Pixelwise Instance Segmentation with a Dynamically Instantiated Network

Anurag Arnab and Philip H.S. Torr
University of Oxford
{anurag.arnab, philip.torr}@eng.ox.ac.uk

Abstract

Semantic segmentation and object detection research have recently achieved rapid progress. However, the former task has no notion of different instances of the same object, and the latter operates at a coarse, bounding-box level. We propose an Instance Segmentation system that produces a segmentation map where each pixel is assigned an object class and instance identity label. Most approaches adapt object detectors to produce segments instead of boxes. In contrast, our method is based on an initial semantic segmentation module, which feeds into an instance subnetwork. This subnetwork uses the initial category-level segmentation, along with cues from the output of an object detector, within an end-to-end CRF to predict instances. This part of our model is dynamically instantiated to produce a variable number of instances per image. Our end-to-end approach requires no post-processing and considers the image holistically, instead of processing independent proposals. Therefore, unlike some related work, a pixel cannot belong to multiple instances. Furthermore, far more precise segmentations are achieved, as shown by our substantial improvements at high AP^r thresholds.

1. Introduction

Semantic segmentation and object detection are well-studied scene understanding problems, and have recently witnessed great progress due to deep learning [21, 12, 6]. However, semantic segmentation – which labels every pixel in an image with its object class – has no notion of different instances of an object (Fig. 1). Object detection does localise different object instances, but does so at a very coarse, bounding-box level. Instance segmentation localises objects at a pixel level, as shown in Fig. 1, and can be thought of being at the intersection of these two scene understanding tasks. Unlike the former, it knows about different instances of the same object, and unlike the latter, it operates at a pixel level. Accurate recognition and localisation of objects enables many applications, such as autonomous driving [8], image-editing [46] and robotics [16].

Many recent approaches to instance segmentation are based on object detection pipelines where objects are first localised with bounding boxes. Thereafter, each bounding box is refined into a segmentation [18, 19, 27, 32, 26]. Another related approach [11, 49] is to use segment-based region proposals [9, 35, 36] instead of box-based proposals. However, these methods do not consider the entire image, but rather independent proposals. As a result, occlusions between different objects are not handled. Furthermore, many of these methods cannot easily produce segmentation maps of the image, as shown in Fig. 1, since they process numerous proposals independently. There are typically far more proposals than actual objects in the image, and these proposals can overlap and be assigned different class labels. Finally, as these methods are based on an initial detection step, they cannot recover from false detections.

Our proposed method is inspired by the fact that instance segmentation can be viewed as a more complex form of semantic segmentation, since we are not only required to label the object class of each pixel, but also its instance identity. We produce a pixelwise segmentation of the image, where each pixel is assigned both a semantic class and instance label. Our end-to-end trained network, which outputs a variable number of instances per input image, begins with an initial semantic segmentation module. The following, dynamic part of the network, then uses information from an object detector and a Conditional Random Field (CRF) model to distinguish different instances. This approach is robust to false-positive detections, as well as poorly localised bounding boxes which do not cover the entire object, in contrast to detection-based methods to instance segmentation. Moreover, as it considers the entire image when making predictions, it attempts to resolve occlusions between different objects and can produce segmentation maps as in Fig. 1 without any post-processing.

Furthermore, we note that the Average Precision (AP) metric [13] used in evaluating object detection systems, and its AP^r variant [18] used for instance segmentation, considers individual, potentially overlapping, object predictions in isolation, as opposed to the entire image. To evaluate methods such as ours, which produce complete segmentation maps and reason about occlusions, we also evaluate using the "Matching Intersection over Union" metric.

Our system, which is based on an initial semantic segmentation subnetwork, produces sharp and accurate in-

(a) Object Detection (b) Semantic Segmentation (c) Instance Segmentation

Figure 1. Object detection (a) localises the different people, but at a coarse, bounding-box level. Semantic segmentation (b) labels every pixel, but has no notion of instances. Instance segmentation (c) labels each pixel of each person uniquely. Our proposed method jointly produces both semantic and instance segmentations. Our method uses the output of an object detector as a cue to identify instances, but is robust to false positive detections, poor bounding box localisation and occlusions. Best viewed in colour.

stance segmentations. This is reflected by the substantial improvements we achieve over state-of-the-art methods at high AP^r thresholds on the Pascal VOC and Semantic Boundaries datasets. Furthermore, our network improves on the semantic segmentation task while being trained for the related task of instance segmentation.

2. Related Work

An early work on instance segmentation was by Winn and Shotton [44]. A per-pixel unary classifier was trained to predict parts of an object. These parts were then encouraged to maintain a spatial ordering, that is characteristic of an instance, using asymmetric pairwise potentials in a Conditional Random Field (CRF).

However, instance segmentation has become more common after the "Simultaneous Detection and Segmentation" (SDS) work of Hariharan *et al*. [18]. This system was based on the R-CNN pipeline [15]: Region proposals, generated by the method of [1], were classified into object categories with a Convolutional Neural Network (CNN) before applying bounding-box regression as post-processing. A class-specific segmentation was then performed in this bounding box to simultaneously detect and segment the object. Numerous works [19, 7, 26] have extended this pipeline. However, approaches that segment instances by refining detections [18, 19, 7, 10, 26] are inherently limited by the quality of the initial proposals. This problem is exacerbated by the fact that this pipeline consists of several different modules trained with different objective functions. Furthermore, numerous post-processing steps such as "superpixel projection" and rescoring are performed. Dai *et al*. [11] addressed some of these issues by designing one end-to-end trained network that generates box-proposals, creates foreground masks from these proposals and then classifies these masks. This network can be seen as an extension of the end-to-end Faster-RCNN [37] detection framework, which generates box-proposals and classifies them. Additionally, Liu *et al*. [32] formulated an end-to-end version of the SDS network [18], whilst [27] iteratively refined object proposals.

On a separate track, algorithms have also been developed that do not require object detectors. Zhang *et al*. [50, 51] segmented car instances by predicting the depth ordering of each pixel in the image. Unlike the previous detection-based approaches, this method reasoned globally about all instances in the image simultaneously (rather than individual proposals) with an MRF-based formulation. However, inference of this graphical model was not performed end-to-end as shown to be possible in [53, 2, 4, 29]. Furthermore, although this method does not use object detections, it is trained with ground truth depth and assumes a maximum of nine cars in an image. Predicting all the instances in an image simultaneously (rather than classifying individual proposals) requires a model to be able to handle a variable number of output instances per image. As a result, [38] proposed a Recurrent Neural Network (RNN) for this task. However, this model was only for a single object category. Our proposed method not only outputs a variable number of instances, but can also handle multiple object classes.

Liang *et al*. [28] developed another proposal-free method based on the semantic segmentation network of [5]. The category-level segmentation, along with CNN features, was used to predict instance-level bounding boxes. The number of instances of each class was also predicted to enable a final spectral clustering step. However, this additional information predicted by Liang's network could have been obtained from an object detector. Arnab *et al*. [3] also started with an initial semantic segmentation network [2], and combined this with the outputs of an object detector using a CRF to reason about instances. This method was not trained end-to-end though, and could not really recover from errors in bounding-box localisation or occlusion.

Our method also has an initial semantic segmentation subnetwork, and uses the outputs of an object detector. However, in contrast to [3] it is trained end-to-end to improve on both semantic- and instance-segmentation performance (to our knowledge, this is the first work to achieve this). Furthermore, it can handle detector localisation er-

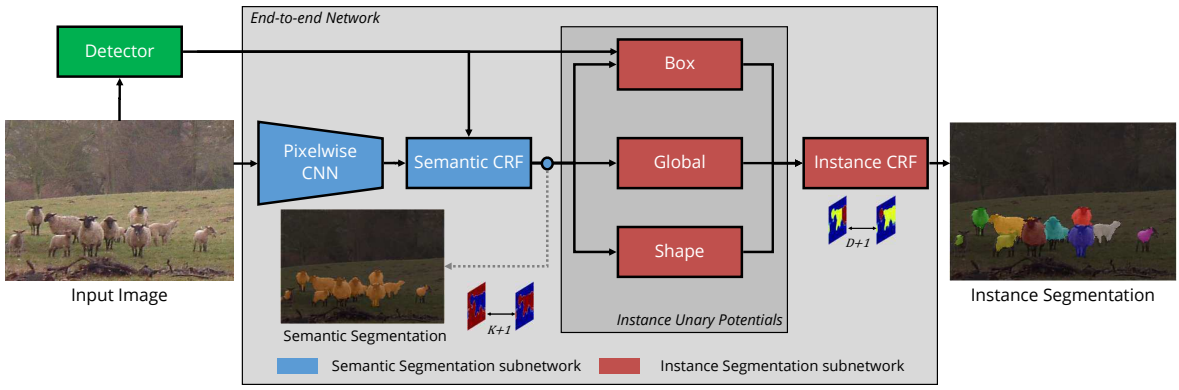


Figure 2. Network overview: Our end-to-end trained network consists of semantic- and instance-segmentation modules. The intermediate category-level segmentation, along with the outputs of an object detector, are used to reason about instances. This is done by instance unary terms which use information from the detector's bounding boxes, the initial semantic segmentation and also the object's shape. A final CRF is used to combine all this information together to obtain an instance segmentation. The output of the semantic segmentation module is a fixed size $W \times H \times (K+1)$ tensor where K is the number of object classes, excluding background, in the dataset. The final output, however, is of a variable $W \times H \times (D+1)$ dimensions where D is the number of detected objects (and one background label).

rors and occlusions better due to the energy terms in our end-to-end CRF. In contrast to detection-based approaches [18, 19, 11, 32], our network requires no additional post-processing to create an instance segmentation map as in Fig. 1(c) and reasons about the entire image, rather than independent proposals. This global reasoning allows our method to produce more accurate segmentations. Our proposed system also handles a variable number of instances per image, and thus does not assume a maximum number of instances like [50, 51].

3. Proposed Approach

Our network (Fig. 2) contains an initial semantic segmentation module. We use the semantic segmentation result, along with the outputs of an object detector, to compute the unary potentials of a Conditional Random Field (CRF) defined over object instances. We perform mean field inference in this random field to obtain the Maximum a Posteriori (MAP) estimate, which is our labelling. Although our network consists of two conceptually different parts – a semantic segmentation module, and an instance segmentation network – the entire pipeline is fully differentiable, given object detections, and trained end-to-end.

3.1. Semantic Segmentation subnetwork

Semantic Segmentation assigns each pixel in an image a semantic class label from a given set, $\mathcal{L}$. In our case, this module uses the FCN8s architecture [33] which is based on the VGG [40] ImageNet model. For better segmentation results, we include mean field inference of a CRF as the module's last layer. This CRF contains the densely-connected pairwise potentials described in [23] and is formulated as a recurrent neural network as in [53]. Additionally, we include the Higher Order detection potential described in [2]. This detection potential has two primary benefits: Firstly, it improves semantic segmentation quality by encouraging consistency between object detections and segmentations. Secondly, it also recalibrates detection scores. This detection potential is similar to the one previously proposed by [25], [41], [45] and [48], but formulated for the differentiable mean field inference algorithm. We employ this potential as we are already using object detection information for identifying object instances in the next stage. We denote the output at the semantic segmentation module of our network as the tensor $\mathbf{Q}$, where $Q_i(l)$ denotes the probability (obtained by applying the softmax function on the network's activations) of pixel i taking on the label $l \in \mathcal{L}$.

3.2. Instance Segmentation subnetwork

At the input to our instance segmentation subnetwork, we assume that we have two inputs available: The semantic segmentation predictions, $\mathbf{Q}$, for each pixel and label, and a set of object detections. For each input image, we assume that there are D object detections, and that the i^{th} detection is of the form (l_i, s_i, B_i) where $l_i \in \mathcal{L}$ is the detected class label, $s_i \in [0, 1]$ is the confidence score and B_i is the set of indices of the pixels falling within the detector's bounding box. Note that the number D varies for every input image.

The problem of instance segmentation can then be thought of as assigning every pixel to either a particular object detection, or the background label. This is based on the assumption that every object detection specifies a potential object instance. We define a multinomial random variable, V, at each of the N pixels in the image, and $\mathbf{V} = [V_1\ V_2 \ldots V_N]^T$. Each variable at pixel i, V_i, is assigned a label corresponding to its instance. This label set, $\{0, 1, 2, ..., D\}$ changes for each image since D, the number of detections, varies for every image (0 is the background label). In the case of instance segmentation of images, the

(a) Semantic Segmentation (b) Instance Segmentation

Figure 3. Instance segmentation using only the "Box" unary potential. This potential is effective when we have a good initial semantic segmentation (a). Occlusions between objects of the same class can be resolved by the pairwise term based on appearance differences. Note that we can ignore the confident, false-positive "bottle" detections (b). This is in contrast to methods such as [7, 18, 19, 26] which cannot recover from detection errors.

quality of a prediction is invariant to the permutations of the instance labelling. For example, labelling the "blue person" in Fig. 1(c) as "1" and the "purple person" as "2" is no different to labelling them as "2" and "1" respectively. This condition is handled by our loss function in Sec. 3.4.

Note that unlike works such as [50] and [51] we do not assume a maximum number of possible instances and keep a fixed label set. Furthermore, since we are considering object detection outputs jointly with semantic segmentation predictions, we have some robustness to high-scoring false positive detections unlike methods such as [7, 19, 32] which refine object detections into segmentations.

We formulate a Conditional Random Field over our instance variables, V, which consists of unary and pairwise energies. The energy of the assignment $\mathbf{v}$ to all the variables, $\mathbf{V}$, is

$$E(\mathbf{V}=\mathbf{v}) = \sum_i U(v_i) + \sum_{i<j} P(v_i, v_j). \quad (1)$$

The unary energy is a sum of three terms, which take into account the object detection bounding boxes, the initial semantic segmentation and shape information,

$$U(v_i) = -\ln[w_1\psi_{Box}(v_i) + w_2\psi_{Global}(v_i) + w_3\psi_{Shape}(v_i)], \quad (2)$$

and are described further in Sections 3.2.1 through 3.2.3. w_1, w_2 and w_3 are all weighting co-efficients learned via backpropagation.

3.2.1 Box Term

This potential encourages a pixel to be assigned to the instance corresponding to the k^{th} detection if it falls within the detection's bounding box. This potential is proportional to the probability of the pixel's semantic class being equal to the detected class $Q_i(l_k)$ and the detection score, s_k.

$$\psi_{Box}(V_i=k) = \begin{cases} Q_i(l_k)s_k & \text{if } i \in B_k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

(a) Only Box term (b) Box and Global terms

Figure 4. The "Global" unary potential (b) is particularly effective in cases where the input detection bounding box does not cover the entire extent of the object. Methods which are based on refining bounding-box detections such as [18, 19, 7, 11] cannot cope with poorly localised detections. Note, the overlaid detection boxes are an additional input to our system.

As shown in Fig. 3, this potential performs well when the initial semantic segmentation is good. It is robust to false positive detections, unlike methods which refine bounding boxes [7, 18, 19] since the detections are considered in light of our initial semantic segmentation, **Q**. Together with the pairwise term (Sec. 3.2.4), occlusions between objects of the same class can be resolved if there are appearance differences in the different instances.

3.2.2 Global Term

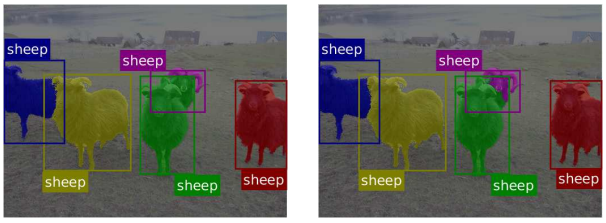
This term does not rely on bounding boxes, but only the segmentation prediction at a particular pixel, Q_i. It encodes the intuition that if we only know there are d possible instances of a particular object class, and have no further localisation information, each instance is equally probable, and this potential is proportional to the semantic segmentation confidence for the detected object class at that pixel:

$$\psi_{Global}(V_i=k) = Q_i(l_k). \quad (4)$$

As shown in Fig. 4, this potential overcomes cases where the bounding box does not cover the entire extent of the object, as it assigns probability mass to a particular instance label throughout all pixels in the image. This is also beneficial during training, as it ensures that the final output is dependent on the segmentation prediction at all pixels in the image, leading to error gradients that are more stable across batches and thus more amenable to backpropagation.

3.2.3 Shape Term

We also incorporate shape priors to help us reason about occlusions involving multiple objects of the same class, which may have minimal appearance variation between them, as shown in Fig. 5. In such cases, a prior on the expected shape of an object category can help us to identify the foreground instance within a bounding box. Previous approaches to incorporating shape priors in segmentation [22, 7, 43] have involved generating "shape exemplars" from the training dataset and, at inference time, matching these exemplars to object proposals using the Chamfer distance [39, 31].



(a) Without shape term (b) With Shape term

Figure 5. The "Shape" unary potential (b) helps us to distinguish between the green and purple sheep, which the other two unary potentials cannot. Input detections are overlaid on the images.

We propose a fully differentiable method: Given a set of shape templates, $\mathcal{T}$, we warp each shape template using bilinear interpolation into $\tilde{\mathcal{T}}$ so that it matches the dimensions of the k^{th} bounding box, B_k. We then select the shape prior which matches the segmentation prediction for the detected class within the bounding box, $\mathbf{Q}_{B_k}(l_k)$, the best according to the normalised cross correlation. Our shape prior is then the Hadamard (elementwise) product ($\odot$) between the segmentation unaries and the matched shape prior:

$$t^* = \arg\max_{t \in \tilde{\mathcal{T}}} \frac{\sum \mathbf{Q}_{B_k}(l_k) \odot t}{\|\mathbf{Q}_{B_k}(l_k)\| \, \|t\|} \tag{5}$$

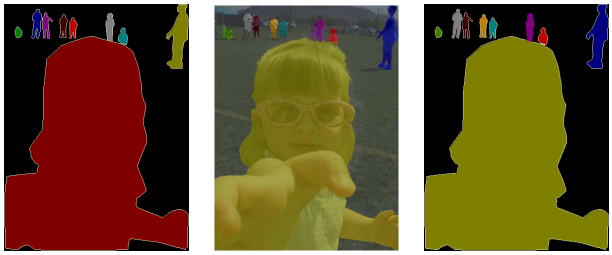
$$\psi(\mathbf{V}_{B_k} = k) = \mathbf{Q}_{B_k}(l_k) \odot t^*. \tag{6}$$

Equations 5 and 6 can be seen as a special case of max-pooling, and the numerator of Eq. 5 is simply a convolution that produces a scalar output since the two arguments are of equal dimension. Additionally, during training, we can consider the shape priors $\mathcal{T}$ as parameters of our "shape term" layer and backpropagate through to the matched exemplar t^* to update it. In practice, we initialised these parameters with the shape priors described in [43]. This consists of roughly 250 shape templates for each of five different aspect ratios. These were obtained by clustering foreground masks of object instances from the training set.

Here, we have only matched a single shape template to a proposed instance. This method could be extended in future to matching multiple templates to an instance, in which case each shape exemplar would correspond to a part of the object such as in DPM [14].

3.2.4 Pairwise term

The pairwise term consists of densely-connected Gaussian potentials [23] and encourages appearance and spatial consistency. The weights governing the importance of these terms are also learnt via backpropagation, as in [53]. We find that these priors are useful in the case of instance segmentation as well, since nearby pixels that have similar appearance often belong to the same object instance. They are often able to resolve occlusions based on appearance differences between objects of the same class (Fig. 3).



(a) Original ground truth, $\mathcal{G}$ (b) Prediction, $\mathcal{P}$ (c) "Matched" ground truth, $\mathcal{G}^*$

Figure 6. Due to the problem of label permutations, we "match" the ground truth with our prediction before computing the loss when training.

3.3. Inference of our Dynamic Instance CRF

We use mean field inference to approximately minimise the Gibbs Energy in Eq. 1 which corresponds to finding the Maximum a Posteriori (MAP) labelling of the corresponding probability distribution, $P(\mathbf{V} = \mathbf{v}) = \frac{1}{Z} \exp\left(-E(\mathbf{v})\right)$ where Z is the normalisation factor. Mean field inference is differentiable, and this iterative algorithm can be unrolled and seen as a recurrent neural network [53]. Following this approach, we can incorporate mean field inference of a CRF as a layer of our neural network. This enables us to train our entire instance segmentation network end-to-end.

Because we deal with a variable number of instances for every image, our CRF needs to be dynamically instantiated to have a different number of labels for every image, as observed in [3]. Therefore, unlike [53], none of our weights are class-specific. This weight-sharing not only allows us to deal with variable length inputs, but class-specific weights also do not make sense in the case of instance segmentation since a class label has no particular semantic meaning.

3.4. Loss Function

When training for instance segmentation, we have a single loss function which we backpropagate through our instance- and semantic-segmentation modules to update all the parameters. As discussed previously, we need to deal with different permutations of our final labelling which could have the same final result. The works of [50] and [51] order instances by depth to break this symmetry. However, this requires ground-truth depth maps during training which we do not assume that we have. Proposal-based methods [11, 18, 19, 32] do not have this issue since they consider a single proposal at a time, rather than the entire image. Our approach is similar to [38] in that we match the original ground truth to our instance segmentation prediction based on the Intersection over Union (IoU) [13] of each instance prediction and ground truth, as shown in Fig. 6.

More formally, we denote the ground-truth labelling of an image, $\mathcal{G}$, to be a set of r segments, $\{g_1, g_2, \ldots, g_r\}$, where each segment (set of pixels) is an object instance

and has an associated semantic class label. Our prediction, which is the output of our network, $\mathcal{P}$, is a set of s segments, $\{p_1, p_2, \ldots, p_s\}$, also where each segment corresponds to an instance label and also has an associated class label. Note that r and s may be different since we may predict greater or fewer instances than actually present. Let $\mathcal{M}$ denote the set of all permutations of the ground-truth, $\mathcal{G}$. As can be seen in Fig. 6, different permutations of the ground-truth correspond to the same qualitative result. We define the "matched" ground-truth, $\mathcal{G}^*$, as the permutation of the original ground-truth labelling which maximises the IoU between the prediction, $\mathcal{P}$, and ground truth:

$$\mathcal{G}^* = \arg\max_{m \in \mathcal{M}} \text{IoU}(m, \mathcal{P}). \tag{7}$$

Once we have the "matched" ground truth, $\mathcal{G}^*$, (Fig. 6) for an image, we can apply any loss function to train our network for segmentation. In our case, we use the common cross-entropy loss function. We found that this performed better than the approximate IoU loss proposed in [24, 38].

Crucially, we do not need to evaluate all permutations of the ground truth to compute Eq. 7, since it can be formulated as a maximum-weight bipartite matching problem. The edges in our bipartite graph connect ground-truth and predicted segments. The edge weights are given by the IoU between the ground truth and predicted segments if they share the same semantic class label, and zero otherwise. Leftover segments are matched to "dummy" nodes with zero overlap.

Additionally, the ordering of the instances in our network are actually determined by the object detector, which remains static during training. As a result, the ordering of our predictions does not fluctuate much during training – it only changes in cases where there are multiple detections overlapping an object.

3.5. Network Training

We first train a network for semantic segmentation with the standard cross-entropy loss. In our case, this network is FCN8s [33] with a CRF whose inference is unrolled as an RNN and trained end-to-end, as described in [53] and [2]. To this pretrained network, we append our instance segmentation subnetwork, and finetune with instance segmentation annotations and only the loss detailed in Sec. 3.4. For the semantic segmentation subnetwork, we train with an initial learning rate of 10^{-8}, momentum of 0.9 and batch size of 20. The learning rate is low since we do not normalise the loss by the number of pixels. This is so that images with more pixels contribute a higher loss. The normalised learning rate is approximately 2×10^{-3}. When training our instance segmentation network as well, we lower the learning rate to 10^{-12} and use a batch size of 1 instead. Decreasing the batch size gave empirically better results. We also clipped gradients (a technique common in training RNNs [34]) with ℓ_2 norms above 10^9. This threshold was set by observing "normal" gradient magnitudes during training. The relatively high magnitude is due to the fact that our loss is not normalised. In our complete network, we have two CRF inference modules which are RNNs (one each in the semantic- and instance-segmentation subnetworks), and gradient clipping facilitated successful training.

3.6. Discussion

Our network is able to compute a semantic and instance segmentation of the input image in a single forward pass. We do not require any post-processing, such as the patch aggregation of [32], "mask-voting" of [11], "superpixel projection" of [18, 19, 26] or spectral clustering of [28]. The fact that we compute an initial semantic segmentation means that we have some robustness to errors in the object detector (Fig. 3). Furthermore, we are not necessarily limited by poorly localised object detections either (Fig. 4). Our CRF model allows us to reason about the entire image at a time, rather than consider independent object proposals, as done in [18, 19, 11, 32, 26]. Although we do not train our object detector jointly with the network, it also means that our segmentation network and object detector do not succumb to the same failure cases. Moreover, it ensures that our instance labelling does not "switch" often during training, which makes learning more stable. Finally, note that although we perform mean field inference of a CRF within our network, we do not optimise the CRF's likelihood, but rather a cross-entropy loss (Sec 3.4).

4. Experimental Evaluation

Sections 4.1 to 4.6 describe our evaluation on the Pascal VOC Validation Set [13] and the Semantic Boundaries Dataset (SBD) [17] (which provides per-pixel annotations to 11355 previously unlaballed images from Pascal VOC). Section 4.7 details results on Cityscapes [8].

4.1. Experimental Details

We first train a network for semantic segmentation, thereafter we finetune it to the task of instance segmentation, as described in Sec. 3.5. Our training data for the semantic segmentation pretraining consists of images from Pascal VOC [13], SBD [17] and Microsoft COCO [30]. Finally, when finetuning for instance segmentation, we use only training data from either the VOC dataset, or from the SBD dataset. We train separate models for evaluating on the VOC Validation Set, and the SBD Validation Set. In each case, we remove validation set images from the initial semantic segmentation pretraining set. We use the publicly available R-FCN object detection framework [12], and ensure that the images used to train the detector do not fall into our test sets for instance segmentation.

4.2. Evaluation Metrics

We report the mean Average Precision over regions (AP^r) as defined by [18]. The difference between AP^r and the AP metric used in object detection [13] is that the Intersection over Union (IoU) is computed over predicted and ground-truth regions instead of bounding boxes. Furthermore, the standard AP metric uses an IoU threshold of 0.5 to determine whether a prediction is correct or not. Here, we use a variety of IoU thresholds since larger thresholds require more precise segmentations. Additionally, we report the AP^r_{vol} which is the average of the AP^r for 9 IoU thresholds ranging from 0.1 to 0.9 in increments of 0.1.

However, we also observe that the AP^r metric requires an algorithm to produce a ranked list of segments and their object class. It does not require, nor evaluate, the ability of an algorithm to produce a globally coherent segmentation map of the image, for example Fig. 1c. To measure this, we propose the "Matching IoU" which matches the predicted image and ground truth, and then calculates the corresponding IoU as defined in [13]. This matching procedure is the same as described in Sec. 3.4. This measure was originally proposed in [47], but has not been used since in evaluating instance segmentation systems.

4.3. Effect of Instance Potentials and End-to-End training

We first perform ablation studies on the VOC 2012 Validation set. This dataset, consisting of 1464 training and 1449 validation images has very high-quality annotations with detailed object delineations which makes it the most suited for evaluating pixel-level segmentations.

In Tab. 1, we examine the effect of each of our unary potentials in our Instance subnetwork on overall performance. Furthermore, we examine the effect of end-to-end training the entire network as opposed to piecewise training. Piecewise training refers to freezing the pretrained semantic segmentation subnetwork's weights and only optimising the instance segmentation subnetwork's parameters. Note that when training with only the "Box" (Eq. 3) unary potential and pairwise term, we also have to add in an additional "Background" detection which encompasses the entire image. Otherwise, we cannot classify the background label.

We can see that each unary potential improves overall instance segmentation results, both in terms of AP^r_{vol} and the Matching IoU. The "Global" term (Eq. 4) shows particular improvement over the "Box" term at the high AP^r threshold of 0.9. This is because it can overcome errors in bounding box localisation (Fig. 4) and leverage our semantic segmentation network's accurate predictions to produce precise labellings. The "Shape" term's improvement in the AP^r_{vol} is primarily due to an improvement in the AP^r at low thresholds. By using shape priors, we are able to recover instances which were occluded and missed out. End-to-end training also improves results at all AP^r thresholds. Training with just the "Box" term shows a modest improvement in the AP^r_{vol} of 1.3%. Training with the "Global" and "Shape" terms shows larger improvements of 2.1% and 2.3% respectively. This may be because the "Box" term only considers the semantic segmentation at parts of the image covered by object detections. Once we include the "Global" term, we consider the semantic segmentation over the entire image for the detected class. Training makes more efficient use of images, and error gradients are more stable in this case.

Table 1. The effect of the different CRF unary potentials, and end-to-end training with them, on the VOC 2012 Validation Set.

	AP^r			AP^r_{vol}	match
	0.5	0.7	0.9		IoU
Box Term (piecewise)	60.0	47.3	21.2	54.9	42.6
Box+Global (piecewise)	59.1	46.1	23.4	54.6	43.0
Box+Global+Shape (piecewise)	59.5	46.4	23.3	55.2	44.8
Box Term (end-to-end)	60.7	47.4	24.6	56.2	46.9
Box+Global (end-to-end)	60.9	48.1	**25.5**	56.7	47.1
Box+Global+Shape (end-to-end)	**61.7**	**48.6**	25.1	**57.5**	**48.3**

Table 2. Comparison of Instance Segmentation performance to recent methods on the VOC 2012 Validation Set

Method	AP^r					AP^r_{vol}
	0.5	0.6	0.7	0.8	0.9	
SDS [18]	43.8	34.5	21.3	8.7	0.9	–
Chen *et al.* [7]	46.3	38.2	27.0	13.5	2.6	–
PFN [28]	58.7	51.3	42.5	31.2	15.7	52.3
Arnab *et al.* [3]	58.3	52.4	45.4	34.9	20.1	53.1
MPA 1-scale [32]	60.3	54.6	45.9	34.3	17.3	54.5
MPA 3-scale [32]	**62.1**	**56.6**	47.4	36.1	18.5	56.5
Ours	61.7	55.5	**48.6**	**39.5**	**25.1**	**57.5**

4.4. Results on VOC Validation Set

We then compare our best instance segmentation model to recent methods on the VOC Validation Set in Tab. 2. The fact that our algorithm achieves the highest AP^r at thresholds above 0.7 indicates that our method produces more detailed and accurate segmentations.

At an IoU threshold of 0.9, our improvement over the previous state-of-the-art (MPA [32]) is 6.6%, which is a relative improvement of 36%. Unlike [32, 18, 7], our network performs an initial semantic segmentation which may explain our more accurate segmentations. Other segmentation-based approaches, [3, 28] are not fully end-to-end trained. We also achieve the best AP^r_{vol} of 57.5%. The relatively small difference in AP^r_{vol} to MPA [32] despite large improvements at high IoU thresholds indicates

Table 3. Comparison of Instance Segmentation performance on the SBD Dataset

Method	AP^r 0.5	AP^r 0.7	AP^r_{vol}	match IoU
SDS [18]	49.7	25.3	41.4	–
MPA 1-scale [32]	55.5	–	48.3	–
Hypercolumn [19]	56.5	37.0	–	–
IIS [26]	60.1	38.7	–	–
CFM [10]	60.7	39.6	–	–
Hypercolumn rescore [19]	60.0	40.4	–	–
MPA 3-scale rescore [32]	61.8	–	52.0	–
MNC [11]	63.5	41.5	–	39.0
MNC, Instance FCN [9]	61.5	43.0	–	–
IIS sp. projection, rescore [26]	**63.6**	43.3	–	–
Ours (piecewise)	59.1	42.1	52.3	41.8
Ours (end-to-end)	62.0	**44.8**	**55.4**	**47.3**

Table 4. Semantic Segmentation performance before and after finetuning for Instance Segmentation

Dataset	Mean IoU [%] before Instance finetuning	Mean IoU [%] after Instance finetuning
VOC	74.2	75.1
SBD	71.5	72.5

that MPA performs better at low IoU thresholds. Proposal-based methods, such as [32, 18] are more likely to perform better at low IoU thresholds since they output more proposals than actual instances in an image (SDS evaluates 2000 proposals per image). Furthermore, note that whilst MPA takes 8.7s to process an image [32], our method requires approximately 1.1s on the same Titan X (Maxwell) GPU. More detailed qualitative and quantitative results are included in the supplementary material.

4.5. Results on SBD Dataset

We also evaluate our model on the SBD dataset, which consists of 5623 training and 5732 validation images, as shown in Tab. 3. Following other works, we only report AP^r results at IoU thresholds of 0.5 and 0.7. However, we provide more detailed results in our supplementary material. Once again, we show significant improvements over other work at high AP^r thresholds. Here, our AP^r at 0.7 improves by 1.5% over the previous state-of-the-art [26]. Note that [26, 32, 19] perform additional post-processing where their results are rescored using an additional object detector. In contrast, our results are obtained by a single forward pass through our network. We have also improved substantially on the AP^r_{vol} measure (3.4%) compared to other works which have reported it. We also used the publicly available source code, model and default parameters of MNC [11] to evaluate the "Matching IoU". Our method improves this by 8.3%. This metric is a stricter measure of segmentation performance, and our method, which is based on an initial semantic segmentation and includes a CRF as part of training therefore performs better.

Table 5. Results on Cityscapes Test Set. Evaluation metrics and results of competing methods obtained from the online server. The "AP" metric of Cityscapes is similar to our AP^r_{vol} metric.

Method	AP	AP at 0.5	AP 100m	AP 50m
Ours	**20.0**	**38.8**	**32.6**	**37.6**
SAIS [20]	17.4	36.7	29.3	34.0
Pixel Encoding [42]	8.9	21.1	15.3	16.7

4.6. Improvement in Semantic Segmentation

Finetuning our network for instance segmentation, with the loss described in Sec. 3.4 improves semantic segmentation performance on both the VOC and SBD dataset, as shown in Tab. 4. The improvement is 0.9% on VOC, and 1% on SBD. The tasks of instance segmentation and semantic segmentation are highly related – in fact, instance segmentation can be thought of as a more specific case of semantic segmentation. As a result, finetuning for one task improves the other.

4.7. Results on Cityscapes

Finally, we evaluate our algorithm on the Cityscapes road-scene understanding dataset [8]. We evaluate on the test set, consisting of 1525 images on the online server, and use none of the 500 validation images for training. We use an initial semantic segmentation subnetwork that is based on the ResNet-101 architecture [52], and all of the instance unary potentials described in Sec. 3.2.

As shown in Tab. 5, our method sets a new state-of-the-art on Cityscapes, surpassing concurrent work [20] and the best previous published work [42] by significant margins.

5. Conclusion

We have presented an end-to-end instance segmentation approach that produces intermediate semantic segmentations, and shown that finetuning for instance segmentation improves our network's semantic segmentations. Our approach differs from other methods which derive their architectures from object detection networks [11, 32, 19] in that our approach is more similar to a semantic segmentation network. As a result, our system produces more accurate and detailed segmentations as shown by our substantial improvements at high AP^r thresholds. Moreover, our system produces segmentation maps naturally, and in contrast to other published work, does not require any post-processing. Finally, our network produces a variable number of outputs, depending on the number of instances in the image.

Acknowledgements We thank Bernardino Romera-Paredes and Stuart Golodetz for insightful discussions and feedback. This work was supported by the EPSRC, Clarendon Fund, ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1 and EPSRC/MURI grant EP/N019474/1.

References

[1] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, pages 328–335. IEEE, 2014. 2

[2] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016. 2, 3, 6

[3] A. Arnab and P. H. S. Torr. Bottom-up instance segmentation with deep higher order crfs. In *BMVC*, 2016. 2, 5, 7

[4] L. Chen, A. Schwing, A. Yuille, and R. Urtasun. Learning deep structured models. In *ICML*, Lille, France, 2015. 2

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

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 1

[7] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In *CVPR*, pages 3470–3478, 2015. 2, 4, 7

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 6, 8

[9] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *ECCV*, 2016. 1, 8

[10] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 2, 8

[11] 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, 8

[12] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 1, 6

[13] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 5, 6, 7

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010. 5

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[16] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*. 2014. 1

[17] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998. IEEE, 2011. 6

[18] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312. Springer, 2014. 1, 2, 3, 4, 5, 6, 7, 8

[19] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. *CVPR*, 2015. 1, 2, 3, 4, 5, 6, 8

[20] Z. Hayder, X. He, and M. Salzmann. Shape-aware instance segmentation. In *arXiv preprint arXiv:1612.03129*, 2016. 8

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[22] X. He and S. Gould. An Exemplar-based CRF for Multi-instance Object Segmentation. In *CVPR*, 2014. 4

[23] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *NIPS*, 2011. 3, 5

[24] P. Krähenbühl and V. Koltun. Parameter learning and convergent inference for dense random fields. In *ICML*, 2013. 6

[25] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr. What, where and how many? combining object detectors and crfs. In *ECCV*, pages 424–437, 2010. 3

[26] K. Li, B. Hariharan, and J. Malik. Iterative Instance Segmentation. In *CVPR*, 2016. 1, 2, 4, 6, 8

[27] X. Liang, Y. Wei, X. Shen, Z. Jie, J. Feng, L. Lin, and S. Yan. Reversible recursive instance-level object segmentation. In *CVPR*, 2016. 1, 2

[28] 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. 2, 6, 7

[29] G. Lin, C. Shen, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 2

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 6

[31] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *CVPR*, pages 1696–1703. IEEE, 2010. 4

[32] S. Liu, X. Qi, J. Shi, H. Zhang, and J. Jia. Multi-scale patch aggregation (mpa) for simultaneous detection and segmentation. In *CVPR*, 2016. 1, 2, 3, 4, 5, 6, 7, 8

[33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3, 6

[34] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *ICML*, pages 1310–1318, 2013. 6

[35] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *NIPS*, 2015. 1

[36] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollar. Learning to refine object segments. In *ECCV*, 2016. 1

[37] 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

[38] B. Romera-Paredes and P. H. Torr. Recurrent instance segmentation. In *ECCV*, 2016. 2, 5, 6

[39] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, volume 1, pages 503–510. IEEE, 2005. 4

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3

[41] M. Sun, B.-s. Kim, P. Kohli, and S. Savarese. Relating things and stuff via object property interactions. *PAMI*, 36(7):1370–1383, 2014. 3

[42] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *German Conference on Pattern Recognition (GCPR)*, 2016. 8

[43] D. Weiss and B. Taskar. Scalpel: Segmentation cascades with localized priors and efficient learning. In *CVPR*, pages 2035–2042, 2013. 4, 5

[44] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006. 2

[45] C. Wojek and B. Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In *ECCV*, pages 733–747. Springer, 2008. 3

[46] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang. Deep interactive object selection. In *CVPR*, 2016. 1

[47] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes. Layered object models for image segmentation. *PAMI*, 2012. 7

[48] J. Yao, S. Fidler, and R. Urtasun. Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation. In *CVPR*, pages 702–709, 2012. 3

[49] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár. A multipath network for object detection. In *BMVC*, 2016. 1

[50] Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *CVPR*, 2016. 2, 3, 4, 5

[51] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *ICCV*, pages 2614–2622, 2015. 2, 3, 4, 5

[52] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *arXiv preprint arXiv:1612.01105*, 2017. 8

[53] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2, 3, 5, 6