Interactive Image Segmentation with Latent Diversity

Zhuwen Li Qifeng Chen Vladlen Koltun
Intel Labs

Abstract

Interactive image segmentation is characterized by multimodality. When the user clicks on a door, do they intend to select the door or the whole house? We present an end-to-end learning approach to interactive image segmentation that tackles this ambiguity. Our architecture couples two convolutional networks. The first is trained to synthesize a diverse set of plausible segmentations that conform to the user's input. The second is trained to select among these. By selecting a single solution, our approach retains compatibility with existing interactive segmentation interfaces. By synthesizing multiple diverse solutions before selecting one, the architecture is given the representational power to explore the multimodal solution space. We show that the proposed approach outperforms existing methods for interactive image segmentation, including prior work that applied convolutional networks to this problem, while being much faster.

1. Introduction

Interactive image segmentation is an essential building block of advanced image editing applications. Research on interactive segmentation dates back decades, with early work focusing on boundary tracing techniques [18, 32]. Modern approaches aim to classify image regions as foreground or background, treating user input as ground-truth labels. The classification view rose to prominence with the graph cut formulation of Boykov and Jolly [4]. Subsequent work introduced iterative fitting of foreground and background distributions [37], alternative distance metrics for label propagation [12, 2, 9, 13, 35], and more powerful random field models for pixel-level labeling [22, 8].

From a machine learning perspective, interactive image segmentation can be viewed as a few-shot active learning problem. From this standpoint, the system uses a classifier with some form of prior knowledge of objects and their appearance in images. The classifier is then given ground-truth labels ('foreground', 'background') for a very small number of pixels in a new image. This information must be used to classify all the other pixels in the image. This perspective highlights two interrelated challenges: (1) how to acquire, represent, and apply prior knowledge about object appearance, and (2) how to minimize the amount of ground-truth data (in the form of foreground/background clicks) that the user must provide at test time.

Until recently, prior knowledge about object appearance was encoded primarily in hand-crafted features and distance metrics [4, 37, 12, 2, 9, 13, 35, 5, 24, 8]. At present, convolutional networks are the starting point for representing prior knowledge about object appearance [26, 25, 30]. Indeed, recent work has shown that convolutional networks can be applied to interactive segmentation and do yield significant gains over prior approaches [41]. Our work builds on these findings and goes further by introducing different architectures and loss functions. Our driving motivation is to make progress on the second challenge: minimizing the number of labeled samples (clicks) that the user must provide to achieve a desired level of accuracy.

We focus on a structural issue that becomes apparent when the number of samples provided by the user is small. The elusive "holy grail" of interactive image segmentation is one-shot selection: the user simply points to an object or a collection of objects by clicking a single pixel, and the system segments the desired region. Considering this scenario clarifies that the problem is ill-posed and characterized by multimodality. When the user clicks on a person's jacket, is the intention to select the jacket, the whole person, or the group that this person is part of? This multimodality is at the center of our work.

In this paper, we present an architecture that tackles the multimodality problem head-on. The first ingredient is a single fully-convolutional network that is trained to synthesize a diverse set of solutions. This network takes a representation of the image and the user's input and produces a set of possible segmentations. The training loss encourages diversity in the synthesized solutions, with the goal that each plausible segmentation is represented by at least one of the proposals. This gives the network the representational power it needs to deal with the multimodality of the solution space. However, this is not sufficient because the system must still produce a single segmentation in order to be compatible with existing image editing interfaces.

This is addressed by our second ingredient: a network that is trained to select one of the synthesized segmentations. Together, the two networks explore the multimodal solution space and then select the most promising mode. Crucially, at test time this is performed in a single forward pass through the compound architecture. This can be done in a fraction of a second on standard hardware, supporting the use of the presented approach in interactive image editing systems.

Our work builds on a long line of ideas on multiple choice learning and diversity in probabilistic models [43, 3, 14, 42, 15, 10, 20, 27, 21]. Most of these formulations were developed in the context of graphical models and are aimed at producing a diverse set of solutions. The recent work of Lee et al. [27] applies these ideas to deep networks, but considers an ensemble of networks and modifies the learning algorithm to selectively channel the gradient flow through individual networks in the ensemble. In contrast, our approach trains a single feed-forward stream that generates diverse solutions and then selects among them. Our formulation is agnostic to the learning algorithm and is compatible with modern gradient-based solvers without any modification. Training a single model rather than an ensemble reduces memory and computational requirements at test time, and producing a single solution in the end means that our approach can be used in existing image editing interfaces and is directly comparable to the most relevant prior work.

We evaluate the presented approach on multiple datasets and compare it to many prior methods for interactive image segmentation. The experiments demonstrate that the presented approach outperforms the state of the art in all regimes. Our approach reduces the number of clicks that are required to reach a certain accuracy, and increases the accuracy that is reached in a given number of clicks. The presented approach is also the fastest. In particular, the state-of-the-art approach of Xu et al. [41], while based on convolutional networks, also relies heavily on postprocessing using graph cuts. In contrast, our approach does not require postprocessing and amounts to a single forward pass through a compound deep network; as demonstrated by the experiments, this approach is both faster and more accurate. We show that the trained model generalizes across datasets and perform a user study that tests the model's performance in real interactive segmentation tasks.

2. Overview

Consider a color image $X \in \mathbb{R}^{w \times h \times 3}$. The interactive image segmentation model allows the user to segment the desired region in the image by successively placing positive ('foreground') and negative ('background') clicks. Every time a click is placed, the segmentation is updated. Once the segmentation matches the region intended by the user, the process concludes.

At every step, the system must segment the image given a fixed number of positive and negative clicks: $\mathcal{S}_p, \mathcal{S}_n \subset \{0,1\}^{w \times h}$. The output is a mask that assigns a binary label to each pixel. We relax the output representation to be continuous: $Y \in [0,1]^{w \times h}$. A hard segmentation is obtained by thresholding each pixel at ½.

Our goal is to train a model that will admit the image and the clicks as input and will produce the segmentation as output: $Y = F(\mathbf{X})$, where $\mathbf{X}$ is the input representation. A key difficulty is that the problem is in general ill-defined and there are often multiple plausible outputs, especially when the cardinality of $\mathcal{S}_p$ and $\mathcal{S}_n$ is small. In particular, this multimodality poses a difficulty for learning the model F: fitting a function approximator is hard if the underlying mapping is not a function. Incorporating explicit provisions for addressing the multimodal nature of the output can thus increase the accuracy of the model, even if a single segmentation is produced at the end. We therefore introduce additional structure into our model.

First, a network f synthesizes M segmentations: $f(\mathbf{X}) = \mathcal{Y}$, where $\mathcal{Y} \in \mathbb{R}^{w \times h \times M}$. (The effect of the hyperparameter M on performance will be studied in Section 5.) That is, $\mathcal{Y}$ is a collection of segmentation masks. Second, a network g selects one of the M solutions synthesized by f. Specifically, g takes a representation of the input and the synthesized segmentation masks, and produces a probability distribution over $\{1, \ldots, M\}$. Thus $g(\mathbf{Z}) \in \mathbb{R}^M$, where $\mathbf{Z}$ is the representation provided to g. Then $F(\mathbf{X}) = f_{\arg\max g(\mathbf{Z})}(\mathbf{X})$. This is illustrated in Figure 1.

This additional structure allows the model to explicitly represent and reason about diverse solutions. A key observation is that this is useful even when a single solution must be produced in the end. If the network can only represent one solution, it will be pulled towards multiple modes during training, with a result that may "split the difference". The intermediate representation $\mathcal{Y}$ allows the network f to produce multiple clean segmentations. The effect of this intermediate representation is demonstrated in Figures 1 and 2. The "diverse segmentations" shown in Figure 1 are intermediate solutions generated by the network f for the representation $\mathcal{Y}$; the network g then selects among these intermediate solutions. Figure 2 illustrates the effect of f and $\mathcal{Y}$ in the setting of one input click: the network f produces plausible segments for the network g to select from. The network g must still choose one of the intermediate solutions in $\mathcal{Y}$, but these intermediate solutions are higher-quality because f could fully commit to each mode.

3. Segmentation

Input representation. The input to the segmentation network f consists of the image X, clicks $\mathcal{S}_p$ and $\mathcal{S}_n$, distance

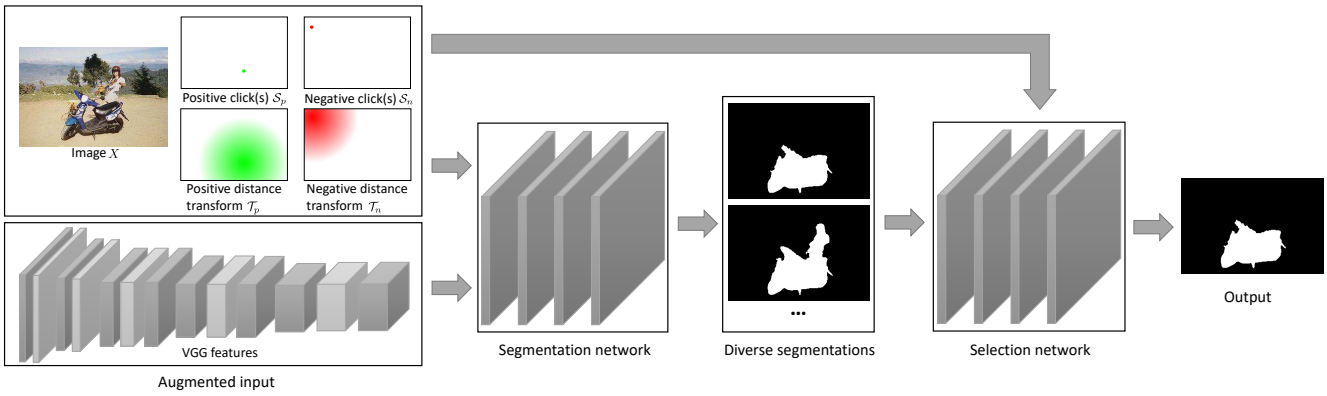


Figure 1. Overview of the presented approach to interactive image segmentation. The input image and the user's input, represented via distance transforms, are augmented by feature activations from a pretrained visual perception network. This augmented input representation is given to a network that synthesizes diverse segmentations that conform to the user's input. A selection network chooses one of the predicted segmentations as the output.

transforms defined by the clicks, and per-pixel VGG features. Our use of distance transforms follows the work of Xu et al. [41]. The distance transforms $\mathcal{T}_p$ and $\mathcal{T}_n$ are single-channel intensity maps, defined as follows:

$$\begin{aligned} \mathcal{T}_p(\mathbf{p}) &= \min_{\mathbf{q}\in\mathcal{S}_p} \|\mathbf{p}-\mathbf{q}\|_2 \\ \mathcal{T}_n(\mathbf{p}) &= \min_{\mathbf{q}\in\mathcal{S}_n} \|\mathbf{p}-\mathbf{q}\|_2. \end{aligned} \tag{1}$$

The distance transforms are truncated at 255 for efficient representation. In addition, we apply a VGG-19 network [39] pretrained on the ImageNet dataset [38] to the image X and extract the feature maps from the following layers: 'conv1_2', 'conv2_2', 'conv3_2', 'conv4_2', and 'conv5_2'. The feature maps are bilinearly upsampled to the resolution of X and are appended to the input tensor. They can be viewed as per-pixel "hypercolumn" features that are used to augment the input at full resolution [17]. The total number of channels in the input is 1,477.

Network architecture. We now present the architecture of the segmentation network f. Since the number of channels in the input tensor is very high (1,477), our first step is to reduce the dimensionality of the data. This is accomplished by a learned affine projection to a lower-dimensional feature space. Concretely, the first layer in f is an affine projection layer (1×1 convolution) that maps the feature column of each pixel to $\mathbb{R}^{64}$, yielding 64 feature maps at resolution $w\times h$.

We then use a context aggregation network (CAN) [44, 7]. The network operates at full resolution and applies 3×3 convolutions with progressively higher dilation, each followed by a leaky ReLU [31]. Note that the augmented input representation enables us to use a dedicated dense prediction network that is trained from scratch for the task at hand, without being constrained by an architecture that could have been pretrained for image classification. The network architecture is summarized in Table 1. The output layer has M channels, one for each synthesized segmentation mask. The final nonlinearity is a sigmoid that maps each pixel to the range $[0,1]$. The next section describes the training loss for synthesizing diverse high-quality segmentations, and the network g that selects among them.

4. Diversity

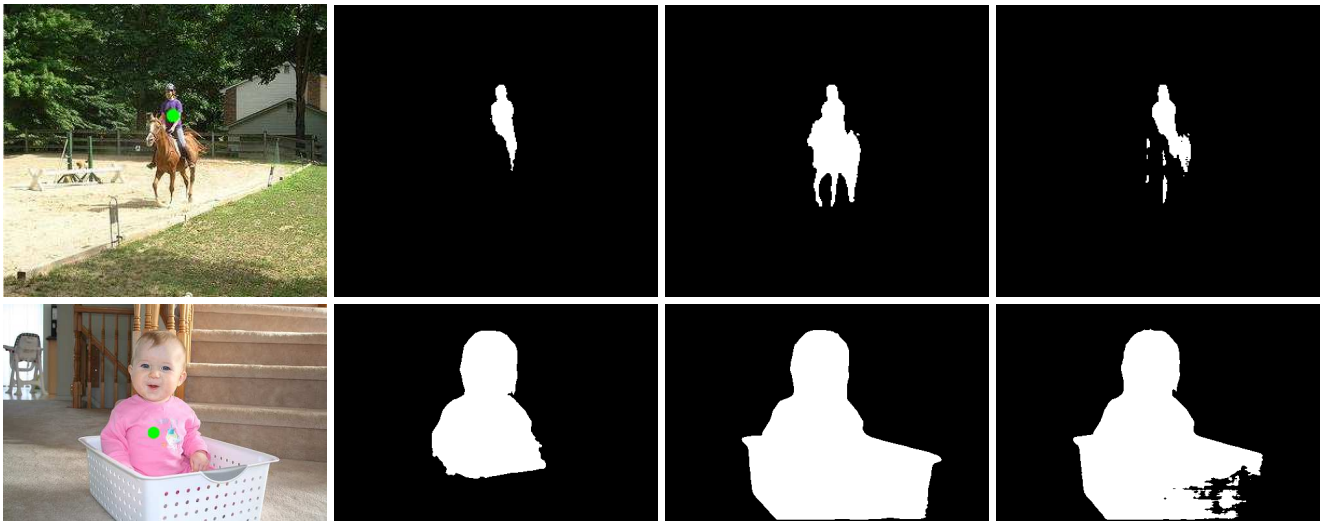
Let $\mathcal{D} = \{(\mathbf{X}_i, Y_i)\}$ be a training set, where $\mathbf{X}_i$ is the augmented input representation and Y_i is the intended segmentation mask for example i. Given the input representation $\mathbf{X}_i$, the network f generates M segmentation masks: $f(\mathbf{X}_i;\boldsymbol{\theta}_f) = \langle f_1(\mathbf{X}_i;\boldsymbol{\theta}_f), \ldots, f_M(\mathbf{X}_i;\boldsymbol{\theta}_f)\rangle$. Here $\boldsymbol{\theta}_f$ is the network's parameter vector. For training f to generate diverse high-quality segmentations, our starting point is the hindsight loss [14, 6]:

$$\sum_i \min_m \ell(Y_i, f_m(\mathbf{X}_i;\boldsymbol{\theta}_f)), \tag{2}$$

where $\ell(A,B)$ is a loss function that measures the distance between the ground-truth segmentation mask $A\in\{0,1\}^{w\times h}$ and the predicted mask $B\in[0,1]^{w\times h}$. There are many possible loss functions for ℓ, including regression (e.g., L_p) and classification (e.g., cross-entropy). We choose instead to use a task loss. Specifically, since segmentation accuracy is commonly evaluated using the Jaccard (IoU) distance, we use a relaxation of this distance [40, 23, 33, 1]:

$$\ell(A,B) = 1 - \frac{\sum_{\mathbf{p}} \min(A(\mathbf{p}), B(\mathbf{p}))}{\sum_{\mathbf{p}} \max(A(\mathbf{p}), B(\mathbf{p}))}, \tag{3}$$

where $A(\mathbf{p})$ and $B(\mathbf{p})$ are the values of the respective masks at pixel $\mathbf{p}$. In addition, the input clicks are used as



(a) Input (b) Segmentation 1 (c) Segmentation 2 (d) Segmentation w/o diversity

Figure 2. Illustration of diversity given one positive input click. (a) shows the input image with a positive click (green). (b) and (c) show two of the intermediate segmentations in $\mathcal{Y}$. (d) shows the segmentation that would have been produced by the same network f without diversity ($M = 1$).

soft constraints:

$$\begin{aligned} \ell_c(\mathcal{S}_p, \mathcal{S}_n, B) &= \|\mathcal{S}_p \odot (\mathcal{S}_p - B)\|_1 \\ &+ \|\mathcal{S}_n \odot (\mathcal{S}_n - (1 - B))\|_1, \end{aligned} \quad (4)$$

where $\odot$ denotes the Hadamard elementwise product. Combining the two losses, we define

$$\begin{aligned} \mathcal{L}_f(\boldsymbol{\theta}_f) &= \sum_i \min_m \big\{\ell(Y_i, f_m(\mathbf{X}_i; \boldsymbol{\theta}_f)) \\ &+ \ell_c(\mathcal{S}_p^i, \mathcal{S}_n^i, f_m(\mathbf{X}_i; \boldsymbol{\theta}_f))\big\}. \end{aligned} \quad (5)$$

The loss $\mathcal{L}_f$ allows the different output channels to specialize and assume responsibility for different modes in the solution space. The remaining problem is to select one of them.

Selection network. The selection network g takes as input the image X, the clicks $\mathcal{S}_p$ and $\mathcal{S}_n$, the distance transforms $\mathcal{T}_p$ and $\mathcal{T}_n$, and the output tensor of the segmentation network, $f(\mathbf{X}_i; \boldsymbol{\theta}_f)$. (At training time, we randomly shuffle the M masks in $f(\mathbf{X}_i; \boldsymbol{\theta}_f)$ before they are handed to g, so that the network is forced to analyze the content.) The goal is to select one of the M segmentation masks for presentation to the user. To this end, the selection network is trained with the cross-entropy loss:

$$\mathcal{L}_g(\boldsymbol{\theta}_g) = \sum_i \left(-g_{\phi_i}(\mathbf{Z}_i; \boldsymbol{\theta}_g) + \log \sum_{m=1}^{M} \exp\left(g_m(\mathbf{Z}_i; \boldsymbol{\theta}_g)\right) \right), \quad (6)$$

where $\mathbf{Z}_i$ is the input to the selection network, ϕ_i is the index of the mask that minimizes the Jaccard distance to Y_i, and $\boldsymbol{\theta}_g$ are the parameters.

Since g is essentially a classification network, we could use an image classification structure [39]. This did not yield good results in our experiments. After testing different architectures we discovered, surprisingly, that a dense prediction structure followed by average pooling performs much better. Specifically, for g we use a similar full-resolution structure to f, as summarized in Table 1. The differences are that layer 1 (dimensionality reduction) is not needed and the output is not a $w \times h \times M$ tensor but an M-vector. This vector is produced by adding a global average pooling layer that pools over each full-resolution $w \times h$ activation map in the final tensor. Remarkably, this full-resolution prediction followed by global average pooling [28, 46, 45] significantly outperforms the baselines in our experiments.

Ranked diversity loss. We have also experimented with a simple alternative approach that allows selecting a single solution without using a selection network. This approach will serve as one of the baselines in our experiments. For this simple approach, we add a term to the loss $\mathcal{L}_f$ that imposes an ordering on the synthesized solutions and encourages the network f itself to rank them. Specifically, we modify the loss function (5) as follows:

$$\begin{aligned} \mathcal{L}_f^{\text{RDL}}(\boldsymbol{\theta}_f) &= \sum_i \min_m \big\{\ell(Y_i, f_m(\mathbf{X}_i; \boldsymbol{\theta}_f)) \\ &+ \ell_c(\mathcal{S}_p^i, \mathcal{S}_n^i, f_m(\mathbf{X}_i; \boldsymbol{\theta}_f))\big\} \\ &+ \sum_i \sum_{m=1}^{M} \lambda_m \ell(Y_i, f_m(\mathbf{X}_i; \boldsymbol{\theta}_f)), \end{aligned} \quad (7)$$

where $\{\lambda_m\}$ is a decreasing sequence, such as

Layer	1	2	3	4	5	6	7	8	9	10	11
Convolution	1×1	3×3	3×3	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	1	2	4	8	16	32	64	128	1	1
Receptive field	1×1	3×3	7×7	15×15	31×31	63×63	127×127	255×255	511×511	513×513	513×513
Width	64	64	64	64	64	64	64	64	64	64	M

Table 1. The architecture of the segmentation network f.

$\lambda_m = 10^{-2} \cdot 2^{M-m}$. The additional term breaks the symmetry between the solutions and imposes an ordering on them.

5. Evaluation

5.1. Experimental setup

Datasets. For training, we use the Semantic Boundaries Dataset (SBD) [16]. This dataset provides high-quality boundaries and is not restricted to a particular domain. SBD uses images from the Pascal VOC challenge [11], but provides many more object masks; it is essentially an augmented version of Pascal VOC with the same images but more comprehensive annotations. Specifically, SBD provides binary object segmentation masks for all the objects in the training and validation sets of the Pascal VOC 2011 challenge. The dataset includes 8,498 training images and 2,820 test images. We use the training set for training, and test on the test set.

To evaluate cross-dataset generalization, we also evaluate the trained model on a number of other datasets:

- GrabCut [37]. This dataset contains 49 images and corresponding segmentation masks that delineate a salient foreground object.
- DAVIS [34]. This dataset was created to evaluate video segmentation algorithms, but individual frames from the video sequences can also be used to evaluate image segmentation. We chose this dataset due to its diversity and the high quality of the ground-truth segmentation masks. The dataset contains 50 HD video sequences with pixel-accurate segmentation masks. We sample 10% of the annotated frames at random, yielding 345 images that are used in the evaluation.
- Microsoft COCO [29]. We sample 800 object instances from the validation set. Specifically, we sample 10 images from each of the 80 categories, and choose a ground-truth object instance at random for each sampled image.

Note that we do not train on GrabCut, DAVIS, or COCO. Our model is trained only once, on the SBD training set. Testing this model on GrabCut, DAVIS, and COCO verifies that the model generalizes across datasets.

Baselines. We compare the presented approach to a number of well-known interactive image segmentation models: graph cuts [4], random walks [12], geodesic matting [2], Euclidean star convexity [13], and geodesic star convexity [13]. We also compare to the deep object section (DOS) approach [41], which is our main baseline. The complete DOS system uses graph cuts to refine the mask produced by a convolutional network. We therefore also evaluate the performance of the DOS network itself, without the graph cut post-processing. The DOS network is trained on the same data as our segmentation network, using the same training procedure.

Training. To generate the positive and negative clicks for training, we follow the simulation protocol of Xu et al. [41]. This generates a set of simulated clicks for each instance in the SBD training set. The network f is trained using Adam [19], with single-image minibatches and learning rate 10^{-4}. Training proceeds for 100 epochs. The network g is then trained using the same procedure.

Testing. At test time, clicks are generated on the fly during the evaluation, to simulate user interaction. Let $\mathcal{O}$ be the ground-truth object mask. The first click is positive and is sampled from the following probability distribution over $\mathcal{O}$:

$$P(\mathbf{p}; \mathcal{O}) = \frac{d(\mathbf{p}, \partial\mathcal{O})}{\sum_{\mathbf{q}\in\mathcal{O}} d(\mathbf{q}, \partial\mathcal{O})}, \tag{8}$$

where $d(\mathbf{p}, \partial\mathcal{O})$ is the geodesic distance of the pixel $\mathbf{p}$ to the boundary $\partial\mathcal{O}$ of the ground-truth mask. Each subsequent click is placed on a pixel that is still misclassified. (Recall that the predicted segmentation is updated after each click.) Let $\mathcal{O}'$ be the set of misclassified pixels. The next click is sampled from the distribution $P(\cdot; \mathcal{O}')$ over $\mathcal{O}'$. This process continues until the number of clicks reaches 20.

5.2. Results

Table 2 reports the average number of clicks required to reach 85% and 90% IoU on each dataset. Our approach outperforms all the baselines on all datasets, including the three datasets on which it was not trained (GrabCut, DAVIS, and COCO). For example, the number of clicks to reach 85% IoU required by DOS (with post-processing) is higher than the number of clicks required by our approach by 59% on the standard GrabCut dataset. Note that the performance

Method	DAVIS		GrabCut		SBD		COCO		mean		time (ms)
	85%	90%	85%	90%	85%	90%	85%	90%	85%	90%	
Random walk	16.71	18.31	11.36	13.77	12.22	15.04	13.62	16.74	13.48	15.96	437
Geodesic matting	18.59	19.50	13.32	14.57	15.36	17.60	16.91	18.63	16.04	17.57	919
Graph cut	15.13	17.41	7.98	10.00	13.60	15.96	15.23	17.61	12.98	15.24	1348
Euclidean star convexity	15.41	17.70	7.24	9.20	12.21	14.86	14.04	16.98	12.25	14.68	1314
Geodesic star convexity	15.35	17.52	7.10	9.12	12.69	15.31	14.39	16.89	12.38	14.63	1249
DOS w/o GC	12.52	17.11	8.02	12.59	14.30	16.79	13.99	16.88	12.21	15.84	112
DOS with GC	9.03	12.58	5.08	6.08	9.22	12.80	9.07	13.55	8.10	11.25	549
Our approach	**5.95**	**9.57**	**3.20**	**4.79**	**7.41**	**10.78**	**7.86**	**12.45**	**6.11**	**9.39**	236

Table 2. Average number of clicks required to reach a certain IoU on each of the four datasets. Lower is better. Our approach outperforms all prior methods. The last column shows the running time of different methods on VGA-resolution images.

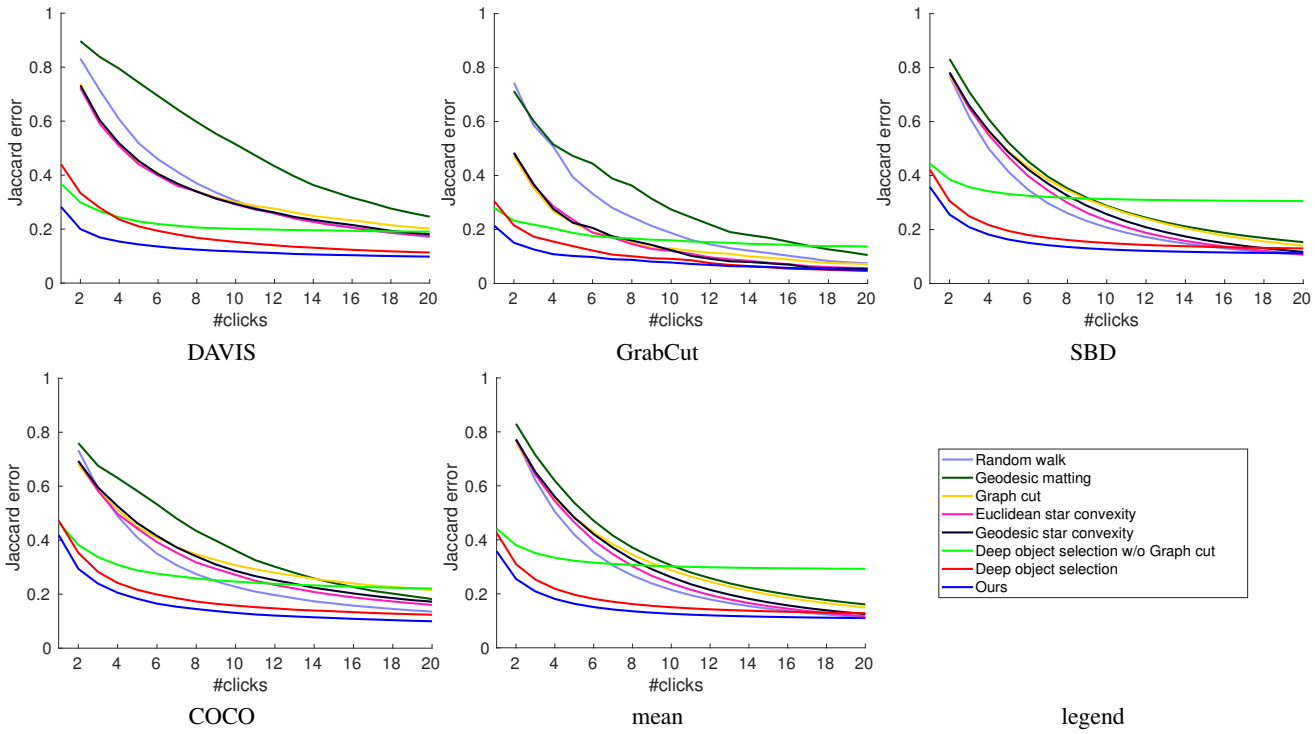


Figure 3. Jaccard error versus the number of clicks on the DAVIS, GrabCut, SBD, and COCO datasets.

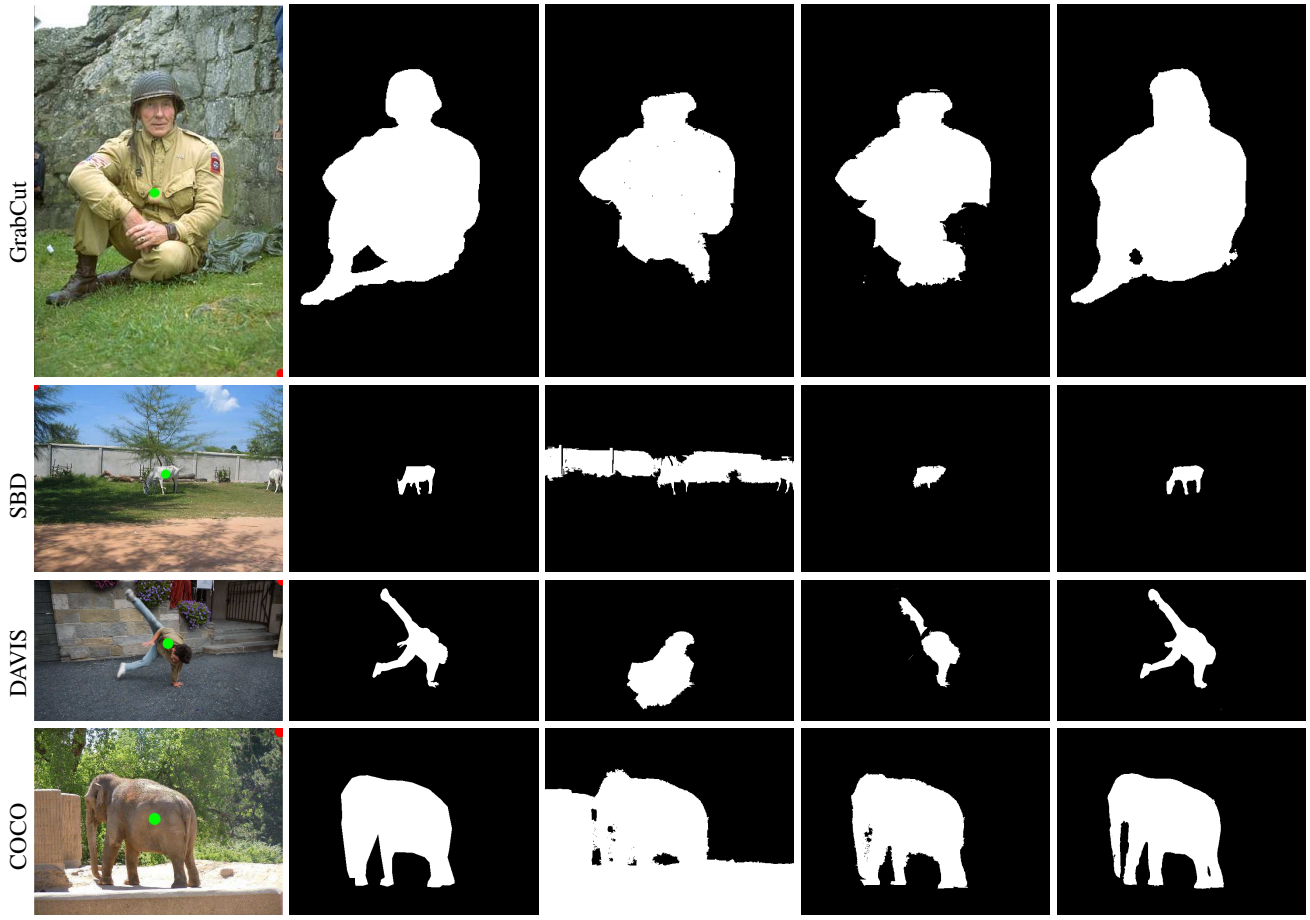
of the DOS network without post-processing is much worse, requiring 151% more clicks on GrabCut and 100% more clicks across the four datasets to reach 85% IoU.

Table 2 also reports the running time of different methods. Running time was measured on a workstation with an i7-5960X 3.0GHz CPU and a Titan X GPU. Our approach is 2.3 times faster than DOS, primarily due to the time-consuming postprocessing employed in that pipeline.

Figure 3 plots the Jaccard error versus the number of clicks on each dataset. Our approach again outperforms all the baselines. For example, after a single click on the DAVIS dataset, the Jaccard error of the segmentation produced by DOS is higher than the error of our segmentation by 87%. (Note that older approaches require at least two clicks – one positive and one negative – to produce a result.) The error of the DOS segmentation after two clicks is higher than our error by 81%.

Qualitative results are provided in Figure 4.

Controlled experiments. Table 3 reports the results of a controlled analysis of each component in the presented approach. For each condition in this experiment, we take our full model and remove or replace a single idea to evaluate its contribution to the total performance. First, we disable



(a) Input (b) Ground truth (c) Graph cuts (d) DOS (e) Our approach

Figure 4. (a) One image from each dataset, with one positive and one negative click. (b) Ground-truth segmentation mask. (c-e) Segmentation results generated for this input by graph cuts [4], deep object selection [41], and our approach.

the augmented input representation (VGG feature maps as input) described in Section 3 ("Without augmented input" condition). Next, we replace the task loss (IoU) by two alternative losses (cross-entropy or L_1). Next, we remove network g, reduce the number of output channels in f to 1, and train f with the task loss to produce a single segmentation mask that is used as the model's output ("Without diversity" condition). Next, we evaluate the performance of the pipeline without the distance transforms $\mathcal{T}_p$ and $\mathcal{T}_n$ in the input representations ("Without distance transforms" condition), without the clicks $\mathcal{S}_p$ and $\mathcal{S}_n$ in the input representation ("Without click input" condition), and without the soft constraints (4) in the loss ("Without soft constraints" condition). Finally, we replace the network g by the ranked diversity loss (RDL, (7)). The results demonstrate that all the ideas presented in the paper contribute to the model's performance.

Model	#clicks
Without augmented input	9.41
Without IoU loss (replaced by cross-entropy)	7.18
Without IoU loss (replaced by L_1)	6.93
Without diversity	7.26
Without distance transforms	9.11
Without click input	9.67
Without soft constraints	6.32
Selection network g replaced by RDL	6.13
Full model	5.95

Table 3. Ablation study that evaluates the contribution of different ideas to the performance of the presented model. The table reports the average number of clicks required to achieve 85% IoU on the DAVIS dataset.

We now analyze the effect of the number M of intermediate solutions. Figure 5 plots the average number of clicks required to achieve 85% or 90% IoU on the DAVIS and GrabCut datasets for different settings of M between 1

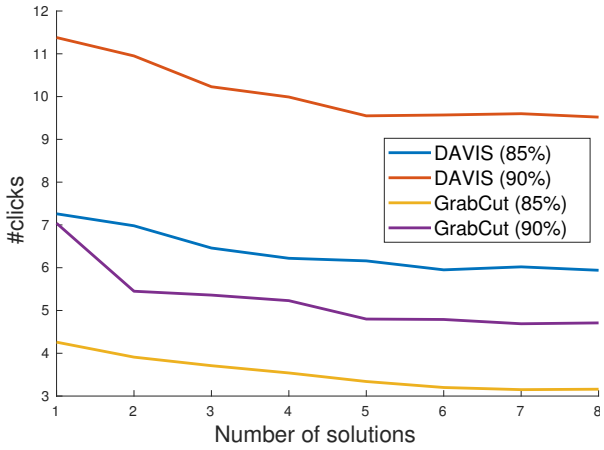


Figure 5. Effect of the hyperparameter M. Each curve shows the average number of clicks (vertical axis) required to achieve 85% or 90% IoU on the DAVIS and GrabCut datasets for different settings of M (horizontal axis).

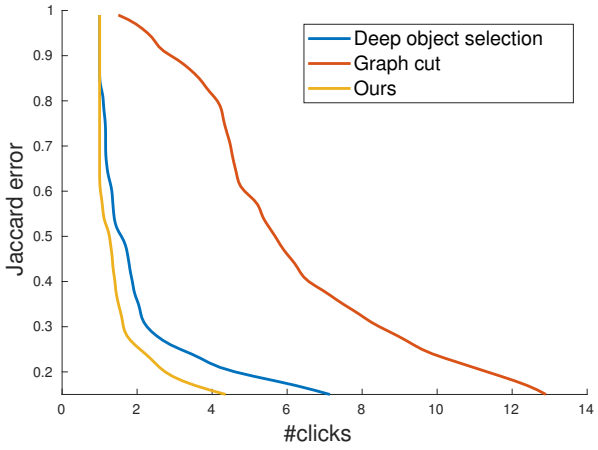


Figure 6. User study. Participants performed interactive segmentation tasks until 20 clicks or 85% IoU were reached. This figure shows the average numbers of clicks (horizontal) required to reach a certain Jaccard error (vertical). Left is better.

and 8. The results indicate that increasing the number of intermediate solutions helps up to $M = 6$, at which point the performance plateaus. We therefore use $M = 6$ in all other experiments.

Next, we perform a controlled experiment that studies the effect of the specific ConvNet architecture we used. As described in Section 3, we used the context aggregation network (CAN) for dense prediction [44, 7]. We can use a different architecture for this purpose, such as the U-net [36]. Table 4 reports the performance of the presented approach when the segmentation network f uses the U-net architecture rather than the CAN. The results indicate that the CAN is somewhat more effective, but the difference is mild and the presented approach would outperform all baselines with the U-net as well.

	DAVIS		GrabCut	
Model	85%	90%	85%	90%
U-net	6.25	9.96	3.22	4.81
CAN	**5.95**	**9.57**	**3.20**	**4.79**

Table 4. The effect of specific architecture of the segmentation subnetwork f. Average number of clicks required to reach 85% and 90% IoU on the DAVIS and GrabCut datasets.

User study. We conducted a user study to evaluate the performance of the presented approach with real human input. Five paid participants were given a tutorial on the interface and then performed 30 interactive segmentation tasks each. In a single task, the participant is given an image and the desired segmentation mask, and is asked to obtain the desired mask by placing positive and negative clicks. The task continues until 20 clicks are placed or 85% IoU is reached. Each image/mask pair is given three times, with a different model driving the segmentation interface in each trial: graph cuts, DOS (with post-processing), and our approach. The order of the conditions is randomized every time. The participants were not told what the conditions are, merely that there are different models driving the interface. The image/mask pairs are randomly sampled from the DAVIS dataset. In total, 50 segmentation tasks were performed with each model, across participants and across randomized trials. The results are shown in Figure 6. The desired segmentation was obtained with the smallest number of clicks when the interface was driven by the presented model. The average number of clicks to reach 85% IoU was 4.36 with the presented approach versus 7.14 with DOS and 12.90 with graph cuts.

6. Conclusion

We have presented a new approach to interactive image segmentation. Our approach reduces user-guided segmentation to a forward pass in a convolutional network. The network is structured so as to represent multiple segmentations before selecting one for display. We have shown that the presented model outperforms all prior approaches to interactive segmentation, while being the fastest.

References

[1] F. Ahmed, D. Tarlow, and D. Batra. Optimizing expected intersection-over-union with candidate-constrained CRFs. In *ICCV*, 2015. 3

[2] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV*, 82(2), 2009. 1, 5

[3] D. Batra, P. Yadollahpour, A. Guzmán-Rivera, and G. Shakhnarovich. Diverse M-best solutions in Markov random fields. In *ECCV*, 2012. 2

[4] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, 2001. 1, 5, 7

[5] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7), 2012. 1

[6] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. 3

[7] Q. Chen, J. Xu, and V. Koltun. Fast image processing with fully-convolutional networks. In *ICCV*, 2017. 3, 8

[8] M. Cheng, V. A. Prisacariu, S. Zheng, P. H. S. Torr, and C. Rother. DenseCut: Densely connected CRFs for realtime GrabCut. *Computer Graphics Forum*, 34(7), 2015. 1

[9] A. Criminisi, T. Sharp, C. Rother, and P. Pérez. Geodesic image and video editing. *ACM Transactions on Graphics*, 29(5), 2010. 1

[10] D. Dey, V. Ramakrishna, M. Hebert, and J. A. Bagnell. Predicting multiple structured visual interpretations. In *ICCV*, 2015. 2

[11] 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), 2010. 5

[12] L. Grady. Random walks for image segmentation. *PAMI*, 28(11), 2006. 1, 5

[13] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, 2010. 1, 5

[14] A. Guzmán-Rivera, D. Batra, and P. Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *NIPS*, 2012. 2, 3

[15] A. Guzmán-Rivera, P. Kohli, D. Batra, and R. A. Rutenbar. Efficiently enforcing diversity in multi-output structured prediction. In *AISTATS*, 2014. 2

[16] B. Hariharan, P. Arbeláez, L. D. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 5

[17] B. Hariharan, P. Arbelaez, R. B. Girshick, and J. Malik. Object instance segmentation and fine-grained localization using hypercolumns. *PAMI*, 39(4), 2017. 3

[18] M. Kass, A. P. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4), 1988. 1

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

[20] A. Kirillov, B. Savchynskyy, D. Schlesinger, D. P. Vetrov, and C. Rother. Inferring M-best diverse labelings in a single one. In *ICCV*, 2015. 2

[21] A. Kirillov, A. Shekhovtsov, C. Rother, and B. Savchynskyy. Joint M-best-diverse labelings as a parametric submodular minimization. In *NIPS*, 2016. 2

[22] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *NIPS*, 2011. 1

[23] P. Krähenbühl and V. Koltun. Parameter learning and convergent inference for dense random fields. In *ICML*, 2013. 3

[24] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014. 1

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[26] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 1989. 1

[27] S. Lee, S. Purushwalkam, M. Cogswell, V. Ranjan, D. J. Crandall, and D. Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *NIPS*, 2016. 2

[28] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014. 4

[29] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 5

[30] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[31] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshops*, 2013. 3

[32] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *SIGGRAPH*, 1995. 1

[33] S. Nowozin. Optimal decisions from probabilistic models: The intersection-over-union case. In *CVPR*, 2014. 3

[34] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. J. V. Gool, M. H. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 5

[35] B. L. Price, B. S. Morse, and S. Cohen. Geodesic graph cut for interactive image segmentation. In *CVPR*, 2010. 1

[36] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 8

[37] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut" – interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3), 2004. 1, 5

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. ImageNet large scale visual recognition challenge. *IJCV*, 115(3), 2015. 3

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 4

[40] D. Tarlow and R. S. Zemel. Structured output learning with high order loss functions. In *AISTATS*, 2012. 3

[41] N. Xu, B. L. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *CVPR*, 2016. 1, 2, 3, 5, 7

[42] P. Yadollahpour, D. Batra, and G. Shakhnarovich. Discriminative re-ranking of diverse segmentations. In *CVPR*, 2013. 2

[43] C. Yanover and Y. Weiss. Finding the M most probable configurations in arbitrary graphical models. In *NIPS*, 2003. 2

[44] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 3, 8

[45] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *CVPR*, 2017. 4

[46] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 4