Object Contour Detection with a Fully Convolutional Encoder-Decoder Network

Jimei Yang
Adobe Research
jimyang@adobe.com

Brian Price
Adobe Research
bprice@adobe.com

Scott Cohen
Adobe Research
scohen@adobe.com

Honglak Lee
University of Michigan, Ann Arbor
honglak@umich.edu

Ming-Hsuan Yang
UC Merced
mhyang@ucmerced.edu

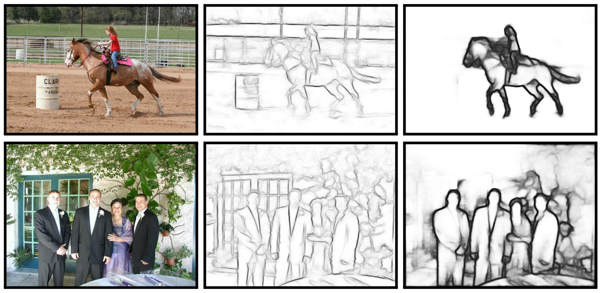
Abstract

We develop a deep learning algorithm for contour detection with a fully convolutional encoder-decoder network. Different from previous low-level edge detection, our algorithm focuses on detecting higher-level object contours. Our network is trained end-to-end on PASCAL VOC with refined ground truth from inaccurate polygon annotations, yielding much higher precision in object contour detection than previous methods. We find that the learned model generalizes well to unseen object classes from the same super-categories on MS COCO and can match state-of-the-art edge detection on BSDS500 with fine-tuning. By combining with the multiscale combinatorial grouping algorithm, our method can generate high-quality segmented object proposals, which significantly advance the state-of-the-art on PASCAL VOC (improving average recall from 0.62 to 0.67) with a relatively small amount of candidates (∼1660 per image).

1. Introduction

Object contour detection is fundamental for numerous vision tasks. For example, it can be used for image segmentation [41, 3], for object detection [15, 18], and for occlusion and depth reasoning [20, 2]. Given its axiomatic importance, however, we find that object contour detection is relatively under-explored in the literature. At the same time, many works have been devoted to edge detection that responds to both foreground objects and background boundaries (Figure 1 (b)). In this paper, we address "object-only" contour detection that is expected to suppress background boundaries (Figure 1(c)).

Edge detection has a long history. Early research focused on designing simple filters to detect pixels with highest gradients in their local neighborhood, e.g. Sobel [16] and Canny [8]. The main problem with filter based methods is that they only look at the color or brightness differences between adjacent pixels but cannot tell the texture differences in a larger receptive field. With the advance of texture descriptors [35], Martin et al. [37] combined color, brightness and texture gradients in their probabilistic boundary detector. Arbelaez et al. [3] further improved upon this by computing local cues from multiscale and spectral clustering, known as *gPb*, which yields state-of-the-art accuracy. However, the globalization step of *gPb* significantly increases the computational load. Lim and Dollar [30, 12] analyzed the clustering structure of local contour maps and developed efficient supervised learning algorithms for fast edge detection [12]. These efforts lift edge detection to a higher abstract level, but still fall below human perception due to their lack of object-level knowledge.



(a) Image (b) [12] Ours

Figure 1. Object contour detection. Given input images (a), our model can effectively learn to detect contours of foreground objects (c) in contrast to traditional edge detection (b).

Recently deep convolutional networks [29] have demonstrated remarkable ability of learning high-level representations for object recognition [18, 10]. These learned features have been adopted to detect natural image edges [25, 6, 43, 47] and yield a new state-of-the-art performance [47]. All these methods require training on ground truth contour annotations. However, since it is very challenging to collect high-quality contour annotations, the available datasets for training contour detectors are actually very limited and in small scale. For example, the standard benchmarks, Berkeley segmentation (BSDS500) [36] and NYU depth v2 (NYUDv2) [44] datasets only include 200 and 381 training images, respectively. Therefore, the representation power of deep convolutional networks has not been entirely harnessed for contour detection. In this paper, we scale up the training set of deep learning based contour detection to more than 10k images on PASCAL VOC [14]. To address the quality issue of ground truth contour annotations, we develop a dense CRF [26] based method to refine the object segmentation masks from polygons.

Given image-contour pairs, we formulate object contour detection as an image labeling problem. Inspired by the success of fully convolutional networks [34] and deconvolutional networks [38] on semantic segmentation, we develop a fully convolutional encoder-decoder network (CEDN). Being fully convolutional, our CEDN network can operate on arbitrary image size and the encoder-decoder network emphasizes its asymmetric structure that differs from deconvolutional network [38]. We initialize our encoder with VGG-16 net [45] (up to the "fc6" layer) and to achieve dense prediction of image size our decoder is constructed by alternating unpooling and convolution layers where unpooling layers re-use the switches from max-pooling layers of encoder to upscale the feature maps. During training, we fix the encoder parameters (VGG-16) and only optimize decoder parameters. This allows the encoder to maintain its generalization ability so that the learned decoder network can be easily combined with other tasks, such as bounding box regression or semantic segmentation.

We evaluate the trained network on unseen object categories from BSDS500 and MS COCO datasets [31], and find the network generalizes well to objects in similar "super-categories" to those in the training set, e.g. it generalizes to objects like "bear" in the "animal" super-category since "dog" and "cat" are in the training set. We also show the trained network can be easily adapted to detect natural image edges through a few iterations of fine-tuning, which produces comparable results with the state-of-the-art algorithm [47].

An immediate application of contour detection is generating object proposals. Previous literature has investigated various methods of generating bounding box or segmented object proposals by scoring edge features [49, 11] and combinatorial grouping [46, 9, 4] and etc. In this paper, we use a multiscale combinatorial grouping (MCG) algorithm [4] to generate segmented object proposals from our contour detection. As a result, our method significantly improves the quality of segmented object proposals on the PASCAL VOC 2012 validation set, achieving 0.67 average recall from overlap 0.5 to 1.0 with only about 1660 candidates per image, compared to the state-of-the-art average recall 0.62 by original *gPb*-based MCG algorithm with near 5140 candidates per image. We also evaluate object proposals on the MS COCO dataset with 80 object classes and analyze the average recalls from different object classes and their super-categories.

The key contributions are summarized below:

- We develop a simple yet effective fully convolutional encoder-decoder network for object contour prediction and the trained model generalizes well to unseen object classes from the same super-categories, yielding significantly higher precision in object contour detection than previous methods.
- We show we can fine tune our network for edge detection and match the state-of-the-art in terms of precision and recall.
- We generate accurate object contours from imperfect polygon based segmentation annotations, which makes it possible to train an object contour detector at scale.
- Our method obtains state-of-the-art results on segmented object proposals by integrating with combinatorial grouping [4].

2. Related Work

Semantic contour detection. Hariharan et al. [19] study top-down contour detection problem. Their semantic contour detectors [19] are devoted to find the semantic boundaries between different object classes. Although they consider object instance contours while collecting annotations, they choose to ignore the occlusion boundaries between object instances from the same class. As combining bottom-up edges with object detector output, their method can be extended to object instance contours but might encounter challenges of generalizing to unseen object classes.

Occlusion boundary detection. Hoiem et al. [20] study the problem of recovering occlusion boundaries from a single image. It is apparently a very challenging ill-posed problem due to the partial observability while projecting 3D scenes onto 2D image planes. They formulate a CRF model to integrate various cues: color, position, edges, surface orientation and depth estimates. We believe our instance-level object contours will provide another strong cue for addressing this problem that is worth investigating in the future.

Object proposal generation. There is a large body of works on generating bounding box or segmented object proposals. Hosang et al. [21] and Jordi et al. [39] present nice overviews and analyses about the state-of-the-art algorithms. Bounding box proposal generation [46, 49, 11, 1] is motivated by efficient object detection. One of their drawbacks is that bounding boxes usually cannot provide accurate object localization. More related to our work is generating segmented object proposals [4, 9, 13, 22, 24, 27, 40]. At the core of segmented object proposal algorithms is contour detection and superpixel segmentation. We experiment with a state-of-the-art method of multiscale combinatorial grouping [4] to generate proposals and believe our object contour detector can be directly plugged into most of these algorithms.

3. Object Contour Detection

In this section, we introduce our object contour detection method with the proposed fully convolutional encoder-decoder network.

3.1. Fully Convolutional Encoder-Decoder Network

We formulate contour detection as a binary image labeling problem where "1" and "0" indicates "contour" and "non-contour", respectively. Image labeling is a task that requires both high-level knowledge and low-level cues. Given the success of deep convolutional networks [29] for learning rich feature hierarchies, image labeling has been greatly advanced, especially on the task of semantic segmentation [10, 34, 32, 48, 38, 33]. Among those end-to-end methods, fully convolutional networks [34] scale well up to the image size but cannot produce very accurate labeling boundaries; unpooling layers help deconvolutional networks [38] to generate better label localization but their symmetric structure introduces a heavy decoder network which is difficult to train with limited samples.

We borrow the ideas of full convolution and unpooling from above two works and develop a fully convolutional encoder-decoder network for object contour detection. The network architecture is demonstrated in Figure 2. We use the layers up to "fc6" from VGG-16 net [45] as our encoder. Since we convert the "fc6" to be convolutional, so we name it "conv6" in our decoder.

Due to the asymmetric nature of image labeling problems (image input and mask output), we break the symmetric structure of deconvolutional networks and introduce a light-weighted decoder. The first layer of decoder "deconv6" is designed for dimension reduction that projects 4096-d "conv6" to 512-d with 1×1 kernel so that we can re-use the pooling switches from "conv5" to upscale the feature maps by twice in the following "deconv5" layer. The number of channels of every decoder layer is properly designed to allow unpooling from its corresponding max-pooling layer. All the decoder convolution layers except "deconv6" use 5×5 kernels. All the decoder convolution layers except the one next to the output label are followed by relu activation function. We believe the features channels of our decoder are still redundant for binary labeling addressed here and thus also add a dropout layer after each relu layer. A complete decoder network setup is listed in Table 1 and the loss function is simply the pixel-wise logistic loss.

Table 1. Decoder network setup.

name	deconv6	deconv5	deconv4	
setup	conv	unpool-conv	unpool-conv	
kernel	$1\times1\times512$	$5\times5\times512$	$5\times5\times256$	
acti	relu	relu	relu	
name	deconv3	deconv2	deconv1	pred
setup	unpool-conv	unpool-conv	unpool-conv	conv
kernel	$5\times5\times128$	$5\times5\times64$	$5\times5\times32$	$5\times5\times1$
activation	relu	relu	relu	sigmoid

3.2. Contour Ground Truth Refinement

Drawing detailed and accurate contours of objects is a challenging task for human beings. This is why many large scale segmentation datasets [42, 14, 31] provide contour annotations with polygons as they are less expensive to collect at scale. However, because of unpredictable behaviors of human annotators and limitations of polygon representation, the annotated contours usually do not align well with the true image boundaries and thus cannot be directly used as ground truth for training. Among all, the PASCAL VOC dataset is a widely-accepted benchmark with high-quality annotation for object segmentation. VOC 2012 release includes 11540 images from 20 classes covering a majority of common objects from categories such as "person", "vehicle", "animal" and "household", where 1464 and 1449 images are annotated with object instance contours for training and validation. Hariharan et al. [19] further contribute more than 10000 high-quality annotations to the remaining images. Together there are 10582 images for training and 1449 images for validation (the exact 2012 validation set). We choose this dataset for training our object contour detector with the proposed fully convolutional encoder-decoder network.

The original PASCAL VOC annotations leave a thin unlabeled (or uncertain) area between occluded objects (Figure 3(b)). To find the high-fidelity contour ground truth for training, we need to align the annotated contours with the true image boundaries. We consider contour alignment as a multi-class labeling problem and introduce a dense CRF model [26] where every instance (or background) is assigned with one unique label. The dense CRF optimization then fills the uncertain area with neighboring instance labels so that we obtain refined contours at the labeling boundaries (Figure 3(d)). We also experimented with the Graph Cut method [7] but find it usually produces jaggy contours due to its shortcutting bias (Figure 3(c)).

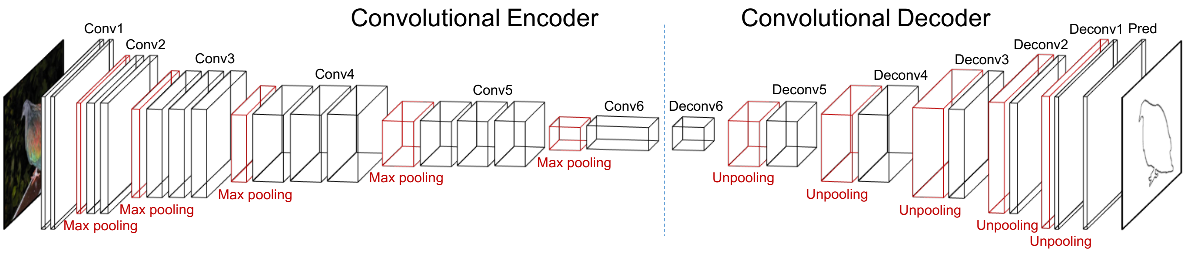
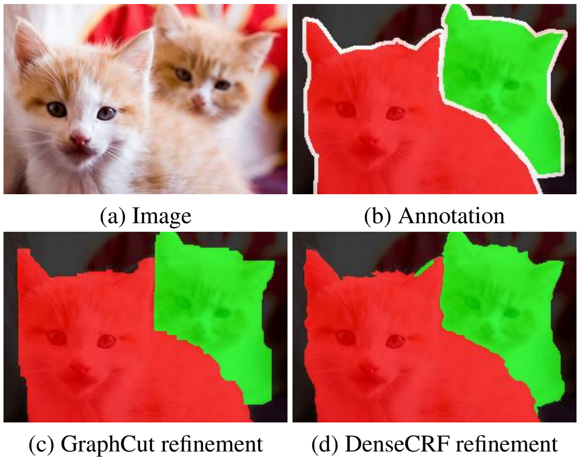


Figure 2. Architecture of the proposed fully convolutional encoder-decoder network.



(a) Image (b) Annotation

(c) GraphCut refinement (d) DenseCRF refinement

Figure 3. Contour refinement. The polygon based annotations (a) cannot be directly used for training due to its inaccurate boundaries (thin white area reflects unlabeled pixels between objects). We align them to image boundaries by re-labeling the uncertain areas with dense CRF (d), compared to Graph Cut (c).

3.3. Training

We train the network using *Caffe* [23]. For each training image, we randomly crop four $224 \times 224 \times 3$ patches and together with their mirrored ones compose a $224 \times 224 \times 3 \times 8$ minibatch. The ground truth contour mask is processed in the same way. We initialize the encoder with pre-trained VGG-16 net and the decoder with random values. During training, we fix the encoder parameters and only optimize the decoder parameters. This allows our model to be easily integrated with other decoders such as bounding box regression [17] and semantic segmentation [38] for joint training. As the "contour" and "non-contour" pixels are extremely imbalanced in each minibatch, the penalty for being "contour" is set to be 10 times the penalty for being "non-contour". We use the Adam method [5] to optimize the network parameters and find it is more efficient than standard stochastic gradient descent. We set the learning rate to 10^{-4} and train the network with 30 epochs with all the training images being processed each epoch. Note that we fix the training patch to 224×224 for memory efficiency and the learned parameters can be used on images of arbitrary size because of its fully convolutional nature.

4. Results

In this section, we evaluate our method on contour detection and proposal generation using three datasets: PASCAL VOC 2012, BSDS500 and MS COCO. We will explain the details of generating object proposals using our method after the contour detection evaluation. More evaluation results are in the supplementary materials.

4.1. Contour Detection

Given trained models, all the test images are fed-forward through our CEDN network in their original sizes to produce contour detection maps. The detection accuracies are evaluated by four measures: F-measure (F), fixed contour threshold (ODS), per-image best threshold (OIS) and average precision (AP). Note that a standard non-maximum suppression is used to clean up the predicted contour maps (thinning the contours) before evaluation.

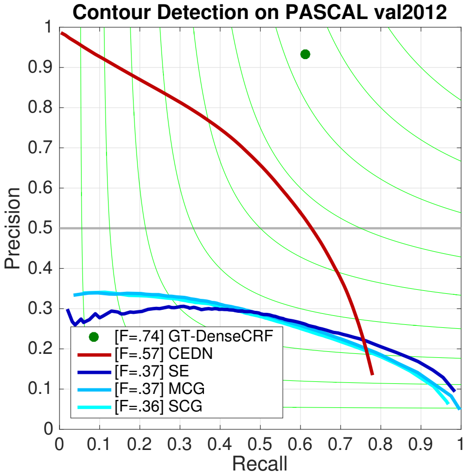


Figure 4. PR curve for contour detection on the PASCAL VOC 2012 validation set.

PASCAL val2012. We first present results on the PASCAL VOC 2012 validation set, shortly "PASCAL val2012", with comparisons to three baselines, structured edge detection (SE) [12], singlescale combinatorial grouping (SCG) and multiscale combinatorial grouping (MCG) [4]. Precision-recall curves are shown in Figure 4. Note that we

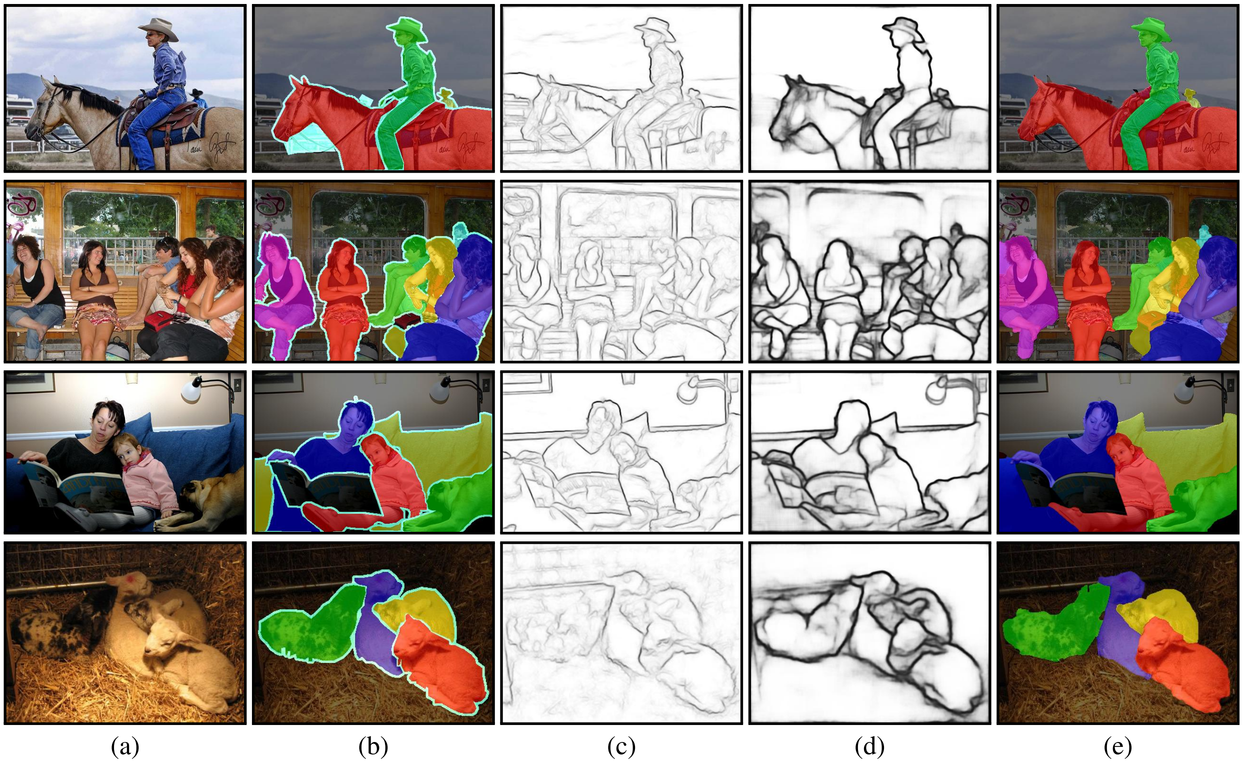


Figure 5. Example results on PASCAL VOC val2012. In each row from left to right we present (a) input image, (b) ground truth annotation, (c) edge detection [12], (d) our object contour detection and (e) our best object proposals.

use the originally annotated contours instead of our refined ones as ground truth for unbiased evaluation. Accordingly we consider the refined contours as the upper bound since our network is learned from them. Its precision-recall value is referred as "GT-DenseCRF" with a green spot in Figure 4. Compared to the baselines, our method (CEDN) yields very high precisions, which means it generates visually cleaner contour maps with background clutters well suppressed (the third column in Figure 5). Note that the occlusion boundaries between two instances from the same class are also well recovered by our method (the second example in Figure 5). We also note that there is still a big performance gap between our current method (F=0.57) and the upper bound (F=0.74), which requires further research for improvement.

BSDS500 with fine-tuning. BSDS500 [36] is a standard benchmark for contour detection. Different from our object-centric goal, this dataset is designed for evaluating natural edge detection that includes not only object contours but also object interior boundaries and background boundaries (examples in Figure 6(b)). It includes 500 natural images with carefully annotated boundaries collected from multiple users. The dataset is divided into three parts: 200 for training, 100 for validation and the rest 200 for test. We first examine how well our CEDN model trained on PASCAL VOC can generalize to unseen object categories in this dataset. Interestingly, as shown in the Figure 6(c), most of wild animal contours, e.g. elephants and fish are accurately detected and meanwhile the background bound-

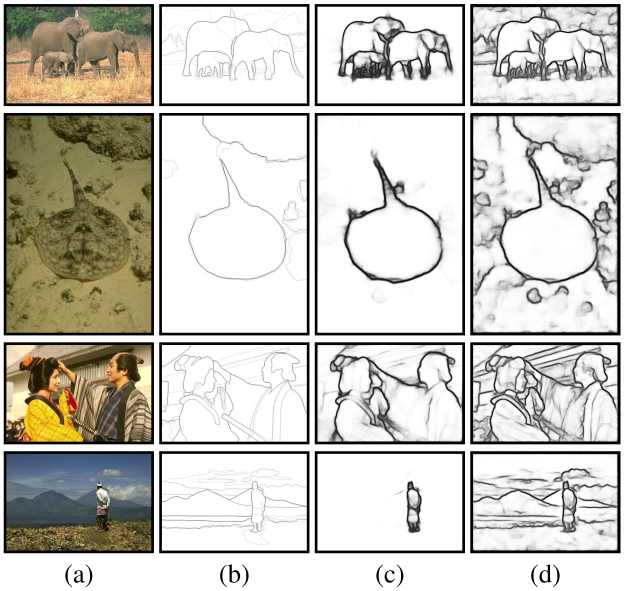


Figure 6. Example results on BSDS500 test set. In each row from left to right we present (a) input image, (b) ground truth contour, (c) contour detection with pretrained CEDN and (d) contour detection with fine-tuned CEDN.

aries, e.g. building and mountains are clearly suppressed. We further fine-tune our CEDN model on the 200 training images from BSDS500 with a small learning rate (10^{-5}) for 100 epochs. As a result, the boundaries suppressed by pretrained CEDN model ("CEDN-pretrain") re-surface from the scenes. Quantitatively, we evaluate both the pretrained and fine-tuned models on the test set in comparisons with previous methods. Figure 7 shows that 1) the pretrained CEDN model yields a high precision but a low recall due to its object-selective nature and 2) the fine-tuned CEDN model achieves comparable performance (F=0.79) with the state-of-the-art method (HED) [47]. Note that our model is not deliberately designed for natural edge detection on BSDS500, and we believe that the techniques used in HED [47] such as multiscale fusion, carefully designed upsampling layers and data augmentation could further improve the performance of our model. A more detailed comparison is listed in Table 2.

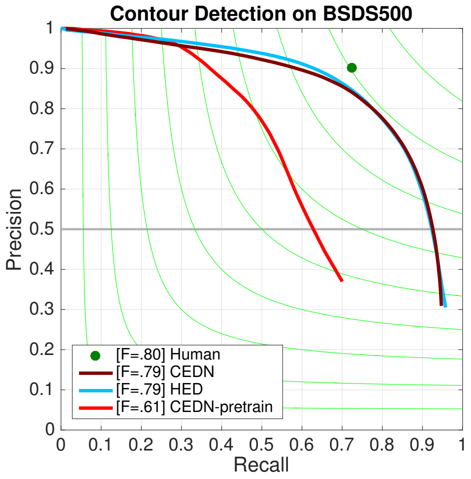


Figure 7. PR curve for contour detection on the BSDS500 set set.

Table 2. Contour detection results on BSDS500.

	ODS	OIS	AP
Human	.80	.80	-
SCG [4]	.739	.758	.773
SE [12]	.746	.767	.803
DeepEdge [6]	.753	.772	.807
DeepContour [43]	.756	.773	.797
HED [47]	.782	.804	.833
HED-new [1]	**.788**	**.808**	**.840**
CEDN-pretrain	.610	.635	.580
CEDN	**.788**	.804	.821

4.2. Object Proposal Generation

Object proposals are important mid-level representations in computer vision. Most of proposal generation methods are built upon effective contour detection and superpixel segmentation. Thus the improvements on contour detection will immediately boost the performance of object proposals. We choose the MCG algorithm to generate segmented object proposals from our detected contours. The MCG algorithm is based the classic *gPb* contour detector. It first computes ultrametric contour maps from multiscale and then aligns them into a single hierarchical segmentation. To obtain object proposals, a multi-objective optimization is designed to reduce the redundancy of combinatorial grouping of adjacent regions. The reduced set of grouping candidates are then ranked according to their low-level features as the final segmented object proposals. Note that the hierarchical segmentation can be obtained directly from single scale, which leads to a fast algorithm of singlescale combinatorial grouping (SCG). Based on the procedure above, we simply replace *gPb* with our CEDN contour detector to generate proposals. The multiscale and singlescale versions are referred to as "CEDNMCG" and "CEDNSCG", respectively.

We evaluate the quality of object proposals by two measures: Average Recall (AR) and Average Best Overlap (ABO). Both measures are based on the overlap (Jaccard index or Intersection-over-Union) between a proposal and a ground truth mask. AR is measured by 1) counting the percentage of objects with their best Jaccard above a certain threshold T and then 2) averaging them within a range of thresholds $T \in [0.5, 1.0]$. It is established in [21, 39] to benchmark the quality of bounding box and segmented object proposals. ABO is measured by calculating the best proposal's Jaccard for every ground truth object and then 2) averaging them over all the objects.

We compare with state-of-the-art algorithms: MCG, SCG, Category Independent object proposals (CI) [13], Constraint Parametric Min Cuts (CPMC) [9], Global and Local Search (GLS) [40], Geodesic Object Proposals (GOP) [27], Learning to Propose Objects (LPO) [28], Recycling Inference in Graph Cuts (RIGOR) [22], Selective Search (SeSe) [46] and Shape Sharing (ShSh) [24]. Note that these abbreviated names are inherited from [4].

PASCAL val2012. Figure 8 shows that CEDNMCG achieves 0.67 AR and 0.83 ABO with ~ 1660 proposals per image, which improves the second best MCG by 8% in AR and by 3% in ABO with a third as many proposals. It takes 0.1 second to compute the CEDN contour map for a PASCAL image on a high-end GPU and 18 seconds to generate proposals with MCG on a standard CPU. We notice that the CEDNSCG achieves similar accuracies with CEDNMCG, but it only takes less than 3 seconds to run SCG. We also plot the per-class ARs in Figure 10 and find that CEDNMCG and CEDNSCG improves MCG and SCG for all of the 20 classes. Notably, the bicycle class has the worst AR and we guess it is likely because of its incomplete annota-

[1] This is the latest result from `http://vcl.ucsd.edu/hed/`

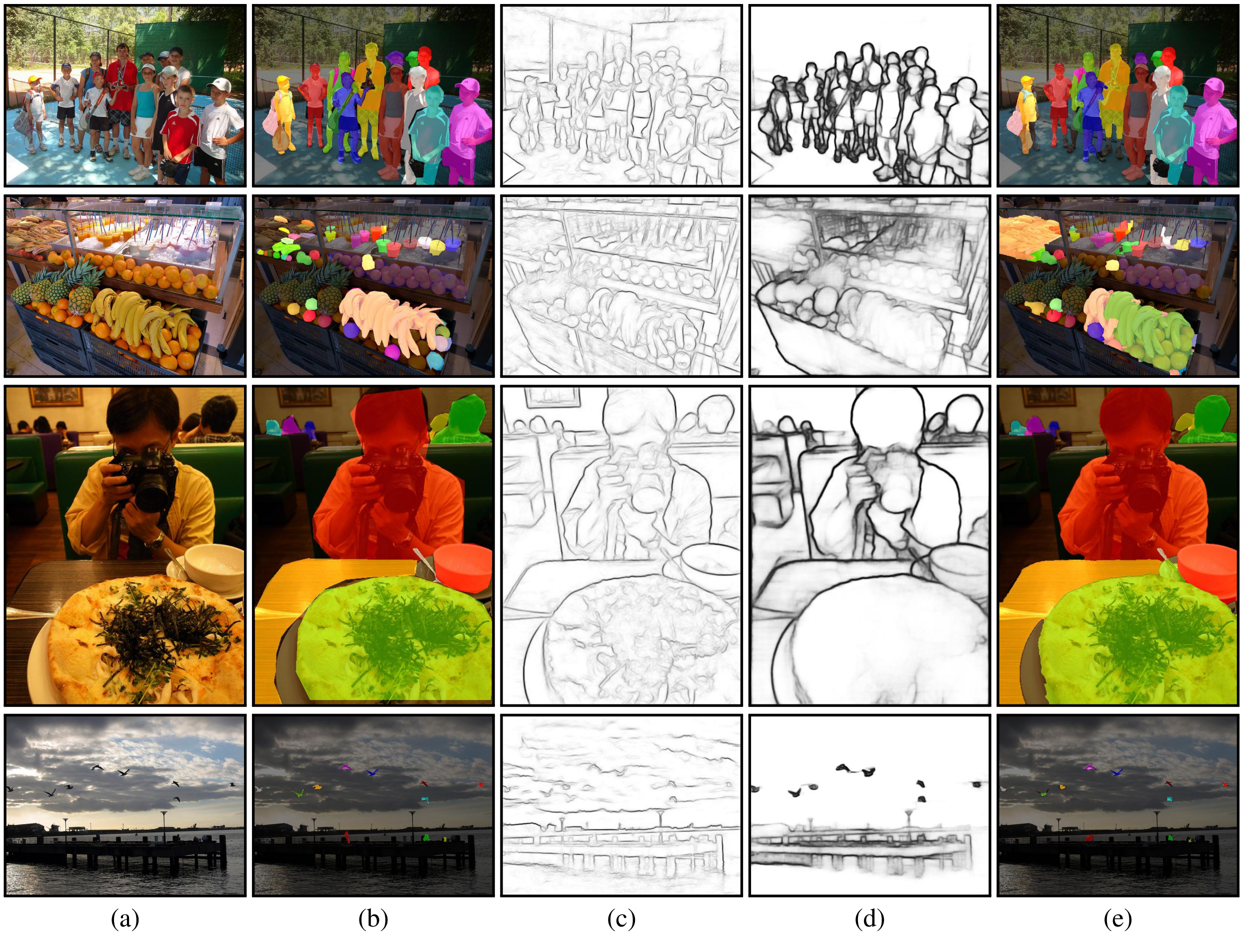


Figure 9. Example results on MS COCO val2014. In each row from left to right we present (a) input image, (b) ground truth annotation, (c) edge detection [12], (d) our object contour detection and (e) our best object proposals.

tions. Some examples of object proposals are demonstrated in Figure 5(d).

MS COCO val2014. We present results in the MS COCO 2014 validation set, shortly "COCO val2014" that includes 40504 images annotated by polygons from 80 object classes. This dataset is more challenging due to its large variations of object categories, contexts and scales. Compared to PASCAL VOC, there are 60 unseen object classes for our CEDN contour detector. Note that we did not train CEDN on MS COCO. We report the AR and ABO results in Figure 11. It turns out that the CEDNMCG achieves a competitive AR to MCG with a slightly lower recall from fewer proposals, but a weaker ABO than LPO, MCG and SeSe. Taking a closer look at the results, we find that our CEDNMCG algorithm can still perform well on *known* objects (first and third examples in Figure 9) but less effectively on certain *unknown* object classes, such as food (second example in Figure 9). It is likely because those novel classes, although seen in our training set (PASCAL VOC), are actually annotated as background. For example, there is a "dining table" class but no "food" class in the PASCAL VOC dataset. Quantitatively, we present per-class ARs in Figure 12 and have following observations:

- CEDN obtains good results on those classes that share common super-categories with PASCAL classes, such as "vehicle", "animal" and "furniture".
- CEDN fails to detect the objects labeled as "background" in the PASCAL VOC training set, such as "food" and "applicance".
- CEDN works well on unseen classes that are not prevalent in the PASCAL VOC training set, such as "sports".

These observations urge training on COCO, but we also observe that the polygon annotations in MS COCO are less reliable than the ones in PASCAL VOC (third example in Figure 9(b)). We will need more sophisticated methods for refining the COCO annotations.

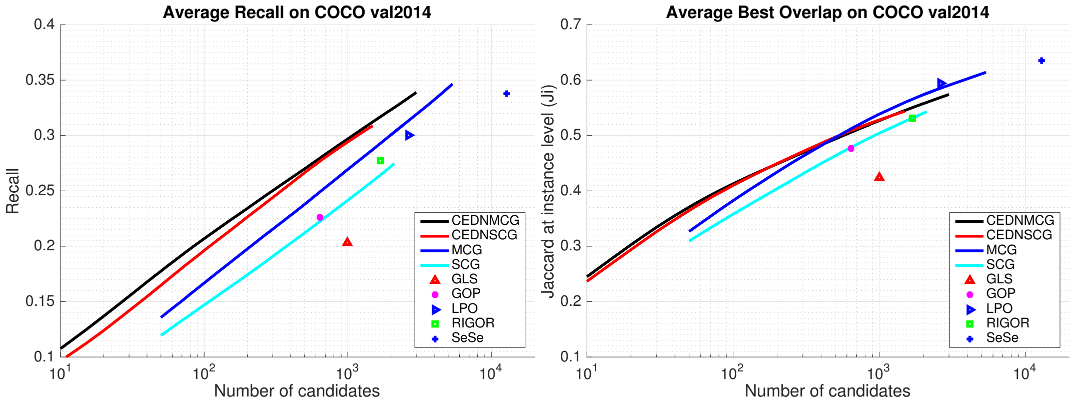


Figure 11. Average best overlap and average recall on the MS COCO 2014 validation set.

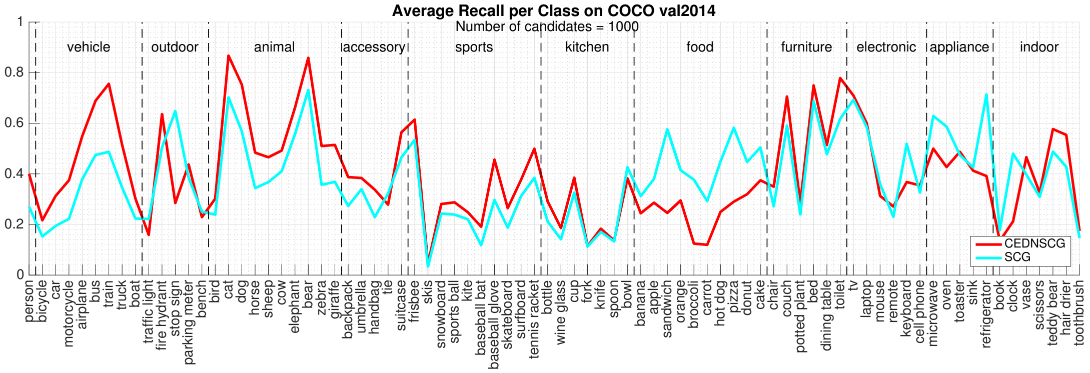


Figure 12. Average recall per class on the MS COCO 2014 validation set.

5. Conclusion and Future Work

We have developed an object-centric contour detection method using a simple yet efficient fully convolutional encoder-decoder network. Concerned with the imperfect contour annotations from polygons, we have developed a refinement method based on dense CRF so that the proposed network has been trained in an end-to-end manner. As a result, the trained model yielded high precision on PASCAL VOC and BSDS500, and has achieved comparable performance with the state-of-the-art on BSDS500 after fine-tuning. We have combined the proposed contour detector with multiscale combinatorial grouping algorithm for generating segmented object proposals, which significantly advances the state-of-the-art on PASCAL VOC. We also found that the proposed model generalizes well to unseen object classes from the known super-categories and demonstrated competitive performance on MS COCO without retraining the network.

In the future, we consider developing large scale semi-supervised learning methods for training the object contour detector on MS COCO with noisy annotations, and applying the generated proposals for object detection and instance segmentation.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34:2189–2202, 2012.

[2] M. R. Amer, S. Yousefi, R. Raich, and S. Todorovic. Monocular extraction of 2.1 D sketch using constrained convex optimization. *IJCV*, 112(1):23–42, 2015.

[3] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

[4] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014.

[5] J. Ba and D. Kingma. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[6] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*, 2015.

[7] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *ICCV*, 2001.

[8] J. Canny. A computational approach to edge detection. *PAMI*, (6):679–698, 1986.

[9] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep con-

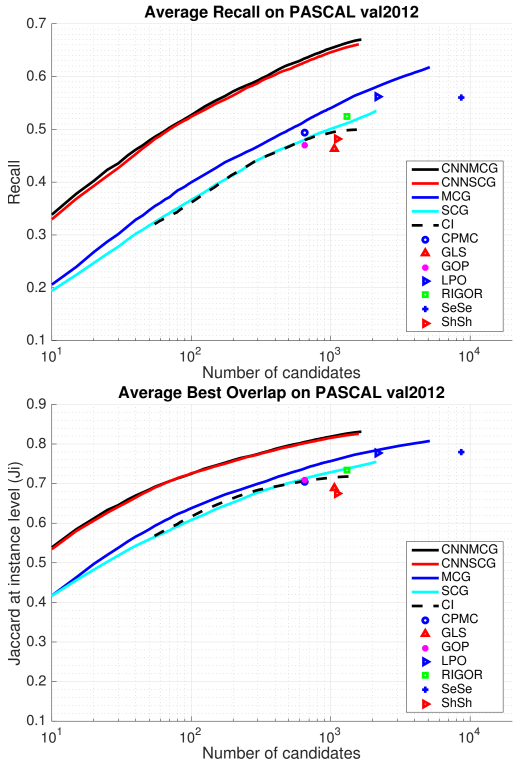


Figure 8. Average best overlap and average recall on the PASCAL VOC 2012 validation set.

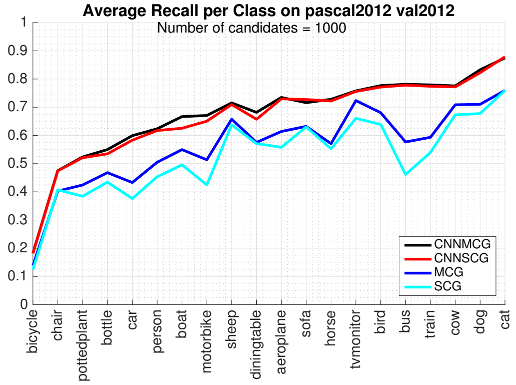


Figure 10. Per-class average recall on the PASCAL VOC 2012 validation set.

volutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[11] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014.

[12] P. Dollár and C. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[13] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.

[14] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.

[15] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *PAMI*, 30(1):36–51, 2008.

[16] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Pearson Education, 2003.

[17] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

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

[19] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[20] D. Hoiem, A. N. Stein, A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *ICCV*, 2007.

[21] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *PAMI*, 2015.

[22] A. Humayun, F. Li, and J. M. Rehg. RIGOR: Reusing inference in graph cuts for generating object regions. In *CVPR*, 2014.

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[24] J. Kim and K. Grauman. Shape sharing for object segmentation. In *ECCV*, 2012.

[25] J. J. Kivinen, C. K. Williams, and N. Heess. Visual boundary prediction: A deep neural prediction network and quality dissection. In *AISTATS*, 2014.

[26] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *NIPS*, 2011.

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

[28] P. Krähenbühl and V. Koltun. Learning to propose objects. In *CVPR*, 2015.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[30] J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013.

[31] 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*, 2014.

[32] S. Liu, J. Yang, C. Huang, and M.-H. Yang. Multi-objective convolutional learning for face labeling. In *CVPR*, 2015.

[33] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015.

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

[35] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 2001.

[36] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[37] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.

[38] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[39] J. Pont-Tuset and L. J. V. Gool. Boosting object proposals: From Pascal to COCO. In *ICCV*, 2015.

[40] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. In *CVPR*, 2014.

[41] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, 2004.

[42] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *IJCV*, 2008.

[43] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *CVPR*, 2015.

[44] N. Silberman, P. Kohli, D. Hoiem, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

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

[46] K. E. A. van de Sande, J. R. R. Uijlingsy, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.

[47] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015.

[48] 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.

[49] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edge. In *ECCV*, 2014.

Appendix

Comparing with HED trained on BSD and PASCAL VOC. We trained the HED model on PASCAL VOC using the same training data as our model with 30000 iterations. Its contour prediction precision-recall curve is illustrated in Figure 13 with comparisons to our CEDN model, the pre-trained HED model on BSDS (referred as HEDB) and others. It can be seen that the F-score of HED is improved (from 0.42 to 0.44) by training on PASCAL VOC but still significantly lower than CEDN (0.57). We will present visual examples in the revised submission. We further feed the HED edge maps into MCG for generating proposals and compare their average recalls with others in Figure 13. We refer the results from PASCAL-trained HED as HEDMCG and the ones from pre-trained HED on BSDS as HEDBMCG. Unfortunately, the HEDMCG does not improve upon HEDBMCG. With 1000 proposals, the average recalls of CEDNMCG, HEDMCG and HEDBMCG are about 0.65, 0.57 and 0.55, respectively.

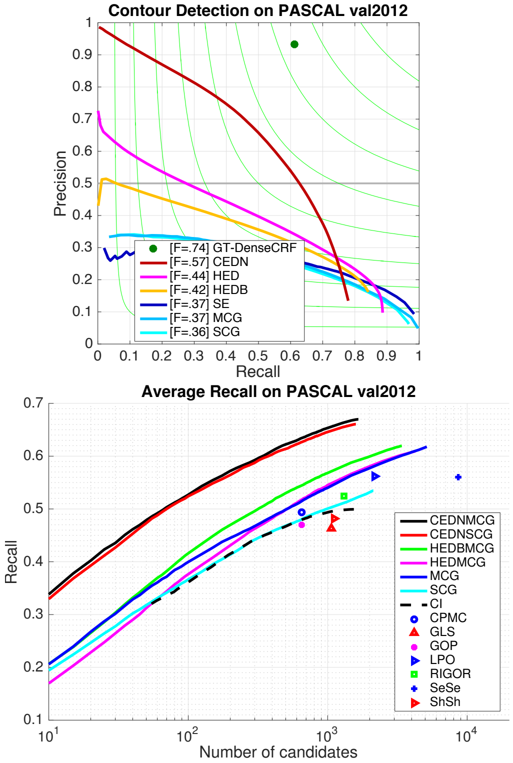


Figure 13. Full comparisons with HED on PASCAL val2012.