Learning Spatial Transforms for Refining Object Segment Proposals

Haoyang Zhang
ANU, Data61 CSIRO
Haoyang.Zhang@anu.edu.au

Xuming He
Data61 CSIRO, ANU
Xuming.He@anu.edu.au

Fatih Porikli
ANU, Data61 CSIRO
Fatih.Porikli@anu.edu.au

Abstract

We address the problem of object segment proposal generation, which is a critical step in many instance-level semantic segmentation and scene understanding pipelines. In contrast to prior works that predict binary segment masks from images, we take an alternative refinement approach to improve the quality of a given segment candidate pool. In particular, we propose an efficient deep network that learns 2D spatial transforms to warp an initial object mask towards nearby object region. We formulate this segment refinement task as a regression problem and design a novel feature pooling strategy in our deep network to predict an affine transformation for each object mask. We evaluate our method extensively on two challenging public benchmarks and apply our refinement network to three different initial segment proposal settings. Our results show sizable improvements in average recall across all the settings, achieving the state-of-the-art performances.

1. Introduction

Scene parsing at object-instance level provides a rich description of images in terms of individual objects and their spatial relations, and has many real-world applications including automatic navigation [3, 8], personal robotics [18, 24] and visual analytics [30]. In particular, instance-level semantic segmentation, which jointly detects and segments all the objects in an image, has attracted much attention recently [13, 5, 20]. As in most modern object detection systems [10, 29], a critical step in object segmentation is to generate generic object segment proposals for its downstream classification and/or global reasoning [5, 25, 26].

Generating object segment proposals, unlike their bounding box counterparts, entails both object-level localization and pixelwise perceptual grouping. Early works build on grouping pixels using mid-level cues, such as graph-cut based CPMC [1], Multiscale combinatorial grouping (MCG) [27], and Selective search [32], which are largely limited by the inaccurate over-segmentation processes. More recent approaches learn deep networks to produce binary masks from the image directly, including DeepMask [25], SharpMask [26] and Multistage networks [5]. Nevertheless, learning such a direct mapping from images to segments has shown to be challenging, which usually produces object masks lacking good boundary alignment and requires post-processing to improve their quality.

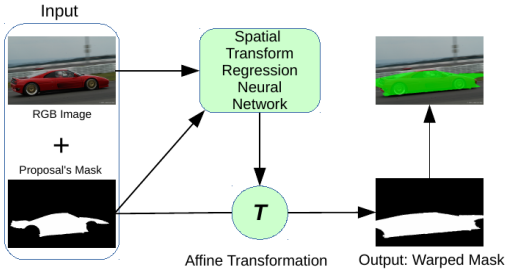

Figure 1. Overview of our segment proposal refinement pipeline. We propose to learn a regression network to warp initial segment candidates towards the groundtruth objects.

An alternative approach to generating better object proposals is to refine an initial set of object segments produced by existing methods [26, 20]. Such a strategy enables us to use the initial segment as a starting point and learn additional feature representations for improving the mask accuracy. Hence it is more flexible than the group-and-rank methods [1, 27]. In addition, as it aims to minimize the residual error between the initial segments and the ground truth, the problem of refinement is conceptually simpler than solving the original image-to-mask mapping task. In essence, it learns a transformation that moves the initial mask predictions 'closer' to the target object segments.

In this work, we propose an efficient object segment refinement method that learns spatial transforms to improve the pixel-level accuracy of the object proposals. In contrast to the prior approaches that build a refinement network to predict pixelwise masks [26], our method takes both image and initial object masks as input, and predicts a spatial affine transformation in 2D image plane for each mask, which is then used to warp the corresponding mask into a more accurate object segment candidate. Figure 1 illustrates an overview of our approach.

Specifically, we formulate the segment refinement as a regression problem, and build a deep network to predict the 2D affine transformation required for improving the mask accuracy. Given the input image, we first extract a hypercolumn feature representation [12] to represent the multi-scale image cues. On these feature maps, we design a novel mask pooling scheme that incorporates cues from both an initial object segment and its spatial context. The pooled features are fed into a four-layer neural network, which outputs affine transformation parameters for warping the object mask. To train the regression network, we precompute the affine transformations from the initial object masks to their corresponding groundtruth masks based on nonrigid registration [31], which are used as our regression targets.

We evaluate our approach extensively on two publicly available datasets with object instance segmentation ground truth, the Cityscapes [3] dataset and the PASCAL VOC dataset [7, 11]. Our refinement network is applied to three different sets of initial object segments generated from MCG, DeepMask and SharpMask respectively, and achieves sizable improvements in the average recall rate across all the experimental settings.

The contributions of our work are three folds: First, we propose a novel refinement method that learns spatial transforms for improving the quality of object segment proposals. Second, we design and train an efficient deep network to predict the instance-level affine transformations based on hypercolumn feature and mask pooling. Finally, our experimental evaluation shows consistent improvements over several state-of-the-art methods on challenging benchmarks. The main strengths of our approach lie in its *generality*, as it can be applied to any initial object segment proposals; and its *simplicity*, as we only need to predict a spatial transform in a low-dimensional space.

2. Related Work

While much progress has been made in semantic segmentation [7, 8], most prior approaches focus on pixelwise labeling of images using semantic classes. Region proposals are first used in semantic segmentation to capture mid-level and object features [1]. However, they do not produce segmentation w.r.t individual object instances. Instance-level semantic segmentation, by contrast, assigns both category and instance labels (mostly for foreground classes) to every pixel, which parses images at a more detailed level and has attracted much attention in vision community recently [13, 21, 12, 20, 5]. As instance segmentation requires simultaneously detecting objects and assigning class labels, object segment proposals have been widely adopted to reduce the overall search space [5, 25].

Generic object segment proposals extend the concept of bounding box candidates used in object detection [32, 36, 6], consisting of a set of regions with arbitrary shapes [1, 27, 16]. Early works in segment proposal generation formulate the problem as a series of foreground region segmentation tasks [1], and solve multiple segmentations with diverse seeds. Recently, Lee *et al*. [19] propose a parametric energy function to combine multiple mid-level cues and generate a diverse set of region proposals. Other approaches group superpixels into a segmentation hierarchy and choose semantic object proposals from all the levels [27]. In [27], the authors generate multiple segmentation trees based on UCM construction and rank singletons, pairs, triplets and 4-tuples of tree nodes to select their object proposals. The method in [28] integrates global and local grouping strategies to generate foreground masks. Yanulevskaya *et al*. also learn a grouping method using manually designed appearance features and similarity metrics [34]. These segmentation-based methods, however, are prone to the inaccuracy in their bottom-up grouping process.

Recent approaches to proposing object segments learn an end-to-end deep network that directly predicts multiple binary object masks from the input image. In particular, DeepMask method builds a multi-branch deep network, jointly producing a binary mask and an objectness score for every patch in an image [25]. Dai *et al*. design a three-stage deep network for instance segmentation, in which the first two stages generate generic bounding box proposals as well as an object mask for each bounding box [5]. Our work, by contrast, takes an alternative path that aims to refine a set of generated object segment proposals.

Only a few attempts have been made to improve the quality of initial candidates in object proposal generation [2, 26]. For object segmentation, recent work of SharpMask [26] builds a refinement network on top of the DeepMask net to obtain better boundary alignment. Our method, in contrast, explicitly learns a spatial transform network to warp any initial object candidate towards its nearest object.

We note that regression has been widely used in object detection pipelines to refine the location of bounding boxes [9, 10, 29], and landmark localization problems to adjust the location of keypoints [35]. However, they are class-specific and limited to simple spatial transforms. Perhaps the most similar work is the Spatial Transformer Network [15], which learns a spatial transform to warp the image region corresponding to a target object class. By contrast, our learned transforms warp class-agnostic binary masks which are initially misaligned with object regions.

Our method is built on several existing feature representation learning techniques. In particular, we adopt the idea of the hypercolumn feature representation [12] to extract low- and mid-level image cues from the Fully Convolutional Network (FCN) [22]. Our mask pooling step extends the pooling strategy in [5, 4] to include both the information within the masked regions and spatial context around them, which is critical for predicting the warping directions.

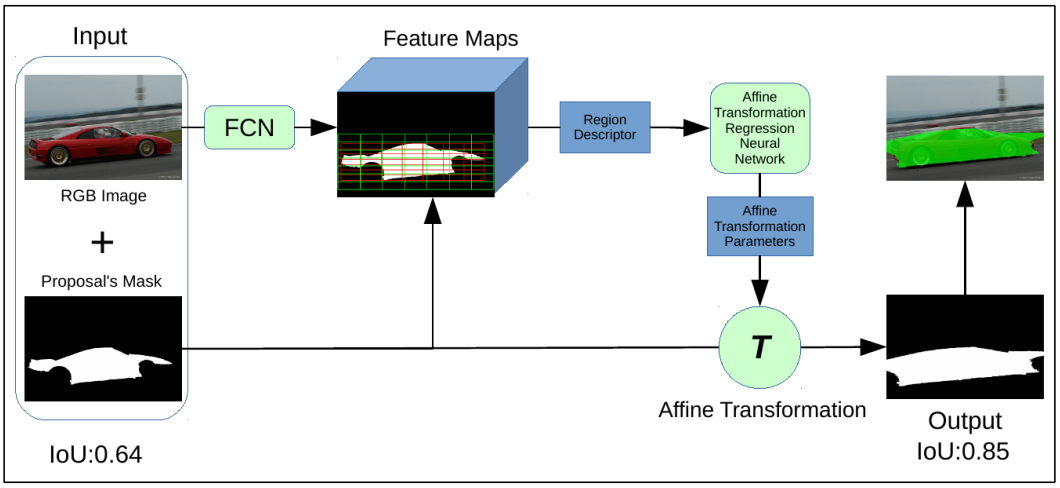

Figure 2. Model structure of our approach. Our system takes as input an image and initial segment proposals. It first extracts deep features to describe a segment and feeds the descriptor into a learned regression network to estimate an affine transformation. We then apply the affine transformation to the segment's mask to obtain the warped mask.

3. Our Approach

We aim to generate a set of high-quality object segment proposals for instance-level semantic scene understanding. To this end, we adopt a refinement strategy to improve the object mask accuracy of any initial segment candidate pool generated from existing methods. Our system takes as input an image and the binary masks of its segment proposals, and produces a transformed object mask for each initial segment proposal.

To achieve this, we design a deep neural network that predicts an affine transformation for each input segment candidate. In particular, we propose a novel mask feature pooling scheme, which allows us to extract multi-level features from a FCN. The features are fed into an efficient multi-layer network, which predicts a low-dimensional affine transformation parameter vector. We then apply the affine transformation to the initial object mask to produce a refined segment candidate. Figure 2 illustrates the overall model structure of our approach. We now describe each module of our system in detail.

3.1. Refinement by Affine Transformation

Our refinement method starts from an initial set of object segments generated by any existing proposal method. In order to evaluate the generality of our refinement procedure, we consider three segment proposal methods to cover different types of proposal mechanism in this work: 1) MCG [27], which is a state-of-the-art method based on hierarchical over-segmentation and ranking; 2) DeepMask [25], which is an end-to-end deep network method for segment generation; 3) SharpMask [26], one of the state-of-the-art method with its own refinement step.

Method(Dataset)	mean PGIoU	mean RGIoU	Gain
SharpMask(Cityscapes)	0.685	0.816	19.12%
DeepMask(Cityscapes)	0.677	0.819	20.97%
MCG(Cityscapes)	0.603	0.694	15.08%
SharpMask(PASCAL VOC)	0.688	0.803	16.72%
DeepMask(PASCAL VOC)	0.671	0.803	19.67%
MCG(PASCAL VOC)	0.628	0.721	14.83%

Table 1. The IoU scores before and after applying the oracle affine transformation to the initial segment proposals and their relative gains. The 'mean PGIoU' denotes the average IoU score of the original proposals, while the 'mean RGIoU' is the average IoU score of the warped proposals.

We note that the initial segment candidates have a large variation in their deviations from the groundtruth object segments due to inaccurate pixel groupings. In general, it requires a rich family of nonrigid transformations to warp these initial segment masks onto the groundtruth masks. However, it is challenging to predicting such nonrigid transforms due to its complexity in model design and training procedure. In this work, we instead consider a simpler family of spatial transformations for warping the input segment masks. Specifically, we adopt the 2D affine transformation for refining the segments, which has only six degrees of freedom. Such a constrained transformation space enables us to design an efficient network to predict the required transformation parameters.

To validate the sufficiency of the affine transformations, we first compute an oracle affine transformation for each input segment mask whose Intersection-over-Union (IoU) with the ground truth is larger than 0.5, and measure the improvements on the quality of segment proposals. We use

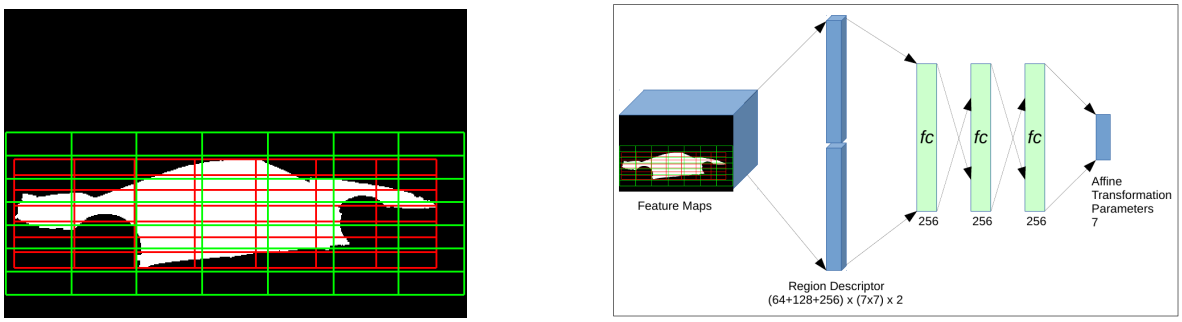

Figure 3. **Left**:The design of our mask feature pooling scheme for the region around an segment mask. We extract two types of features for a segment, denoted by the red and green grids respectively. See text for details. **Right**: The architecture of our regression network, which has four fully-connected layers and outputs 7 affine transformation parameters.

the off-the-shelf nonrigid registration toolbox [17] to compute the oracle affine transformation between an input and its nearest groundtruth mask. Table 1 shows the average IoU values before and after applying the oracle affine transformations, as well as its overall gains in percentage, on two public datasets. We can see that, while not perfect, the affine transformations are capable of achieving significant improvements over SharpMask, DeepMask and MCG, which shows their effectiveness for the refinement.

3.2. Affine Transformation Regression Network

Given an input image and an initial segment mask, we formulate the refinement as a regression problem, in which we use the image and input mask cues to predict the required affine transformation. To this end, we design a deep regression network that consists of two main components: a mask feature pooling module and a transformation regression module. We now introduce the details of these two modules as follows.

3.2.1 Mask Feature Pooling

Our mask feature pooling module is built on top of the FCN. For an input image, we first feed it into an FCN to generate multiple convolutional feature maps for the entire image. Specifically, we adopt the FCN-8s model [22], which produces feature maps from $pool_1$ to $pool_5$ with different spatial resolutions. We take the convolutional feature maps from $pool_1$, $pool_2$ and $pool_3$ for extracting our mask features, as they encode the low- and mid-level image cues and capture the geometric information required for estimating spatial transformation[1].

We design a mask feature pooling module for each input segment candidate as in most detection networks [10]. However, as our initial segments are mostly misaligned with the groundtruth object regions, we propose a dual pooling strategy to capture both the mask information and the spatial context cue of the initial segment. Specifically, we conduct the mask feature pooling with two different receptive fields and form the segment descriptors by concatenating the two types of pooled feature representations.

The first mask feature pooling aims to capture the shape of the segment mask and the convolutional features in the segment. To achieve this, we form a tight bounding box enclosing the mask and divide it evenly into $nH * nW = 7 \times 7$ cells (as illustrated by the red grid in Figure 3 (Left)). In each cell, we adopt the convolutional feature masking [4] to compute its pooled features. Specifically, we map each cell in the image domain (where the binary mask is defined) onto each layer of feature maps, *e.g.* the $pool_1$ feature maps, according to the receptive field geometry [33]. For each mapped cell, we conduct the max-pooling in the partial mask inside the cell. If no mask overlaps with the cell, the pooling output will be $\mathbf{0}$. For $pool_k$ $(k = 1, 2, 3)$ feature maps with n_k layers, we then obtain a feature vector with $n_k * nH * nW$ elements after pooling, and the first pooled feature representation is formed by concatenating such feature vectors from all three types of convolutional maps.

The second mask feature pooling captures more contextual information around the initial segment. As many masks only partially cover a groundtruth object region, we consider using a larger receptive field to pool the features so that it can provide more global information for the regression network to predict the affine transformations. Concretely, for each segment, we expand the previous tight bounding box by increasing its height and width by 1.5 times. We then pool the feature representation of the larger bounding box in a similar manner to the first mask feature pooling (as illustrated by the green grid in Figure 3 (Left)). However, we do not use mask information here and only conduct standard max-pooling within each cell.

3.2.2 Regression Network Architecture

The transformation regression module takes the segment descriptor as its input and predict the affine transformation to warp the input segment mask. Instead of generating the affine transformation matrix directly, we represent the trans-

[1]We also investigated other settings that add $pool_4$ and $pool_5$ feature maps, but did not obtain noticeable improvements.

formation by seven parameters corresponding to translation in x,y directions, rotation, scaling and shearing in x,y directions, denoted as $(t_x, t_y, r, s_x, s_y, h_x$ and $h_y)$, respectively. Formally, the 2D affine transformation T (in homogeneous coordinates) is defined as follows,

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} cos(r) & sin(r) & 0 \\ -sin(r) & cos(r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

We found this parametrization leads to a better performance in practice. Our regression network consists of three fully-connected (FC) layers followed by a linear layer to output seven parameters for the predicted affine transformation. Each fully-connected layer has 256 neurons and uses RELU as their activation functions. We also add batch normalization [14] to each layer and a dropout layer to each of the first two layers. Figure 3 (Right) illustrates the architecture of our network. We use the MatConvNet [33] toolbox to implement our network in this work.

3.3. Network Training

While our full network can be trained in an end-to-end manner, we take a two-stage training strategy due to high memory requirement in the mask feature pooling module. In particular, we first pre-train the FCN-8s model using semantic segmentation datasets (see Section 4 for details), which is used to compute the convolutional feature maps. In the second stage, we train the transformation regression network that maps the segment descriptors computed from the mask feature pooling module to the affine transformation parameters.

Training Data for Regression Network. The dataset for training the regression network is built as follows. From the initial object candidate set, we first select the object segments whose IoU with its corresponding groundtruth mask is greater than 0.5. The oracle affine transformations are then estimated using the nonrigid registration toolbox [17] and used as our ground truth for training the regression network. More concretely, we use a larger bounding box of the initial segment to crop a region of interest, and estimate the required warping from the initial mask to the corresponding groundtruth mask in that region. Interestingly, we also find that adding initial candidates with lower IoU scores does not improve the network performance.

Details of Training Procedure. Given the pairs of segment descriptor and affine transformation parameters, we train the transformation regression network to minimize the L_1 $loss$ of the training set, which is more robust than the L_2 $loss$. We use stochastic gradient descent with a batch size of 1,024 examples, momentum of 0.9, weight decay of 0.0005 and train the network for 10 epochs. The learning rate we use for each epoch gradually decreases from 0.1 to 0.0001 evenly in the log space.

4. Experiments

In this section, we evaluate our object segment proposal refinement method on two publicly available datasets: the Cityscapes dataset [3] and the PASCAL VOC dataset [7, 11]. Both datasets provide instance-level annotations for semantic segmentation.

4.1. Dataset

Cityscapes [3] is a newly released large-scale dataset for semantic urban scene understanding. It is comprised of a large diverse set of stereo video sequences recorded on streets from 50 different cities. 5,000 of these images have high quality instance-level annotations for humans and vehicles and they are split into separate training (2,975 images), validation (500 images) and test (1,525 images) sets. In our experiments, we further split the training set into two subsets: one for training (2,614 images) and the other for validation (361 images taken at Tubingen, Ulm and Zurich). We use their validation set (500 images) to evaluate the approaches, as the ground truth of the test set is withheld and their evaluation server does not provide results on proposal generation. The dataset provides instance-level annotations for humans (person and rider) and vehicles (car, truck, bus,bicycle, motorbike, caravan and trailer), which are considered as object proposal ground truth in our experiments. To compute the convolutional feature maps, we first pre-train an FCN-8s on the PASCAL-Context dataset [23], and then apply the FCN to the images with a reduced resolution of 512×1024 due to memory limitation on GPU.

The PASCAL VOC dataset [7, 11] currently contains annotations from 11,355 images taken from the PASCAL VOC 2011 dataset. For each image, it provides both category-level and instance-level segmentations for the 20 object categories in the VOC 2011 challenge. In total, it consists of 8,498 training images and 2,857 validation images. We randomly select 1,000 images from the training set as our validation set and use the 2,857 original validation images as our test set. We compute the convolutional feature maps using an FCN-8s pre-trained on this dataset.

4.2. Evaluation Metrics and Protocols

We employ three sets of metrics to evaluate the performance of our proposal refinement method: 1) The recall vs. number of proposals at three different IoU thresholds, including $IoU = 0.5, 0.6$ and 0.7; 2) The average recall (AR) vs. number of proposals; 3) The recall vs. IoU from 0.5 to 1 with 1,000 segment proposals.

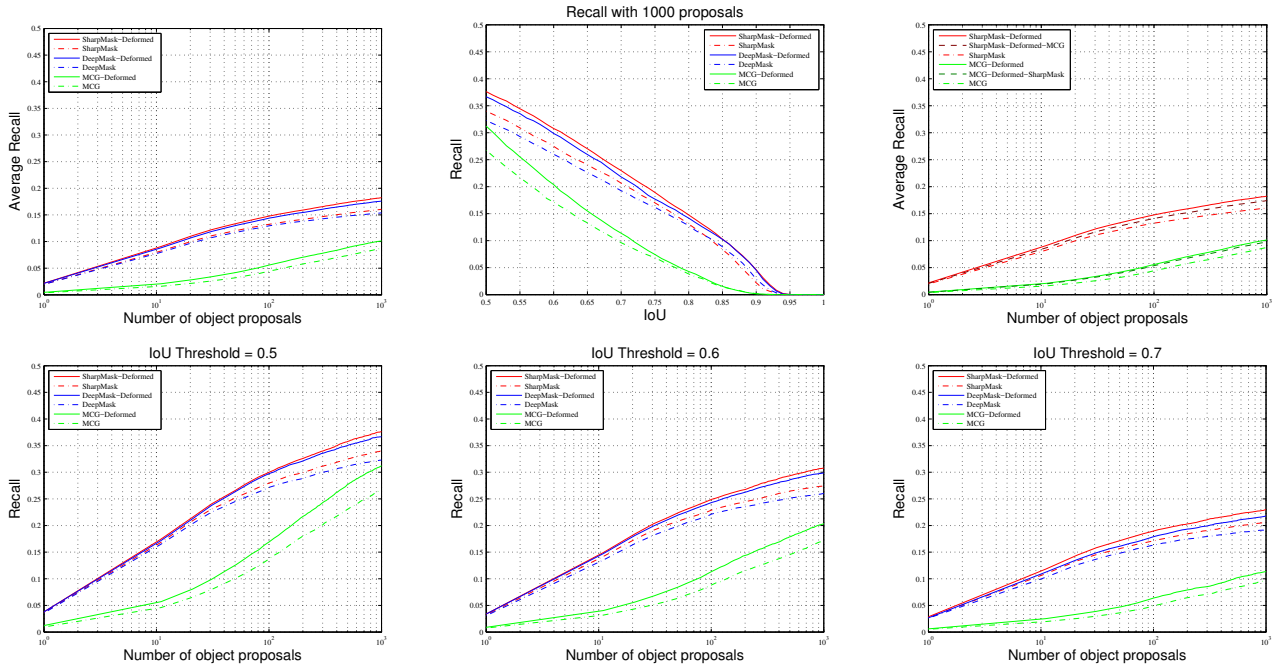

Figure 4. Results on **Cityscapes:** Top-left and Top-right: Average recall vs. number of proposals; Top-middle: Recall vs. different IoU thresholds for 1,000 proposals; Bottom: Recall vs. number of proposals under different IoU thresholds.

As our goal is to refine object segment proposals, we select three state-of-the-art segment proposal generation methods to produce the initial set of segmentation proposals, which include SharpMask [26], DeepMask [25] and MCG [27]. They are also considered as the baselines for our comparison. We apply the pre-trained MCG, DeepMask and SharpMask models provided by the authors to generate their results on the two datasets.

In order to test the efficacy of our method, we learn three affine transformation regression networks for SharpMask, DeepMask and MCG respectively and apply them to the corresponding methods. Moreover, we verify the generality of our learned regression networks by applying the learned network for SharpMask to MCG proposals and the learned network for MCG to SharpMask proposals.

4.3. Results

4.3.1 Cityscapes

In Figure 4 (top left panel), we first report the AR vs. number of proposals and comparisons to the baselines on the Cityscapes dataset. It shows that our approach consistently improves the quality of initial segment proposals generated by the three top-performing methods. We also achieve sizable performance gains over these baselines. In particular, with 1,000 proposals, our method boosts the AR of SharpMask, DeepMask and MCG from 0.160, 0.154 and 0.088 to 0.182, 0.176 and 0.101 respectively and the corresponding performance gains are 13.75%, 14.29% and 14.77%.

Method	AR@10	AR@100	AR@1000	AUC
SharpMask-Deformed	0.091	0.148	0.182	0.166
SharpMask	0.082	0.133	0.160	0.147
DeepMask-Deformed	0.088	0.144	0.176	0.161
DeepMask	0.080	0.130	0.154	0.143
MCG-Deformed	0.021	0.056	0.101	0.082
MCG	0.016	0.045	0.087	0.069

Table 2. Quantitative results on **Cityscapes**: AR at different number of proposals (10, 100 and 1,000) and AUC (AR averaged across all proposal counts).

We also report the recall across different IoU thresholds with 1,000 proposals in Figure 4 (top middle panel), which evidences that our method is capable of refining the object segmentation proposals with different qualities while maintaining the quality of segments with high IoU scores.

In Figure 4 (top right panel), we compare the performances (AR vs. number of proposals) of our networks when applying them to the proposals from the original initial method and a different one. We can see that the AR (0.174 for SharpMask and 0.097 for MCG) obtained by applying the learned network to the other initial proposals are just slightly lower than the original ones (0.182 and 0.101), which demonstrates the generality of our learned network.

The remaining plots in Figure 4 describe the recalls of baselines and our method when varying the number of object proposals under different IoU thresholds. Again, they

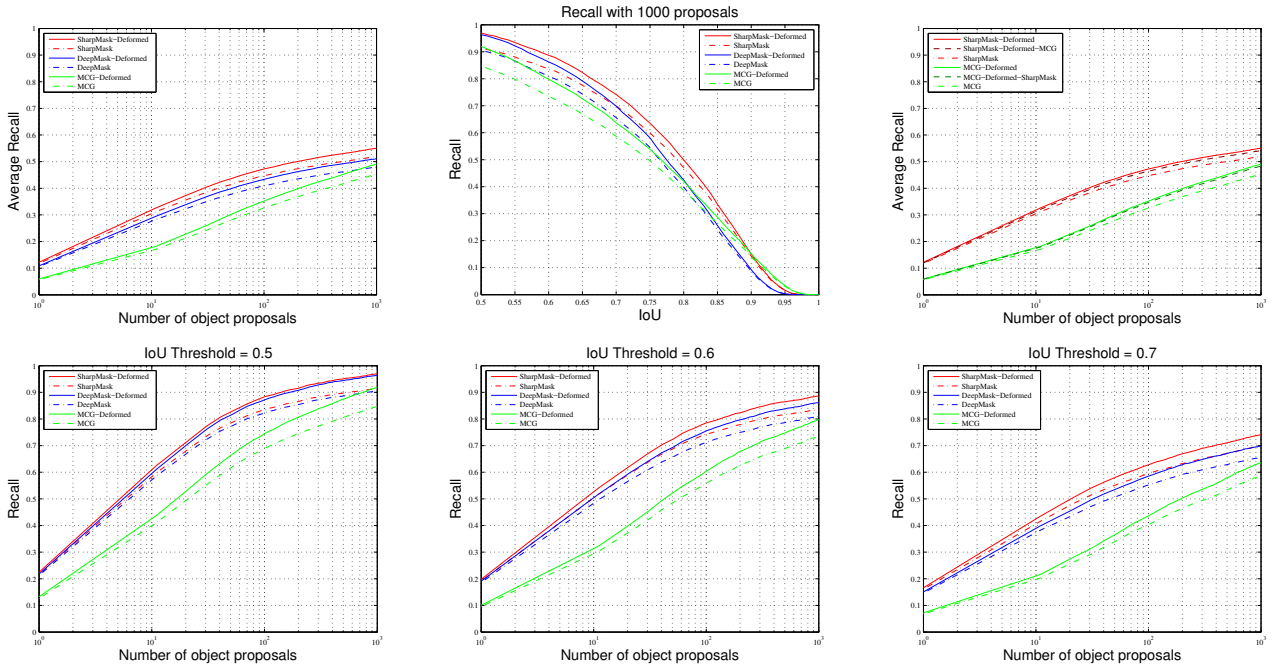

Figure 5. Results on **PASCAL VOC:** Top-left and Top-right: Average recall vs. number of proposals; Top-middle: Recall vs. different IoU thresholds for 1,000 proposals; Bottom: Recall vs. number of proposals under different IoU thresholds.

show that our approach can consistently enhance the quality of the initial object proposals across different IoU thresholds and with different number of proposals. For example, when the IoU threshold being 0.5, the recall improvements for SharpMask, DeepMask and MCG are 10.59% (from 0.340 to 0.376), 13.62% (from 0.323 to 0.367) and 14.93% (from 0.268 to 0.308) respectively.

More detailed quantitative results for the Cityscapes dataset are shown in Table 2, where we report the AR at three settings with different selected numbers of proposals, and the averaged AR across all proposal numbers (AUC). In addition, we show some qualitative examples of the mask refinement on the Cityscapes dataset in Figure 6. We note that our method is able to warp the initial segment masks towards the groundtruth objects, including translation (top), expansion (middle) and shrinkage (bottom).

4.3.2 PASCAL VOC

We report the AR vs. the number of object proposals in Figure 5 (top left panel), which shows that our approach can improve the AR metric for three baseline methods on the PSACAL VOC dataset as well. Specifically, for the setting of 1,000 proposals, our method increases the AR of SharpMask, DeepMask and MCG by 6.17% (from 0.519 to 0.551), 6.68% (from 0.479 to 0.511) and 8.39% (from 0.453 to 0.491) respectively. We note that the quantitative improvements on the PASCAL VOC are less than those on the Cityscapes. One possible reason is that the performance

Method	AR@10	AR@100	AR@1000	AUC
SharpMask-Deformed	0.321	0.473	0.551	0.514
SharpMask	0.307	0.447	0.519	0.486
DeepMask-Deformed	0.292	0.434	0.511	0.476
DeepMask	0.281	0.409	0.479	0.447
MCG-Deformed	0.182	0.353	0.491	0.430
MCG	0.170	0.327	0.453	0.396

Table 3. Quantitative results on **PASCAL VOC:** AR at different number of proposals (10, 100 and 1,000) and AUC (AR averaged across all proposal counts).

of these three methods on the PASCAL VOC is better than theirs on the Cityscapes, leading to a narrower margin for improvement.

The top middle panel in Figure 5 shows the recall changes across different IoU thresholds with 1,000 proposals. Again, we can see that the improvement for the initial object proposals is evident.

In the top right panel of Figure 5, we compare the original results with the ones obtained by applying the learned network to a different initial proposal method in terms of AR vs. number of proposals. The results clearly show the generality of our networks w.r.t. the initial proposal set.

Similarly, the remaining plots in Figure 5 show the recall improvement under different IoU thresholds when varying the number of proposals. It demonstrates again that our approach can consistently improve the quality of the original

IoU Interval	[0.3, 0.5)	[0.5, 0.6)	[0.6, 0.7)	[0.7, 0.8)
mean PGIoU	0.386	0.548	0.648	0.75
mean RGIoU	0.431	0.599	0.698	0.784
Gain	11.63%	9.42%	7.84%	4.57%
mean PGIoU	0.388	0.549	0.649	0.748
mean RGIoU	0.421	0.579	0.669	0.763
Gain	8.56%	5.6%	3.18%	1.95%

Table 4. Statistics for the improvements in the quality of DeepMask proposals with different initial IoU scores on **Cityscapes** (Top) and **PASCAL VOC** (Bottom).

object proposals across the range of all different settings.

We report the detailed quantitative results for the PASCAL VOC in Table 3, which describes the AR at three settings with selected numbers of proposals and the averaged AR across all proposal numbers (AUC). Finally, some qualitative examples of the mask refinement on the PASCAL VOC dataset are shown in Figure 7. Again, we can see our method achieves better region alignment for a variety of scenarios.

4.3.3 Ablation Study

To gain more insight into our approach, we conduct an ablation study by computing the improvements in the quality of DeepMask proposals with different initial IoU scores on two datasets. We first divide the initial proposals set into 4 groups, which correspond to the IoU intervals of [0.3, 0.5), [0.5, 0.6), [0.6, 0.7) and [0.7, 0.8). We then compute the mean IoU improvements for each group after warping the initial proposals through our method, which are shown in Table 4. The results show that our method is more effective on correcting large errors than obtaining fine-grained details, which is most likely due to the coarse-level warping generated by the affine transformations.

5. Conclusion

In this paper, we propose a novel method for refining object segment proposals, which can generate object segment candidates with better quality for instance-level semantic segmentation. The main contribution of our work is to formulate the refinement as a regression problem that estimates 2D affine transformations to warp the initial segment masks towards groundtruth objects. We design and train a deep network to predict the affine transformation parameters based on a new mask pooling strategy defined on hypercolumn features. Extensive experimental evaluations on two challenging datasets, the Cityscapes and the PASCAL VOC, demonstrate that our approach can consistently achieve sizable improvements on the IoU quality of the object segment proposals over three state-of-the-art methods.

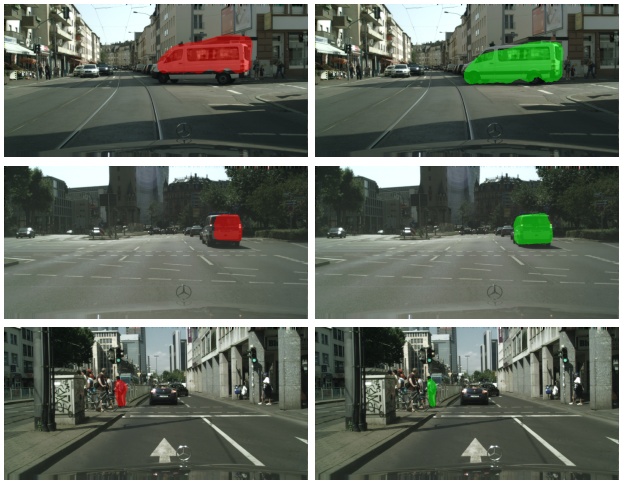

Figure 6. Qualitative results on **Cityscapes**. Red: original proposal's mask. Green: transformed mask.

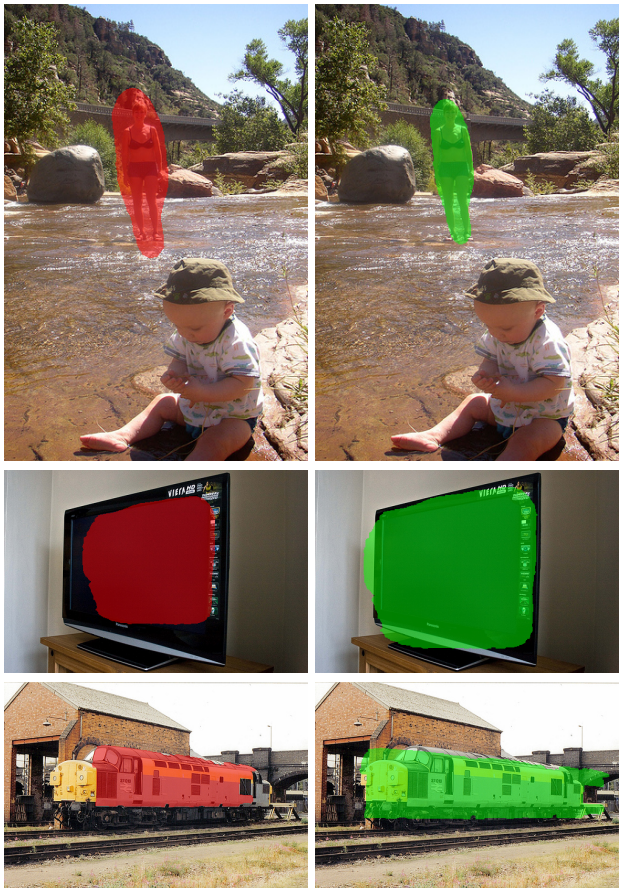

Figure 7. Qualitative results on **PASCAl VOC**. Red: original proposal's mask. Green: transformed mask.

Acknowledgment Data61 is part of the Commonwealth Scientific and Industrial Research Organisation (CSIRO) which is the federal government agency for scientific research in Australia. The Tesla K40 used for this research was donated by the NVIDIA Corporation.

References

[1] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1312–1328, 2012.

[2] X. Chen, H. Ma, X. Wang, and Z. Zhao. Improving object proposals with multi-thresholding straddling expansion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2587–2595, 2015.

[3] 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. *arXiv preprint arXiv:1604.01685*, 2016.

[4] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3992–4000, 2015.

[5] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. *arXiv preprint arXiv:1512.04412*, 2015.

[6] I. Endres and D. Hoiem. Category independent object proposals. In *Computer Vision–ECCV 2010*, pages 575–588. Springer, 2010.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015.

[10] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[11] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011.

[12] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015.

[13] X. He and S. Gould. An exemplar-based crf for multi-instance object segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 296–303. IEEE, 2014.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[15] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[16] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *Computer Vision–ECCV 2014*, pages 725–739. Springer, 2014.

[17] D.-J. Kroon. B-spline grid, image and point based registration. Matlabcentral No.20057, 2011.

[18] K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3d scenes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1330–1337. IEEE, 2012.

[19] T. Lee, S. Fidler, and S. Dickinson. Learning to combine mid-level cues for object proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1680–1688, 2015.

[20] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. *arXiv preprint arXiv:1511.08498*, 2015.

[21] 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 *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[22] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[23] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[24] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[25] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1981–1989, 2015.

[26] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollar. Learning to refine object segments. 2016.

[27] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. In *arXiv:1503.00848*, March 2015.

[28] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2417–2424, 2014.

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[30] B. Romera-Paredes and P. H. Torr. Recurrent instance segmentation. *arXiv preprint arXiv:1511.08250*, 2015.

[31] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE transactions on medical imaging*, 18(8):712–721, 1999.

[32] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[33] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, pages 689–692. ACM, 2015.

[34] V. Yanulevskaya, J. Uijlings, and N. Sebe. Learning to group objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3134–3141, 2014.

[35] X. Yu, F. Zhou, and M. Chandraker. Deep deformation network for object landmark localization. *arXiv preprint arXiv:1605.01014*, 2016.

[36] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014.