pre-trained on ImageNet. The other weights are initialized randomly from a zero-mean Gaussian distribution with std 0.01. We then train our model for 20k iterations with a learning rate of 0.001, and 5k iterations with a reduced learning rate of 0.0001.

The first module in our network first generates $\sim$12k bounding boxes, which are pruned via non-maximum suppression (NMS). As in [8], we use an NMS threshold of 0.7, and finally keep the top 300 bounding box proposals. In our OMN, we use $K = 5$ probability maps to encode the (approximate) truncated distance transform. After decoding these maps via Eq. 3, we make use of a threshold of 0.4 to obtain a binary mask. This mask is then used to pool the features, and we finally obtain the semantic label via the classification module.

At test time, our BAIS network takes an input image and first computes the convolutional feature maps. The RPN module then generates 300 bounding box proposals and our OMN module predicts the corresponding object masks. These masks are categorized according to the class scores and a class-specific non-maximum suppression is applied with an IoU threshold of 0.5. Finally, we apply the in-mask voting scheme of [8] to each category independently to further refine the instance segmentations.

5. Experiments

In this section, we demonstrate the effectiveness of our method on instance-level semantic segmentation and segment proposal generation. We first discuss the former, which is the main focus of this work, and then turn to the latter. In both cases, we compare our approach to the state-of-the-art methods in each task.

Datasets and setup. To evaluate our approach, we make use of two challenging, standard datasets with multiple instances from a variety of object classes, *i.e.*, Pascal VOC 2012 and Cityscapes.

The Pascal VOC 2012 dataset [9] comprises 20 object classes with instance-level ground-truth annotations for 5623 training images and 5732 validation images. We used the instance segmentations of [13] for training and validation. We used all the training images to learn our model, but, following the protocols used in [14, 15, 7, 8, 6], used only the validation dataset for evaluation. Following standard practice, we report the mean Average Precision (mAP) using IoU thresholds of 0.5 and 0.7 for instance semantic segmentation, and the Average Recall (AR) for different number and sizes of boxes for segment proposal generation.

The Cityscapes dataset [5] consists of 9 object categories for instance-level semantic labeling. This dataset is very challenging since each image can contain a much larger number of instances of each class than in Pascal VOC, most of which are very small. It comprises 2975 training images from 18 cities, 500 validation images from 3 cities and 1525 test images from 6 cities. We only used the training dataset for training, and the test dataset to evaluate our method's performance on the online test-server. Following the Cityscapes dataset guidelines, we computed the average precision (AP) for each class by averaging it across a range of overlap thresholds. We report the mean average precision (mAP) using an IoU threshold of 0.5, as well as mAP100m and mAP50m, where the evaluation is restricted to objects within 100 meters and 50 meters, respectively.

5.1. Instance-level Semantic Segmentation

We first present our results on the task of instance-level semantic segmentation, which is the main focus of this paper. We report results on the two datasets discussed above. In both cases, we restricted the number of proposals to 300. For our 5-stage model, this means 300 after the first RPN and 300 after bounding-box refinement.

5.1.1 Results on VOC 2012

Let us first compare the results of our Boundary-aware Instance Segmentation (BAIS) network with the state-of-the-art approaches on Pascal VOC 2012. These baselines include the SDS framework of [14], the Hypercolumn representation of [15], the InstanceFCN method of [6] and the MNC framework of [8]. In addition to this, we also report the results obtained by a Python re-implementation of the method in [8], which we refer to as MCN-new. The results of this comparison are provided in Table 1. Note that our approach outperforms all the baselines, by a considerable margin in the case of a high IOU threshold. Note also that our approach is competitive in terms of runtime. Importantly, the comparison with BAIS-inside BBox, which restricts our masks to the spatial extent of the bounding boxes clearly evidences the importance of allowing the masks to go beyond the boxes' extent.

Following the evaluation of MNC in [8], we also study the influence of the number of stages in our model. We therefore learned different versions of our model using either our three-stage or five-stage cascade. At test time, because of parameter sharing across the stages, both versions are tested following a 5-stage procedure. The results of these different training strategies, for both MNC and our approach, are shown in Table 2. Note that, while our model trained with five-stages achieves the best results, our three-stage model still outperforms the two MNC baselines.

VOC 2012 (val)	mAP (0.5)	mAP (0.7)	time/img (s)
SDS [14]	49.7	25.3	48
PFN [22]	58.7	42.5	~ 1
Hypercolumn [15]	60.0	40.4	>80
InstanceFCN [6]	61.5	43.0	1.50
MNC [8]	63.5	41.5	0.36
MNC-new	65.01	46.23	0.42
BAIS - insideBBox (ours)	64.97	44.58	0.75
BAIS - full (ours)	**65.69**	**48.30**	0.78

Table 1. **Instance-level semantic segmentation on Pascal VOC 2012.** Comparison of our method with state-of-the-art baselines. The results of [14, 15] are reproduced from [8].

VOC 2012 (val)	Training	Testing	mAP (0.5)	mAP (0.7)
MNC [8]	3 stage	5 stage	62.6	-
MNC-new	5 stage	5 stage	65.01	46.23
BAIS - full (ours)	3 stage	5 stage	65.51	47.13
BAIS - full (ours)	5 stage	5 stage	**65.69**	**48.30**

Table 2. **Influence of the number of stages during training.** Whether trained using 3 stages or 5, our approach outperforms both MNC baselines.

A detailed comparison with MNC [8] including results for all the classes is provided in the supplementary material.

5.1.2 Results on Cityscapes

We now turn to the Cityscapes dataset. In Table 3, we first report the results obtained from the online evaluation server on the test data, which is not publicly available. Note that our approach outperforms all the baselines significantly on all the metrics. In Table 4, we provide a detailed comparison of our approach and the best performing baseline (DWT) in terms of AP(100m) and AP(50m), respectively. Note that we outperform this method on most classes.

Additionally, we also compare our approach with MNC-new on the validation data. In this case, both models were trained using the training data only. For MCN, we used the same image size, RPN batch size, learning rate and number of iterations as for our model. Both models were trained using 5 stages. Table 5 shows that, again, our model outperforms this baseline, thus demonstrating the benefits of allowing the masks to go beyond the box proposals.

Cityscapes (test)	AP	AP (50%)	AP (100m)	AP (50m)
Instance-level Segmentation of Vehicles by Deep Contours [17]	2.3	3.7	3.9	4.9
R-CNN + MCG convex hull [5]	4.6	12.9	7.7	10.3
Pixel-level Encoding for Instance Segmentation [33]	8.9	21.1	15.3	16.7
RecAttend [27]	9.5	18.9	16.8	20.9
InstanceCut [19]	13.0	27.9	22.1	26.1
DWT [2]	15.6	30.0	26.2	31.8
BAIS - full (ours)	**17.4**	**36.7**	**29.3**	**34.0**

Table 3. **Instance-level semantic segmentation on Cityscapes.** We compare our method with the state-of-the-art baselines on the Cityscapes test set. These results were obtained from the online evaluation server.

Cityscapes (test)	person	rider	car	truck	bus	train	motorcycle	bicycle	AP (50m)
DWT	27.0	20.4	57.0	**39.6**	**51.3**	37.9	12.8	8.1	31.8
BAIS - full (ours)	**31.5**	**23.4**	**63.1**	32.2	50.5	**40.4**	**16.5**	**14.6**	**34.0**

Cityscapes (test)	person	rider	car	truck	bus	train	motorcycle	bicycle	AP (100m)
DWT	27.0	19.8	52.8	**29.0**	36.4	25.1	11.7	7.8	26.2
BAIS - full (ours)	**30.3**	**22.7**	**58.2**	24.9	**38.6**	**29.9**	**15.3**	**14.3**	**29.3**

Table 4. **Detailed comparison with DTW: Top:** AP(50m), **Bottom:** AP(100m). Note that our approach outperforms this baseline on all the classes except truck for the Cityscapes test dataset.

Cityscapes (val)	IoU	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
MNC-new	0.5	23.25	25.19	43.26	31.65	50.99	42.51	14.00	17.53	31.05
BAIS - full (ours)	0.5	23.30	25.67	43.19	33.01	54.36	44.87	15.95	18.84	**32.40**
MNC-new	0.7	9.09	1.86	34.81	24.46	39.08	33.33	1.98	4.55	18.64
BAIS - full (ours)	0.7	9.09	2.53	35.05	25.75	39.35	33.04	2.73	5.30	**19.10**

Table 5. **Comparison with MNC-new on the Cityscapes validation data.** Note that our approach outperforms this baseline, thus showing the importance of allowing the masks to go beyond the box proposals.

In Fig. 4, we provide some qualitative results of our approach on Cityscapes. Note that we obtain detailed and accurate segmentations, even in the presence of many instances in the same image. Some failure cases are shown in Fig. 5. These failures typically correspond to one instance being broken into several ones.

5.2. Segment Proposal Generation

As a second set of experiments, we evaluate the effectiveness of our object mask network (OMN) at generating high-quality segment proposals. To this end, we made use of the 5732 Pascal VOC 2012 validation images with ground-truth from [13], and compare our approach with the state-of-the-art segmentation proposal generation methods according to the criteria of [14, 23]. In particular, we report the results of MCG [1], Deep-Mask [25] and Sharp-Mask [26] using the publicly available pre-computed segmentation proposals. We also report the results of MNC by reproducing them from [8], since these values were slightly better than those obtained from the publicly available segments. For our method, since the masks extend beyond the bounding box, the scores coming from the RPN, which correspond to the boxes, are ill-suited. We therefore learned a scoring function to re-rank our proposals. For the comparison to be fair, we also learned a similar scoring function for the MNC proposals. We refer to this baseline as MNC+score.

The results of our comparison are provided in Table 6. Our approach yields state-of-the-art results when considering 10 or 100 proposals. For 1000, SharpMask yields slightly better AR than us. Note, however, that, in practice, it is not always possible to handle 1000 proposals in later processing stages, and many instance-level segmentation methods only consider 100 or 300, which is the regime where our approach performs best. In Fig. 6, we report recall vs IOU threshold for all methods. Interestingly, even for 1000 segmentation proposals, our results outperform most of the baselines at high IOU thresholds. We refer the reader to the supplementary material for a comparison of the methods across different object sizes.

PASCAL VOC 2012	AR@10	AR@100	AR@1000
Selective Search [34]	7.0	23.5	43.3
MCG [1]	18.9	36.8	49.5
Deep-Mask [25]	30.3	45.0	52.6
Sharp-Mask [26]	33.3	48.8	**56.5**
MNC [8]	33.4	48.5	53.8
InstanceFCN [6]	38.9	49.7	52.6
MNC+score	45.7	49.1	52.5
OMN (ours)	**47.8**	**51.8**	54.7

Table 6. **Evaluation of our OMN on the PASCAL VOC 2012 validation set.** We compare our method with state-of-the-art segmentation proposal baselines according to the criteria of [14, 23]. Note that our approach outperforms the state-of-the-art methods for the top 10 and 100 segmentation proposals, which correspond to the most common scenarios when later processing is involved, e.g., instance level segmentation.

6. Conclusion

In this paper, we have introduced a distance transform-based mask representation that allows us to predict instance segmentations beyond the limits of initial bounding boxes. We have then shown how to infer and decode this representation with a fully-differentiable Object Mask Network (OMN) relying on a residual-deconvolutional architecture. We have then employed this OMN to develop a Boundary-Aware Instance Segmentation (BAIS) network. Our experiments on Pascal VOC 2012 and Cityscapes have demonstrated that our BAIS network outperforms the state-of-the-art instance-level semantic segmentation methods. In the future, we intend to replace the VGG16 network we rely on with deeper architectures, such as residual networks, to further improve the accuracy of our framework.

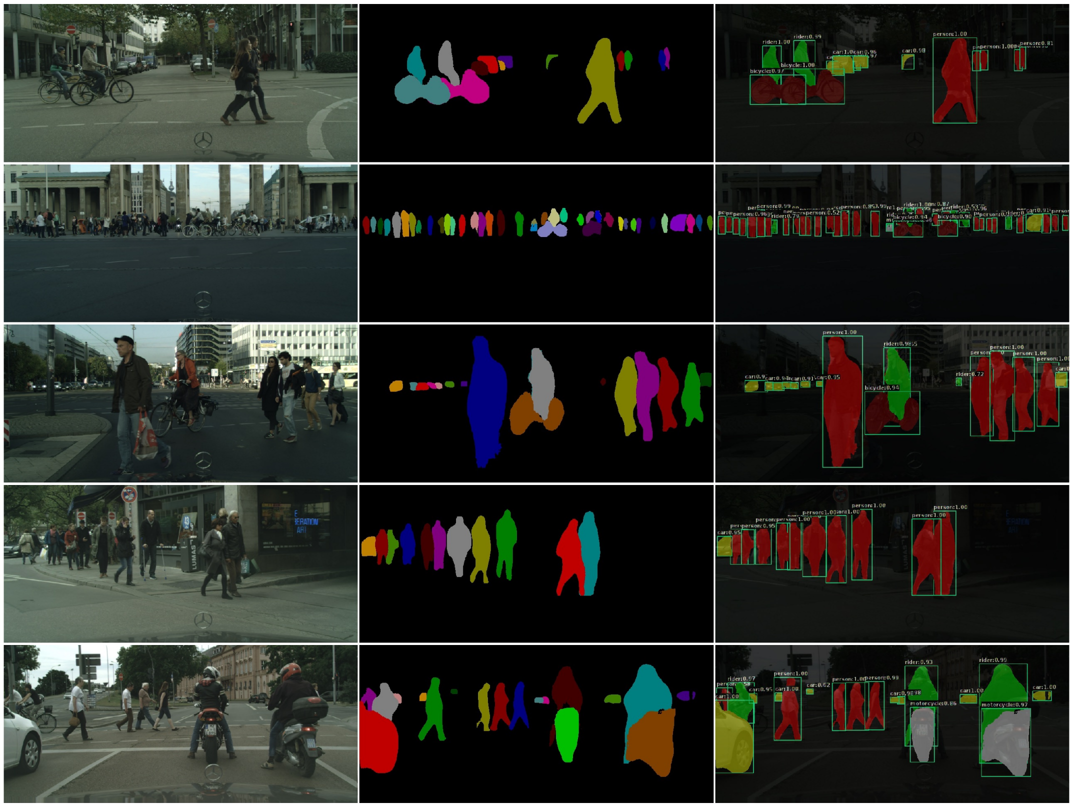

Figure 4. **Qualitative results on Cityscapes.** From left to right, we show the input image, our instance level segmentations and the segmentations projected onto the image with class labels. Note that our segmentations are accurate despite the presence of many instances.

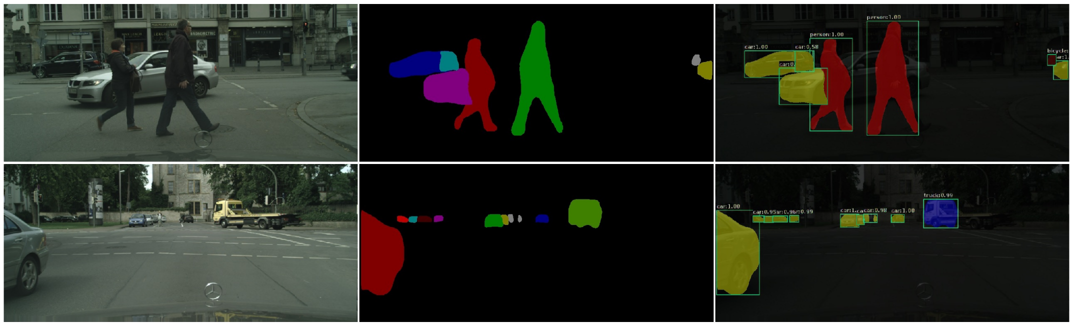

Figure 5. **Failure cases.** The typical failures of our approach correspond to cases where one instance is broken into multiple ones.

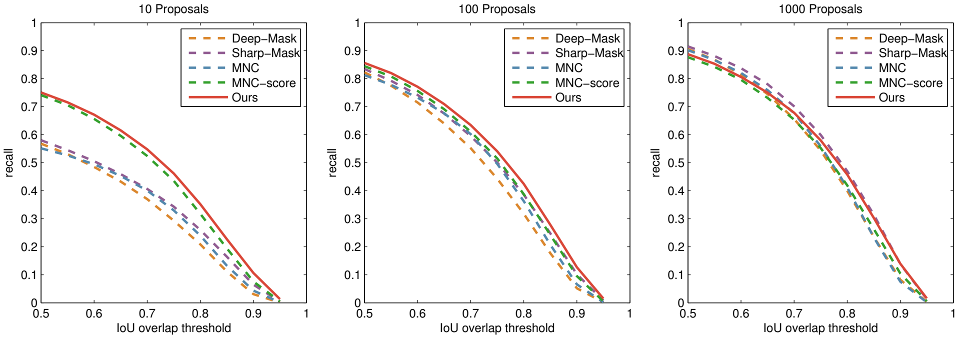

Figure 6. **Recall v.s. IoU threshold on Pascal VOC 2012.** The curves were generated using the highest-scoring 10, 100 and 1000 segmentation proposals, respectively. In each plot, the solid line corresponds to our OMN results. Note that we outperform the baselines when using the top 10 and 100 proposals. For 1000, our approach still yields state-of-the-art results at high IoU thresholds.

References

[1] P. Arbelaez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 2, 7

[2] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017. 6

[3] G. Borgefors. Distance transformations in digital images. *Computer vision, graphics and image processing*, 1986. 1, 3

[4] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2

[5] 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. 1, 2, 5, 6

[6] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *ECCV*, 2016. 1, 2, 5, 6, 7

[7] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 5

[8] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 1, 2, 3, 4, 5, 6, 7

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 1, 2, 5

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE TPAMI*, 2013. 2

[11] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 2, 5

[12] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, 2014. 1

[13] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 5, 7

[14] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 1, 2, 5, 6, 7

[15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 2, 5, 6

[16] X. He and S. Gould. An exemplar-based crf for multi-instance object segmentation. In *CVPR*, 2014. 2

[17] R. M. Jan van den Brand, Matthias Ochs. Instance-level segmentation of vehicles using deep contours. In *Workshop on Computer Vision Technologies for Smart Vehicle, in ACCV*, 2016. 6

[18] R. Kimmel, N. Kiryati, and A. M. Bruckstein. Sub-pixel distance maps and weighted distance transforms. *Journal of Mathematical Imaging and Vision*, 1996. 1

[19] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. InstanceCut: from edges to instances with multicut. In *CVPR*, 2017. 6

[20] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014. 2

[21] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. In *CVPR*, 2016. 1, 2

[22] X. Liang, Y. Wei, X. Shen, J. Yang, L. Lin, and S. Yan. Proposal-free network for instance-level object segmentation. *CoRR*, abs/1509.02636, 2015. 2, 6

[23] T. Lin, M. Maire, S. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 1, 7

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[25] P. O. Pinheiro, R. Collobert, and P. Dollr. Learning to segment object candidates. In *NIPS*, 2015. 1, 2, 7

[26] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollr. Learning to refine object segments. In *ECCV*, 2016. 2, 7

[27] M. Ren and R. S. Zemel. End-to-end instance segmentation and counting with recurrent attention. In *CVPR*, 2017. 6

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 4, 5

[29] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *ECCV*, 2016. 2

[30] H. Scharr, M. Minervini, A. Fischbach, and S. A. Tsaftaris. Annotated image datasets of rosette plants. In *ECCV*, 2014. 1

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4

[32] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014. 2

[33] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *GCPR*, 2016. 2, 6

[34] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013. 2, 7

[35] Z. Zhang, S. Fidler, and R. Urtasun. Instance-Level Segmentation with Deep Densely Connected MRFs. In *CVPR*, 2016. 1, 2

[36] Z. Zhang, A. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *ICCV*, 2015. 2