The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Tap and Shoot Segmentation

Ding-Jie Chen, Jui-Ting Chien, Hwann-Tzong Chen, Long-Wen Chang
National Tsing Hua University, Taiwan
{djchen.tw, ydnaandy123}@gmail.com , {htchen, lchang}@cs.nthu.edu.tw

Abstract

We present a new segmentation method that leverages latent photographic information available at the moment of taking pictures. Photography on a portable device is often done by tapping to focus before shooting the picture. This *tap-and-shoot* interaction for photography not only specifies the region of interest but also yields useful focus/defocus cues for image segmentation. However, most of the previous interactive segmentation methods address the problem of image segmentation in a post-processing scenario without considering the action of taking pictures. We propose a learning-based approach to this new tap-and-shoot scenario of interactive segmentation. The experimental results on various datasets show that, by training a deep convolutional network to integrate the selection and focus/defocus cues, our method can achieve higher segmentation accuracy in comparison with existing interactive segmentation methods.

Introduction

Interactive image segmentation incorporates a *human-in-the-loop* mechanism to collect hints from the user as a guide toward the expected segmentation result. Existing interactive segmentation algorithms adopt different types of user inputs such as bounding boxes, scribbles, control points, or simple clicks, which aim to specify the target object or the region of interest. Most of the existing algorithms operate as an interactive tool in a post-processing scenario, that is, the to-be-segmented image is assumed fixed and the user's inputs are not necessarily relevant to how the image is taken. We propose to consider the action of taking pictures as part of the interactive segmentation process. In such a scenario, the user may tap the touchscreen of the smartphone/camera to focus on the target object. The *tap-and-shoot* interaction provides useful latent information about the scene, and our approach can learn to extract the selection and defocus cues for segmenting the target object.

A camera with a lens is often preferable to a pinhole camera in that the exposure time can be greatly reduced when taking a picture. However, a lens can focus at only one distance in a scene, and the other spots that are nearer or farther from that distance will be out of focus and look blurry in the captured image. This kind of blur is called the *defocus blur*. The degree of defocus blur is correlated with the distance from the capture point to the camera, the position of the focal plane, the lens focal length, and the lens aperture size. The process of focusing the lens is to make the target object locate on the focal plane so that it can look sharper. In this work we build a convolutional network that learns to extracts the inherent defocus cue in the image captured via tap-and-shoot interaction.

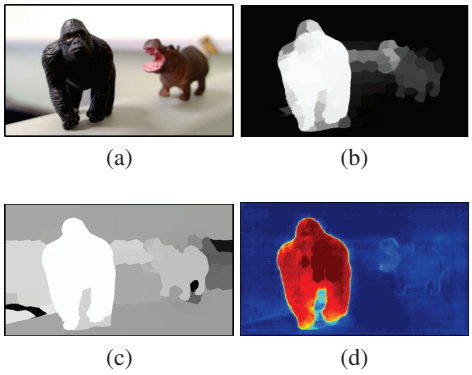


Figure 1: An example of segmentation with the selection and defocus cues. Notice that, the defocus map in (c) is just for visualization. Our method in fact implicitly learns the defocus cue via a convolutional network. (a) A tap-and-shoot image focusing on the gorilla. (b) The selection cue of (a), in which the brighter intensity values represent the higher preferences to the target object according to the user's tapping position. (c) The defocus cue of (a), in which the brighter intensity values represent the lower degrees of estimated defocus blur. (d) The predicted probability map of the target object. The final segmentation result can be easily derived from the probability map via thresholding.

Recently, semantic segmentation becomes a popular research topic owing to the greatly improved segmentation accuracy via deep network learning. A semantic segmentation task aims to label the regions of some predefined object classes. On the other hand, in a general figure-ground segmentation problem the region of interest of a given image is ambiguous to define since different users may prefer different regions of interest. We address the figure-ground

segmentation problem using an easy-to-train convolutional network architecture. The selection and defocus cues are extracted from the tap-and-shoot image. The selection cue is estimated based on the user's tapping position and is represented as an augmented image channel to guide the deep convolutional network toward localizing the region of interest without training with all possible object classes. The network is also designed to learn to extract the defocus cue from the tap-and-shoot images. The augmented channel and the extracted defocus cue are integrated to predict the region of interest. Fig. 1a shows a tap-and-shoot image for segmentation. Figs. 1b-1d illustrate the augmented channel of the selection cue, the defocus map for visualization, and the prediction result produced by the proposed convolutional network.

The main ideas of this work are summarized as follows.

- We address a new scenario of image segmentation involving tap-and-shoot interaction. The task is challenging because only one input/click from the user is available to specify the target.
- We propose an end-to-end trained convolutional network that is effective in extracting the full-scale features for predicting the region of interest.
- We propose to use the selection and defocus cues to guide the prediction of the convolutional network for the tap-and-shoot segmentation task. Our method improves the segmentation accuracy on several datasets in comparison with previous interactive segmentation methods.

Related Work

We briefly review previous work on defocus estimation and image segmentation in semantic and interactive manners.

Defocus Blur Estimation. Methods for defocus blur estimation on monocular images can be categorized into gradient based methods and frequency based methods. Gradient based methods (Elder and Zucker 1998; Zhuo and Sim 2011; Peng, Zhao, and Cosman 2015; Chen, Chen, and Chang 2016) draw upon the observation that the gradient magnitudes at edge locations often decay rapidly after blurring. Frequency based methods (Zhang and Hirakawa 2013; Zhu et al. 2013; Shi, Xu, and Jia 2015) are based on the fact that blurring would suppress high-frequency components and enhance low-frequency components. We find that the existing defocus estimation methods are not robust enough to extract the defocus cue for guiding the segmentation task. Instead, we train our convolutional network to extract the inherent defocus cue in the image by itself for predicting the target region.

Deep Network Architectures for Semantic Segmentation. Although our work is aimed at solving figure-ground segmentation with latent cues, we study the state-of-the-art deep network architectures for semantic segmentation, and adopt useful ideas to design our network for tap-and-shoot segmentation. Long et al. proposes an efficient end-to-end network architecture called Fully Convolutional Networks (FCN), which replaces all fully-connected layers with convolutions (Long, Shelhamer, and Darrell 2015). FCN is widely used in many applications that involve dense structured predictions. U-net extends FCN by adding more shortcut connections and a larger feature channels in the upsampling phase (Ronneberger, Fischer, and Brox 2015). The architecture becomes more symmetric and yields better segmentation results than FCN. Yu and Koltun further replace the pooling layer with the dilated convolution, which increases the receptive filed without decreasing the image resolution. The network uses the same numbers of parameters but gets more precise segmentation (Yu and Koltun 2016). DeepLab adopts the idea of dilated convolution and proposes a new pooling strategy called *atrous spatial pyramid pooling* (ASPP) (Chen et al. 2016; 2017). Recently, DenseNet shows a more efficient architecture by densely connecting different layers (Huang et al. 2017). Its performance is better than other previous architectures on classification tasks of CIFAR-10, CIFAR-100, SVHN, and ImageNet. Jégou et al. extend DenseNets to semantic segmentation and obtain state-of-the-art results on CamVid dataset (Jégou et al. 2017). PSPNet is also a state-of-the-art architecture for semantic segmentation. It has ranked at the first place in ImageNet Scene Parsing Challenge 2016, PASCAL VOC 2012 benchmark, and Cityscapes benchmark (Zhao et al. 2017).

Interactive Image Segmentation. Interactive image segmentation allows the user to specify annotations in forms of scribbles/seeds (Boykov and Jolly 2001; Grady 2006; Gulshan et al. 2010; Wang, Han, and Collomosse 2014; Feng et al. 2016; Xu et al. 2016), control points (Kass, Witkin, and Terzopoulos 1988; Mortensen and Barrett 1995; Badoual et al. 2017), or bounding boxes (Rother, Kolmogorov, and Blake 2004; Wu et al. 2014; Cheng et al. 2015b) to guide the segmentation toward the region of interest. These algorithms usually employ *graph cuts*, *random walks*, *level sets*, or *geodesic distance* to segment the images according to the annotations provided by the user. In general, it is easier for users to indicate the target object via a bounding box, but the segmentation accuracy is also constrained by how precisely the box is drawn. The seed based algorithms and the control-points based algorithms can both tackle the situations of complex-shaped objects as long as sufficient user inputs are given—more rounds of interactions are needed than the bounding-box based algorithms.

Figure-Ground Segmentation with User Inputs We propose a deep convolutional network for image segmentation by leveraging the selection and defocus cues available at the moment of taking pictures. Our approach differs from the existing ones in several aspects: First, the task we aim to solve is a new style of one-seed interactive image segmentation involving a tap-and-shoot imaging mechanism. The user only needs to interact with the touchscreen to focus the target object by tapping, and our model will try to predict the region of interest. Second, in comparison with the similar work (Xu et al. 2016), our deep convolutional network

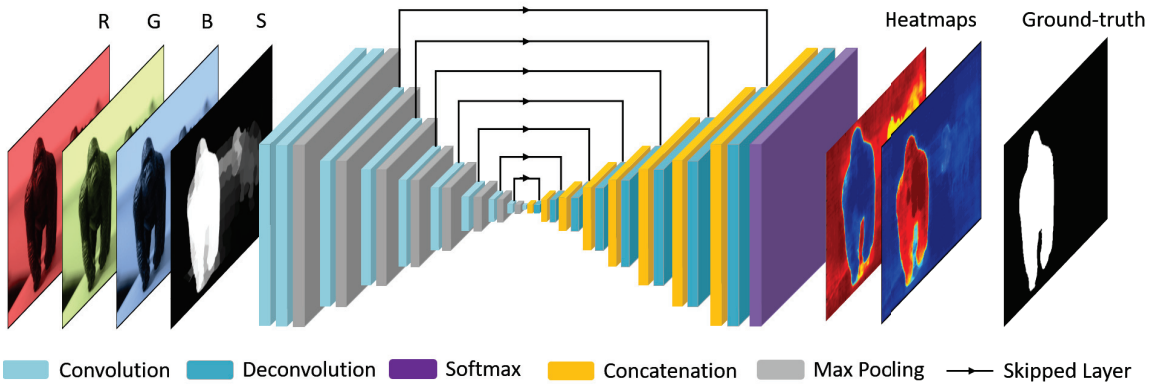


Figure 2: An overview of the proposed deep convolutional network. Given an input image with an augmented channel S representing the selection cue, the network can predict two heatmaps with respect to the target region and the background.

is end-to-end trained with significantly less training time. Our predicted heatmaps have clearer object boundaries and hence have no need to use the graph cut optimization as (Xu et al. 2016). Last but not the least, our model integrates the selection and defocus cues for predicting the region of interest. The defocus cue implicitly learned by our deep convolutional network is shown to be more beneficial to the segmentation accuracy than handcrafted defocus estimation.

Approach

When the user taps the touchscreen of smartphone/camera to focus on the target object and takes a picture, the *tapping* action provides a selection cue, which indicates the region of interest. The captured image also contains an inherent defocus cue, which separates the scene into an in-focus plane and an out-of-focus background. We propose a deep convolutional network that leverages the selection and defocus cues derived from the tap-and-shoot interaction to solve the figure-ground segmentation task. The selection cue is transformed into an intensity map S as one additional input channel. The image augmented with the additional selection channel $RGBS$ is then used to train the proposed convolutional network. The defocus cue could also be estimated as an intensity map D. However, D is not used in our approach, but will be evaluated in the experiment section for comparison. The trained network predicts two heatmaps with respect to the regions of interest (foreground) and non-interest (background). Fig. 2 illustrates the architecture of our convolutional network.

To train our deep network, we first prepare data for the tap-and-shoot scenario. We scale all input images in proportion and center-crop into size 256×256.

Training Samples. For each image provided with ground-truth segmentation, we generate nine training images as follows: we apply three different blurring levels to the background region, and for each blurred image we randomly pick three different tapping positions on the foreground region. The training samples are constructed in this way to simulate the tap-and-shoot scenario, in which the selected region is in-focus and the background is out-of-focus.

Testing Samples. For each testing image, we directly augment the image with the additional selection channel S and then feed the augmented image $RGBS$ into the trained network for prediction. The segmentation result can be easily obtained by thresholding the probability output of the network. Notice that, our model does not need to apply additional graph cut optimization as (Xu et al. 2016) to align the segmentation boundaries.

Network Architecture

We formulate our problem as a binary classification problem, in which each pixel either belongs to a foreground region or a background region. The network input can be a 3-channel RGB image or an augmented image with any combinations of additional feature channels, such as S, D or both S and D. In our final approach, we choose to use a 4-channel input $RGBS$. The output resolution is the same as the original input resolution. The output contains two channels (heatmaps) representing the probabilities of being foreground or background. We use cross entropy to measure the probability error. The complete loss function is defined as follows:

$$L = -\sum_{i=1}^{P}\sum_{c=1}^{2}\left[y_i^c \log \sigma(x_i^c) + (1 - y_i^c)\log(1 - \sigma(x_i^c))\right] , \tag{1}$$

where c could be 1 or 2 to indicate the foreground or the background. P is the number of total pixels, x_i^c is the predicted value, y_i^c is the ground truth label, and $\sigma(\cdot)$ is the softmax function. We have evaluated different models and the comparisons will be discussed in the discussion section. Our final model, illustrated in Fig. 2, is similar to U-net (Ronneberger, Fischer, and Brox 2015) but much simpler. Except the input and output layer, we use exactly one convolutional layer followed by a max-pooling layer in the down-sampling phase and exactly one transpose convolutional layer followed by a concatenated layer in the up-sampling phase.

Through the experiments we show that with this much simplified architecture, our model achieves competitive performance but requires significantly less space and training time.

Implementation Details

Channel Representation. To account for the computational cost with the augmented image channels, we represent each image as a superpixel-level graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$. The vertex set $\mathcal{V}$ is the set of superpixels and the edge set $\mathcal{E}$ contains all links between every two adjacent superpixels. The weighting function $\omega : \mathcal{E} \rightarrow \mathbb{R}_0^+$ is defined as

$$\omega_{ij} = e^{-\theta_1 \|f_i - f_j\|_2}, \tag{2}$$

where f_i and f_j denote the CIE-Lab mean color features of the adjacent superpixels v_i and v_j. The set $\mathcal{V}$ is constructed via SLIC algorithm (Achanta et al. 2012) with roughly 800 superpixels. We keep $\theta_1 = 10$ fixed for all experiments.

Constructing the Selection Channel. Given a graph $\mathcal{G}$ with a specified vertex $v_j \in \mathcal{V}$, the shortest path $\Phi(v_i|v_j)$ from vertex v_i to the specified vertex v_j is defined as the accumulated edge weights along the path. The shortest path function Φ can be defined as

$$\Phi(v_i|v_j) = \min_{v'_1=v_i,\ldots,v'_m=v_j} \sum_{k=1}^{m-1} \omega(v'_k, v'_{k+1}), \forall (v'_k, v'_{k+1}) \in \mathcal{E}, \tag{3}$$

where m denotes the path length. We use the function Φ to estimate the label similarity between the unlabeled superpixels and the labeled superpixel.

When the user taps the touchscreen of the smartphone/camera to focus on the target object, the tapping location yields a reliable annotation for indicating the region of interest. In contrast, since the superpixels locating on the image border are rarely to be the region of interest, assuming these superpixels belong to the background is reasonable (Wei et al. 2012; Wang, Han, and Collomosse 2014). Therefore, we respectively label the two kinds of superpixels as the region of interest (F) and the background (B). Thus, we can obtain two label similarity values for each unlabeled superpixel.

To obtain the selection channel, we further use the sigmoid function ρ to convert the difference between the two label similarities (F and B) of each superpixel v to the range $[0, 1]$:

$$\rho(|v^F - v^B|) = \frac{1}{1 + e^{-\theta_2(|v^F - v^B|)}}, \tag{4}$$

where v^F denotes the label similarity of v to the region of interest, v^B denotes the label similarity of v to the background, and θ_2 is defined as 0.33 in our experiments. Fig. 1b shows the augmented channel of the selection cue.

Network Parameters. During the down-sampling phase, we apply eight (2×2)-stride max-pooling layers on the input of RGBS image with size $256 \times 256 \times 4$ and extract the most high-level $1 \times 1 \times 512$ features at the bottleneck. In the up-sampling phase we reversely convert the high-level features into dense predictions with eight (2×2)-stride transpose convolution (deconvolution) layers. The complete setting can be found in Table 1. All the activation functions are ReLU except the output layer, in which we use softmax. We use batch normalization after every convolution/deconvolution layers to accelerate the converging speed. We also apply dropout layers on the layers Deconv8 to Deconv5 for avoiding over-fitting. The dropout probability is 0.2 during training and 0.0 during testing. In our experiments, we also have tried to normalize the intensity values of the input image into the range of $[-1.0, 1.0]$, but the training process becomes unstable. We find that simply subtracting the intensity values of the input image by 127.5 could makes the training process easier to train.

Table 1: This table summarizes the parameters of our network architecture. The input size is 256×256 and has four channels $RGBS$.

Layer name		Numbers of feature map	output size
Conv1_input		32	256×256
Conv1	+ pooling	64	128×128
Conv2	+ pooling	128	64×64
Conv3	+ pooling	256	32×32
Conv4	+ pooling	512	16×16
Conv5	+ pooling	512	8×8
Conv6	+ pooling	512	4×4
Conv7	+ pooling	512	2×2
Conv8	+ pooling	512	1×1
Conv_bottleneck		512	1×1
Deconv8	+ concatenate	512	2×2
Deconv7	+ concatenate	512	4×4
Deconv6	+ concatenate	512	8×8
Deconv5	+ concatenate	512	16×16
Deconv4	+ concatenate	512	32×32
Deconv3	+ concatenate	128	64×64
Deconv2	+ concatenate	64	128×128
Deconv1	+ concatenate	32	256×256
Conv_output		2	256×256

Experiments

We first describe the evaluation methodology, the algorithms in comparison, and the datasets that are evaluated in our experiments. Then we show the comparison results of the existing interactive image segmentation algorithms and our method.

Methodology. To evaluate the segmentation accuracy, we use the intersection-over-union (IoU) metric, which is defined as $\frac{|R \bigcap G|}{|R \bigcup G|}$, where R denotes the segmentation result and G denotes the ground-truth foreground object. All algorithms are run on the same environment (Intel i7-4770 3.40 GHz CPU, 8GB RAM, NVIDIA Titan X GPU).

In Comparison with Other Algorithms. Our approach is compared with six well-known interactive image segmentation algorithms, which are box based algorithms: GrabCut

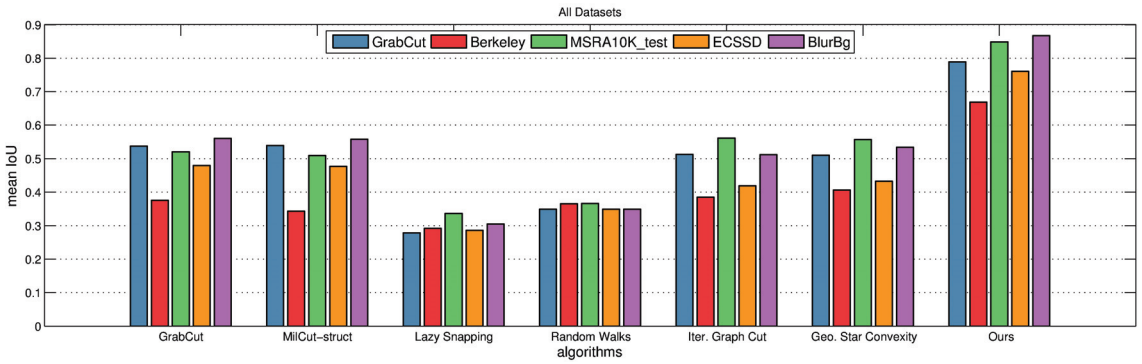


Figure 3: The mean IoU score of each algorithm on five datasets.

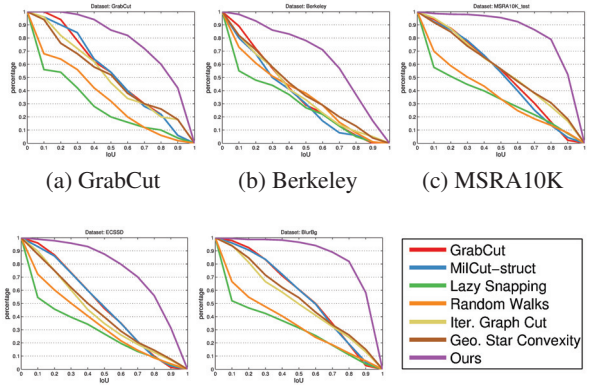
(Rother, Kolmogorov, and Blake 2004), MilCut-struct (Wu et al. 2014); and seed/scribble based algorithms: Lazy Snapping (Li et al. 2004), Random Walks (Grady 2006), Interactive Graph Cuts (Boykov and Jolly 2001), Geodesic Star Convexity (Gulshan et al. 2010)[1].

Datasets. We evaluate all algorithms on four public datasets. Each image contains one foreground region with pixel-level ground-truth labeling. **GrabCut dataset** (Rother, Kolmogorov, and Blake 2004): It contains 50 natural images. **Berkeley dataset** (McGuinness and O'Connor 2010): It contains 100 images. The images are from the popular Berkeley dataset (Martin et al. 2001). **Extended complex scene saliency dataset (ECSSD)** (Shi et al. 2016): The dataset contains 1,000 natural images. **MSRA10K dataset** (Cheng et al. 2015a): This dataset contains 10,000 natural images. We use MSRA10K dataset for training and testing our network. We partition the dataset into three non-overlapping subsets with the numbers of 8,000, 1,000, and 1,000 for training, validation, and testing. **BlurBg dataset**: We select 500 *blur-perceivable* images from the above four datasets for evaluation.

Quantitative Results

We use three kinds of figures to compare the segmentation quality of each segmentation algorithm. Fig. 3 shows the average IoU score of each segmentation algorithm. It obviously demonstrates that our approach has higher segmentation quality than all other algorithms on each dataset. As we have expected, our approach has the best performance while dealing with the blur-perceivable images such as the images in the dataset 'BlurBg.' Notice that, Fig. 3 shows that the blur-perceivable images are not trivial for other segmentation algorithms.

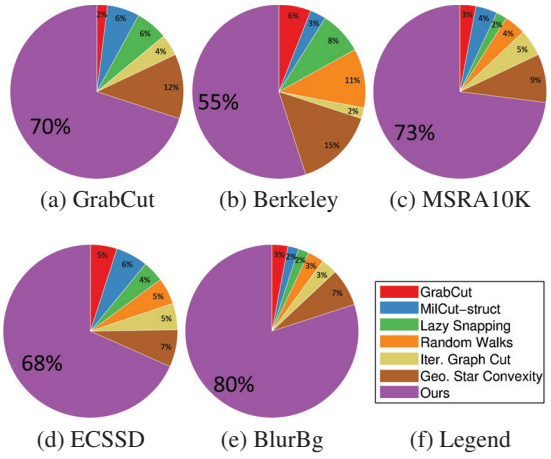
[1]The programs of MilCut-struct are provided by the authors. The programs of GrabCut and Lazy Snapping are implemented by Gupta and Ramnath http://www.cs.cmu.edu/~mohitg/segmentation.htm. The code of Random Walks is from http://cns.bu.edu/~lgrady/software.html The programs of Interactive Graph Cuts and Geodesic Star Convexity are from http://www.robots.ox.ac.uk/~vgg/research/iseg/.



(a) GrabCut (b) Berkeley (c) MSRA10K

(d) ECSSD (e) BlurBg (f) Legend

Figure 4: The ratio of each algorithm above the specific segmentation qualities in IoU metric.



(a) GrabCut (b) Berkeley (c) MSRA10K

(d) ECSSD (e) BlurBg (f) Legend

Figure 5: The ratio of each algorithm gets the top segmentation accuracy.

Fig. 4 shows the ratio of each algorithm above the specific segmentation qualities in IoU metric on each dataset.

Table 2: Three architectures. The first column shows the size of network weights. The second column show the elapsed time after 16K updates. The last five columns show the mean IoU scores on different datasets. We only update 16K times here for comparisons while we update 260K times in our final setting, and the differences in performance get smaller at convergence.

	Space (MB)	Time (hrs)	Grabcut	Berkeley	MSRA10K	ECSSD	BlurBg
FCN	652.45	4	**0.7228**	**0.5564**	**0.8111**	**0.7025**	**0.8248**
FCN-extended	801.71	6	0.7159	0.5548	0.8021	0.6847	0.8227
AE (Ours)	**98.99**	**1.5**	0.6909	0.5203	0.7689	0.6537	0.7992

The larger 'area under the curve' means the better segmentation quality. This figure also demonstrates that our approach surpasses all other algorithms in each demand IoU quality.

Fig. 5 shows the ratio of each algorithm that gets the top segmentation accuracy on each dataset. The higher ratio that an algorithm gets in a given dataset means the algorithm is more suitable for being used in that dataset. This figure shows that our approach performs at least 40% better than other algorithms on the five datasets.

For comparison, our method takes about 0.28 seconds per image with the aid of GPU. In comparison with the similar work (Xu et al. 2016), our approach has roughly 15% and 2% IoU increments[2] on the GrabCut dataset and the Berkeley dataset, respectively.

Qualitative Results

Fig. 6 shows the segmentation results. The results in Fig. 6 demonstrate that our segmentation method achieves better performance than other algorithms.

Discussions

Network Comparison

We analyze three different deep networks in the aspect of used memory, training speed, and performance as shown in Table 2. All models are optimized using ADAM algorithm with same learning rate 0.0001. The batch size is 9 and the network is running on TitanX. We implement all of them in Tensorflow and the Table 2 shows the results after 16k updates.

The first model (FCN) is a direct adoption of FCN but we initialize the weights with VGG19 pretrained on ImageNet. Regarding the additional input channel, we initialize it with zero values as (Xu et al. 2016). The second one (FCN-extend) is based on FCN but we extend the up-sampling phase by adding additional deconvolution layers and train it from scratch. The skip layer also increases along with the deconvolution, and the fuse layers are replaced with concatenation. The third one (AE), which is used in our approach, is a simple auto-encoder as described in the previous section.

As shown in Table 2, our model uses fewer weights and only requires one-sixth space compared with FCN, while the FCN-extend uses the most parameters since it contains additional layers. Regarding the training time, The table shows that our model is three times faster than FCN and outperforms others. However, the prediction scores are almost the same no matter what model is chosen. FCN gets higher scores but it is initialized with pretrained model weights. One main possible reason is that for the domain-specific task like our scenario, a complicated architecture and redundant parameters are not necessary. The results show that, in the tap-and-shoot segmentation task, our simple model is good enough and achieves comparable results with significant reduction of time and space. For reference, (Xu et al. 2016) takes three days to fine-tune FCN-32s and five days to fine-tune FCN-16s and FCN-8s. Our network only takes less than two days for training from scratch but gets better performance on GrabCut and Berkeley datasets.

[2]Since the code of the related work (Xu et al. 2016) is not yet released, we report the values according to Fig. 4. of their paper.

Defocus Blur Learning

While training the convolutional network, we observe that directly augment the input with a defocus channel D, which is estimated by some off-the-shelf defocus estimation algorithms, is actually harmful to the network performance (see Fig. 7). This is because the current off-the-shelf defocus estimation methods are not able to provide reliable estimations. The noisy defocus estimations would misguide the network training. One possible solution is generating the blur-background images, as is done in our training samples, to enforce the network to learn the defocus blur by itself. It seems that preparing training samples with multiple blur-background images and then augmenting each image with the additional selection channel can make the network achieve better segmentation quality. Fig. 8 compares the example segmentation results using different types of training samples.

In Fig. 8, the network trained with RGB images without blurring the image backgrounds is hard to separate those objects that have similar features but locate at different depths (see the top two rows of Fig. 8). In contrast, the network trained with RGBD images without blurring the image backgrounds may fail to separate different objects locating at the same depth (see the bottom two rows of Fig. 8).

Conclusion

We propose a highly efficient convolutional network that learns to solve the figure-ground segmentation task under the scenario of tap-and-shoot interaction. The trained network is able to leverage the selection and defocus cues that are available at the moment of taking pictures. Our method achieves high segmentation accuracy on various public datasets, especially for those images that contain in-focus foreground objects and out-of-focus backgrounds.

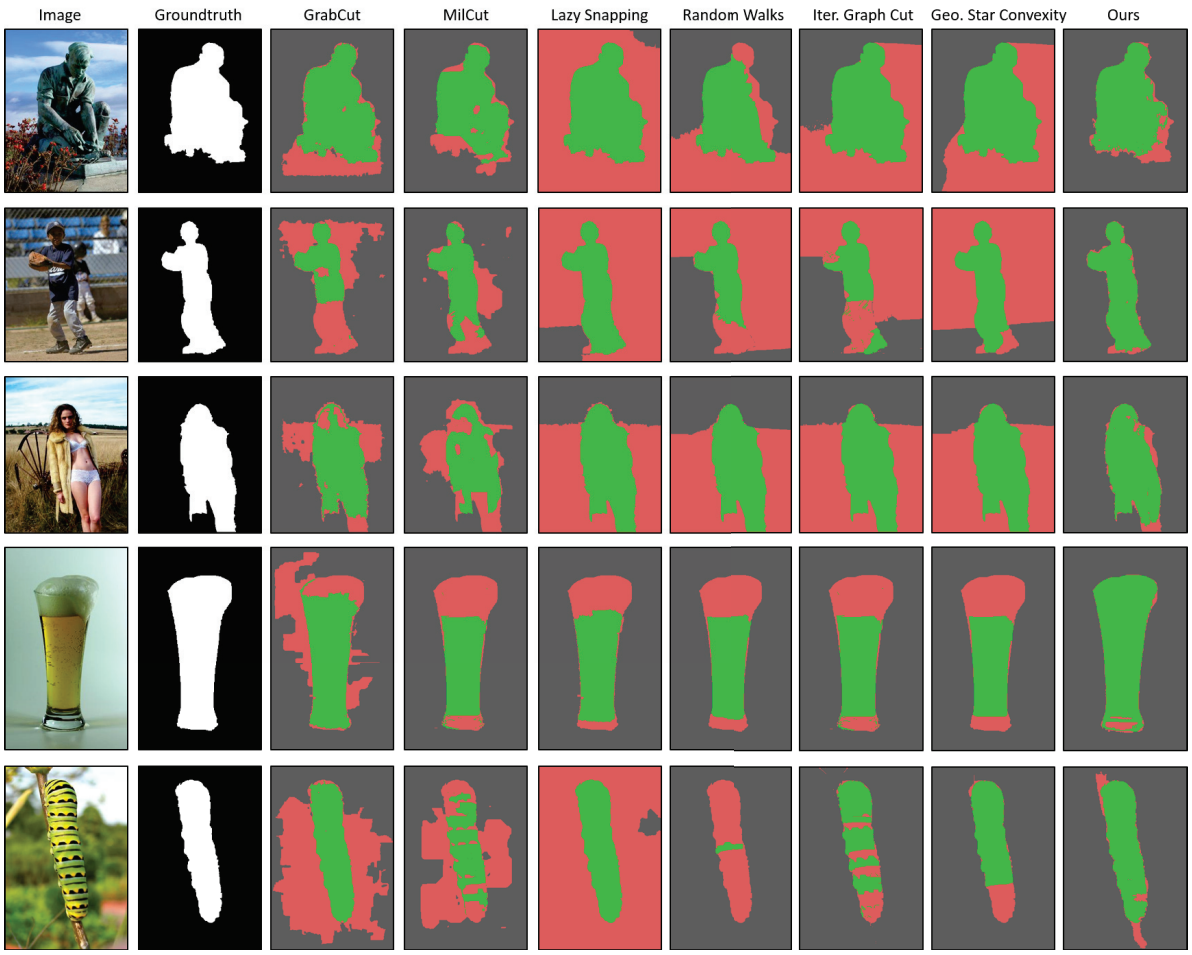


Figure 6: The segmentation results of all algorithms. The green and red regions respectively indicate the correct and the wrong segmentations. GrabCut and MilCut are guided by the box covering 81% central area of each image. Lazy Snapping, Random Walks, Iterative Graph Cut, and Geodesic Star Convexity are guided by the foreground seed, locating on the centroid of the ground-truth foreground region, and the background seed, selected randomly from the background area.

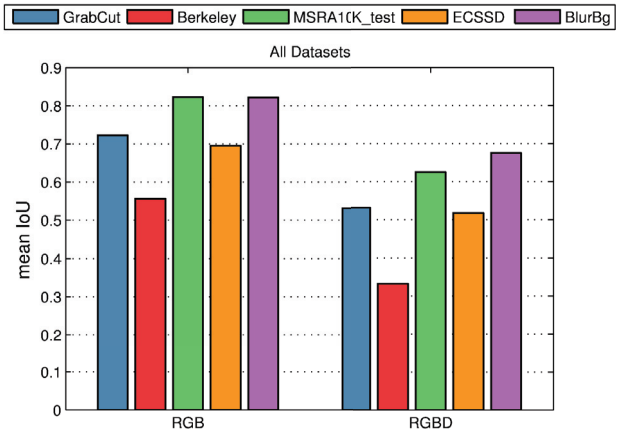


Figure 7: The mean IoU scores of our network trained with two kinds of samples without blurring the image backgrounds. Left set: using RGB training samples. Right set: using RGBD training samples.

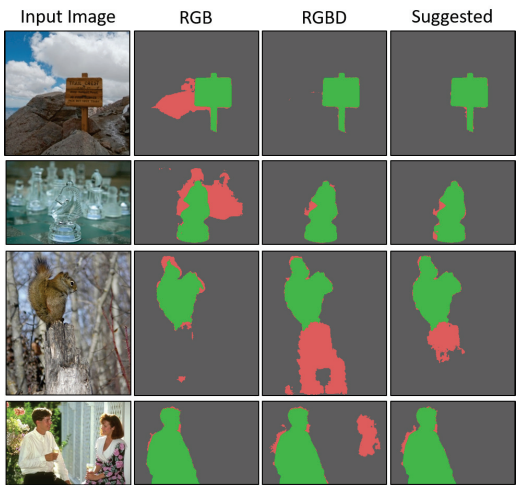


Figure 8: The green and red regions respectively indicate the correct and the wrong segmentations.

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(11):2274–2282.

Badoual, A.; Schmitter, D.; Uhlmann, V.; and Unser, M. 2017. Multiresolution subdivision snakes. *IEEE Trans. Image Processing* 26(3):1188–1201.

Boykov, Y., and Jolly, M. 2001. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*.

Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR* abs/1606.00915.

Chen, L.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *CoRR* abs/1706.05587.

Chen, D.; Chen, H.; and Chang, L. 2016. Fast defocus map estimation. In *ICIP*.

Cheng, M.; Mitra, N. J.; Huang, X.; Torr, P. H. S.; and Hu, S. 2015a. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 37(3):569–582.

Cheng, M.; Prisacariu, V. A.; Zheng, S.; Torr, P. H. S.; and Rother, C. 2015b. Densecut: Densely connected crfs for realtime grabcut. *Comput. Graph. Forum* 34(7):193–201.

Elder, J. H., and Zucker, S. W. 1998. Local scale control for edge detection and blur estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(7):699–716.

Feng, J.; Price, B.; Cohen, S.; and Chang, S. 2016. Interactive segmentation on rgbd images via cue selection. In *CVPR*.

Grady, L. 2006. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(11):1768–1783.

Gulshan, V.; Rother, C.; Criminisi, A.; Blake, A.; and Zisserman, A. 2010. Geodesic star convexity for interactive image segmentation. In *CVPR*.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR*.

Jégou, S.; Drozdzal, M.; Vázquez, D.; Romero, A.; and Bengio, Y. 2017. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *CVPR*.

Kass, M.; Witkin, A. P.; and Terzopoulos, D. 1988. Snakes: Active contour models. *International Journal of Computer Vision* 1(4):321–331.

Li, Y.; Sun, J.; Tang, C.; and Shum, H. 2004. Lazy snapping. *ACM Trans. Graph.* 23(3):303–308.

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

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

McGuinness, K., and O'Connor, N. E. 2010. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition* 43(2):434–444.

Mortensen, E. N., and Barrett, W. A. 1995. Intelligent scissors for image composition. In *SIGGRAPH*.

Peng, Y.; Zhao, X.; and Cosman, P. C. 2015. Single underwater image enhancement using depth estimation based on blurriness. In *ICIP*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23(3):309–314.

Shi, J.; Yan, Q.; Xu, L.; and Jia, J. 2016. Hierarchical image saliency detection on extended CSSD. *IEEE Trans. Pattern Anal. Mach. Intell.* 38(4):717–729.

Shi, J.; Xu, L.; and Jia, J. 2015. Just noticeable defocus blur detection and estimation. In *CVPR*.

Wang, T.; Han, B.; and Collomosse, J. P. 2014. Touchcut: Fast image and video segmentation using single-touch interaction. *Computer Vision and Image Understanding* 120:14–30.

Wei, Y.; Wen, F.; Zhu, W.; and Sun, J. 2012. Geodesic saliency using background priors. In *ECCV*.

Wu, J.; Zhao, Y.; Zhu, J.; Luo, S.; and Tu, Z. 2014. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *CVPR*.

Xu, N.; Price, B. L.; Cohen, S.; Yang, J.; and Huang, T. S. 2016. Deep interactive object selection. In *CVPR*.

Yu, F., and Koltun, V. 2016. Multi-scale context aggregation by dilated convolutions. In *ICLR*.

Zhang, Y., and Hirakawa, K. 2013. Blur processing using double discrete wavelet transform. In *CVPR*.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *CVPR*.

Zhu, X.; Cohen, S.; Schiller, S.; and Milanfar, P. 2013. Estimating spatially varying defocus blur from A single image. *IEEE Trans. Image Processing* 22(12):4879–4891.

Zhuo, S., and Sim, T. 2011. Defocus map estimation from a single image. *Pattern Recognition* 44(9):1852–1858.