DeepCrop: Directed Object Segmentation with Deep Learning

Andrey Kurenkov

Abstract

1. Introduction

(a) (b)

Figure 1: (a) Segmentation with the specification of a bounding box and areas in the foreground and background (source) (b) this project's goal, directed object segmentation

Recognizing the boundaries of objects within an image, referred to as object segmentation or foreground extraction, is a classic Computer Vision problem. Low effort tools for object segmentation would be highly useful for photo editing, yet programs such as Photoshop still require outlining or drawing over the surface of an object to segment it out instead of just providing the bounding box or even just clicking on the object. Though multiple algorithms have been developed for this task (see Fig 1), they typically require at least the object's bounding box as input and are still challenged by overlapping objects, a lack of a clear bounding box, and complicated objects.

Algorithms incorporating Machine Learning have become more common in the past decade and have been shown to be robust to these challenging cases. Most recently, Deep Learning approaches - or approaches with neural nets that have many layers and varied layer types - have set state of the art benchmarks on the problem. DeepMask is one of the newest and most successful of such approaches, and due it being open source is a particularly good model to iterate on for the task of directed object segmentation. It is designed to work directly from image inputs, and the modified 'DeepCrop' model just accepts an image and a pixel coordinate in order to enable one-click 'directed' object segmentation.

2. Related Work

Many non machine-learning algorithms, such as Grabcut [9], have been developed for the tasks of object segmentation and object detection (finding the bounding boxes of images within an image). In the last several years, Deep Learning approaches have achieved state of the art results on these problem. This began with the "R-CNN" model by Girshick et al [3] , which significantly outperformed prior approaches on the task of object detection. R-CNN works in two steps: first a set of object location proposals is generated by a pre-existing algorithm, and then a deep 'ConvNet' classifier is trained to identify the best proposals. Later state-of-the-art models kept the basic two-step structure but iterated

on the implementation : OverFeat made bounding box proposal be part of the trained model[10], and multiple groups developed efficiency and architecture improvements that made the basic R-CNN approach scale and perform better [4][11][2].

DeepMask is similar to R-CNN in that it is based on generating object location proposals with corresponding scores for those proposals[7]. The key difference is that it addresses the problem of object segmentation (producing objects' outlines) rather than object detection. The same group later published SharpMask[8], which produces refined object segmentations. More recently, an different method called ISFCN [1] has achieved performance surpassing that of DeepMask/SharpMask. There is no research I am aware of on one-click directed object segmentation, and as DeepMask was open source and established when this project began the proposed approach to directed object segmentation is based directly off of it.

3. DeepMask Overview

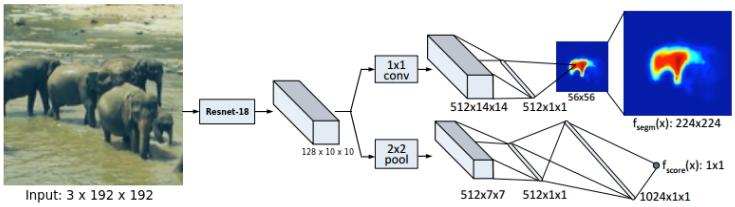

Figure 2: A diagram of DeepMask, from the original paper on it[7]

At a high level DeepMask simply inputs an image to a set of feature-extraction layers pre-trained on ImageNet (in particular, the same feature extraction architecture as in the 'Resnet' object classification network[5]), and then passes on the computed feature to a 'mask' branch and a 'score' branch that are both made up of fully connected linear layers (Figure 2). The two branches are jointly trained with image 'patch' inputs in which a single object is centered and fully contained. For full-image inference, the trained model is run in a convolution across the full image at multiple scales, and the outputs from the score branch are used to select the best segmentation masks.

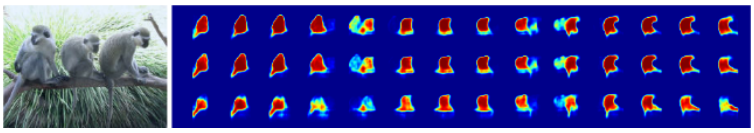

Figure 3: Visualization of DeepMask full image inference via Convolution, again from [7]

4. DeepCrop Overview

The first approach, 'DeepCropV1', has three key differences from DeepMask:

- Several inputs that represent the 'pixel click' location are added to the mask branch. These added inputs include three measures of the distance of each pixel from the 'click' pixel: the euclidean distance of the pixel coordinate, the euclidean distance of the pixel RGB values, and the absolute difference of the pixel luminance value.
- The object is no longer constrained to be in the center of the image. So, the full training image is used as input, and there is no need for convolution during full-

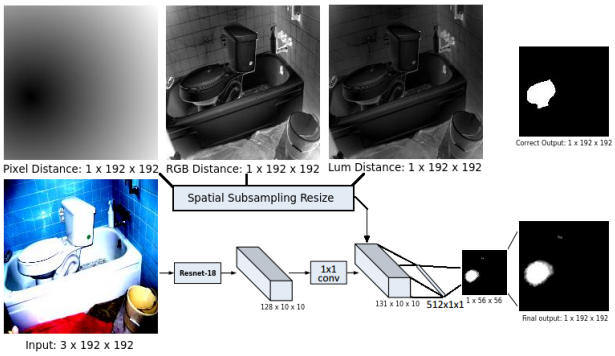

Figure 4: The 'DeepCropV1' model, with inputs and outputs from training. The score branch is removed, and inputs for the 'click' pixel are added to the mask branch.

image inference. The reasoning for this is that the added inputs should aid the model sufficiently to make it redundant.

- The score branch is removed entirely, since there is no constraint on the object being centered and no need for convolution during full-image inference.

There are several problems with DeepCropV1: the input image is significantly downsized which limits accuracy, segmenting a non-centered is harder to train for, and the feature extraction net is not provided the click location. A second version of DeepCrop was developed to take this into account. This revised model is the same as DeepMask, but the inputs it is trained with include a fourth image plane that represents distance from the clicked pixel. This 'DeepCropV2' is a minimal modification of DeepMask with the added input and works significantly better than DeepCropV1, and also slightly outperforms DeepMask.

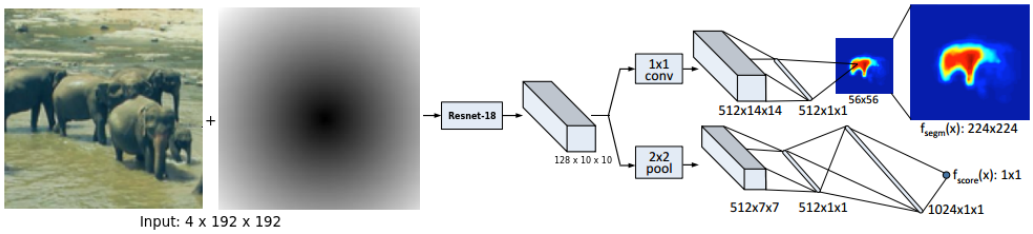

Figure 5: Diagram of the DeepCropV2

The code for DeepMask, implemented in Torch7, is open source and was directly modified for this project. The DeepCrop code can be viewed here. As with DeepMask, the MS COCO dataset is used for training and testing ([6]). The dataset contains a total of 123,287 with 886,284 outlined objects, though only 80 kinds of objects (see Figure 5). Optimization is done using stochastic gradient descent, as a sum of soft margin criterion loss from the mask branch and binary 1/0 loss from the score branch. The code is run on a personal computer as well as three Amazon AWS g2.2xlarge EC2 instances[1].

1. At a cost of $1000, paid for by a grant from the Cloud Credits for Research program.

5. Results

Figure 6: Graphs from training. An epoch is 4000 batches, and a batch has 16 examples. The line for 'DeepCropV2_50' is incomplete due to insufficient time to train it fully.

Results for training DeepMask, DeepCropV1 and DeepCropV2 are shown in Figure 8. Additionally, a version of DeepCropV2 with a larger feature extraction network (Resnet-50 instead of Resnet-18) and weights initialized to that of a pre-trained DeepMask model is included as 'DeepCropV2_50'. The loss decreases consistently for all models, though notably DeepMask outperforms DeepCropV1, and DeepCropV2 outperforms both. This makes sense, due the cited issues with DeepCropV1 and since the additional input to DeepMaskV2 makes it easier to learn a model quickly. Predictably, the partially pre-trained and larger DeepCropV2_50 model begins with lower loss but also does not improve significantly.

In contrast, the mean and median intersection-over-union (IoU)[2] evaluated on the test set during training does not diverge significantly between DeepMask and DeepCropV2. This implies the quality of the segmentation proposals by the two models is similar. But, since the losses differ significantly it is implied that DeepCropV2 should be superior at scoring the appropriate patch highly during full image inference. As with loss, the larger and pretrained DeepCropV2_50 model performs best on the IoU metrics.

Figure 7: Inputs and outputs from 'DeepCropV2_50', saved during training.

Quantitative summary metrics of the final trained models are listed in Table 1, and are largely consistent with the evaluation done during training. DeepCropV2 is slightly better than DeepMask on all the metrics, though the larger DeepCropV2_50 is about the same as DeepMask_50. I suspect that this is due to not having trained DeepCropV2_50 sufficiently, but on the whole the results reinforce the idea DeepCropV2 performs about the same in terms of segmentation accuracy as DeepMask.

Unfortunately, an implementation of full-image inference evaluation was not finished in time to get quantitative metrics for that. However, an interactive demo of DeepCropV2 (reachable at http://54.183.190.239:8000/) was implemented and used to evaluate the model

2. This is the intersection of the output with the correct output, divided by the union of the two.

Metric / Model	Train			Test		
	Mean IoU	Median IoU	IoU@0.7	Mean IoU	Median IoU	IoU@0.7
DeepMask	61.3%	69.0%	48.5%	60.4%	68.16%	47.0%
DeepMask_50	**68.0**%	**75.1**%	**59.9**%	65.8%	73.1%	**55.4**%
DeepCropV1	15.5%	0.0%	8.0%	11.6%	0.0%	4.1%
DeepCropV2	63.8%	70.0%	50.1%	63.5%	69.6%	49.3%
DeepCropV2_50	67.5%	74.6%	58.2%	**66.1**%	**73.2**%	55.3%

Table 1: Comparison of models **evaluated on image patches**. IoU@0.7 is IoU recall with 70% threshold. All values were evaluated with 5000 sampled examples. DeepMask_50 is a Facebook supplied pre-trained model, from which DeepCropV2_50's weights were initialized.

Figure 8: Top 5 segmentation proposals by DeepCropV2_50, though full-image inference.

qualitatively. It is clear from interacting with the demo that the model did learn to segment objects where the input pixel coordinate is located, but that the segmentations are generally of very poor quality. I suspect that this is due to the pixel distance being normalized during training and evaluation, which results in good results for image patches but does not work during convolutional full-image inference.

6. Conclusion

The project presented several challenges: understanding DeepMask, learning to program in Torch, and the sheer amount of compute power these models require. Overall I consider the project successful, since DeepCropV1 provided interesting results as to the limitations of DeepMask without the convolution operation and DeepCropV2 was shown to most likely be a viable approach for directed object segmentation. There are three major areas for future work: refining DeepCropV2 training to work better for whole images, integrating DeepCrop with SharpMask[8], and further developing a GUI to test the usefulness of DeepCrop.

References

[1] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. *CoRR*, abs/1603.08678, 2016.

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

[3] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729, 2014.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[6] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[7] Pedro O. Pinheiro, Ronan Collobert, and Piotr Dollr. Learning to segment object candidates. In *NIPS*, 2015.

[8] Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollr. Learning to refine object segments. In *ECCV*, 2016.

[9] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004.

[10] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[11] Christian Szegedy, Scott E. Reed, Dumitru Erhan, and Dragomir Anguelov. Scalable, high-quality object detection. *CoRR*, abs/1412.1441, 2014.