Deep Interactive Thin Object Selection

Jun Hao Liew[1] Scott Cohen[2] Brian Price[2] Long Mai[2] Jiashi Feng[1]
[1] National University of Singapore [2] Adobe Research
liewjunhao@u.nus.edu {scohen,bprice,malong}@adobe.com elefjia@nus.edu.sg

Abstract

Existing deep learning based interactive segmentation methods have achieved remarkable performance with only a few user clicks, e.g. *DEXTR [32] attaining 91.5% IoU on PASCAL VOC with only four extreme clicks. However, we observe even the state-of-the-art methods would often struggle in cases of objects to be segmented with elongated thin structures (*e.g. *bug legs and bicycle spokes). We investigate such failures, and find the critical reasons behind are two-fold: 1) lack of appropriate training dataset; and 2) extremely imbalanced distribution w.r.t. number of pixels belonging to thin and non-thin regions. Targeted at these challenges, we collect a large-scale dataset specifically for segmentation of thin elongated objects, named ThinObject-5K. Also, we present a novel integrative thin object segmentation network consisting of three streams. Among them, the high-resolution edge stream aims at preserving fine-grained details including elongated thin parts; the fixed-resolution context stream focuses on capturing semantic contexts. The two streams' outputs are then amalgamated in the fusion stream to complement each other for help producing a refined segmentation output with sharper predictions around thin parts. Extensive experimental results well demonstrate the effectiveness of our proposed solution on segmenting thin objects, surpassing the baseline by* $\sim$ 30% *IoU*$_{\text{thin}}$ *despite using only four clicks. Codes and dataset are available at* https://github.com/liewjunhao/thin-object-selection.

1. Introduction

Interactive image segmentation task aims to extract a high quality segmentation mask delineating the object-of-interest using only a few user clicks. On this task, deep learning based methods [51, 23, 50, 32, 21, 15, 31, 16, 27, 41] have been very successful. For example, the recent state-of-the-art method DEXTR [32] achieved 90% IoU using only four extreme clicks. Advancement in this task would significantly benefit other tasks like image/video composition, localized image editing and large-scale dataset annotation [4, 2, 28].

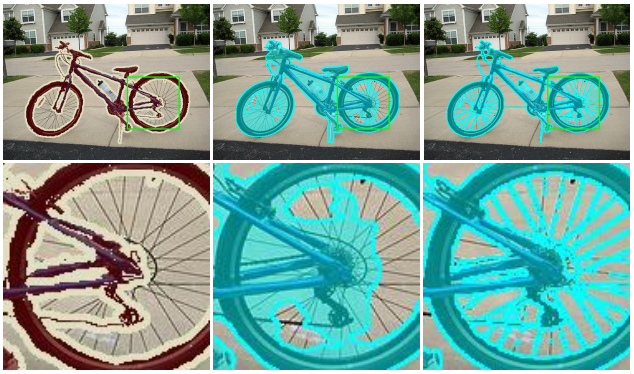
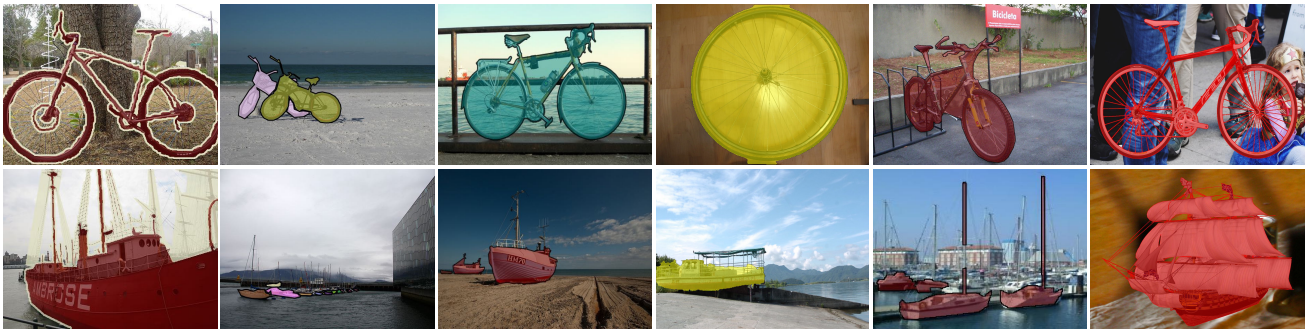


Figure 1: **Left:** Ground truth annotation of PASCAL VOC validation set [9]. Gray lines indicate the "void" label. **Middle:** Segmentation results of DEXTR [32] trained on PASCAL-10K. **Right:** Segmentation prediction by our proposed TOS-Net trained on our ThinObject-5K dataset.

Despite the overall good segmentation performance, the state-of-the-art methods can hardly be applied to professional high-end applications (*e.g.* Photoshop), especially when the objects to be segmented have elongated thin structures (*e.g.* bug legs and bicycle spoke). In such cases, high annotation accuracy is required or otherwise a large number of user clicks or manual delineation is inevitable. As shown in Fig. 1, the current state-of-the-art DEXTR is only capable of producing a rough object mask with details along the thin spokes missing. In this work, we study the reasons behind such failures and provide an effective solution for thin object segmentation.

We attribute the failure of existing methods to the following factors:

1) Poor training data quality: Existing datasets used for training interactive segmentation models (*e.g.* PASCAL [9], COCO [26]) are often coarsely annotated, where fine-grained details including elongated thin parts are ignored (*e.g.* bicycle spokes in Fig. 1(a) and 2), leading to significant label noise. This may also partially explain why most learned models often output "blobby" predictions and fail



PASCAL VOC MS COCO LVIS Open Images V6 ADE20K ThinObject-5K (ours)

Figure 2: Example annotations for 'bicycle' and 'boat' class from existing datasets (*e.g.* PASCAL [9], COCO [26], LVIS [11], Open Images V6 [18] and ADE20K [53]). Elongated thin parts (*e.g.* bicycle spoke and boat stay) are usually ignored during annotation for labeling efficiency. On the other hand, our ThinObject-5K dataset provides much finer annotation.

to capture thin structures (Fig. 3).

2) Class imbalance between thin and non-thin pixels: Most interactive segmentation models adopt per-pixel cross-entropy loss for training as in semantic segmentation. However, this loss treats every pixel independently, which means misclassification of a few pixels would produce low cost to the overall loss. This is especially devastating for delineation of thin structures because pixels belonging to the thin parts usually occupy a small fraction of the entire object. Recent methods employ class-balancing cross-entropy loss [32] or IoU loss [21] to tackle the class imbalance problem between foreground and background samples. Nevertheless, the highly imbalanced distribution between thin and non-thin pixels remains unaddressed.

Based on above observations, in this work we first propose a large-scale dataset for segmentation of thin objects, named ThinObject-5K. It consists of 5,743 unique foreground objects collected from the Internet. Like other synthetic dataset construction pipeline [49], we composite these foreground objects on background images taken from other datasets for training. Some example images of ThinObject-5k are shown in Fig. 4, with elongated thin structures finely annotated.

Also, we present a novel solution specially for thin object segmentation, named TOS-Net. It is designed based on the idea that the distribution between thin and non-thin pixels can be better balanced by converting the learning target to edge-based representation, such that the object interior which mainly comprises of non-thin pixels can be ignored. We thus explicitly separate the processing of high-resolution boundary information, which preserves fine-grained details including thin structures, from that of the context which focuses on capturing semantic contexts. In this way, the two separate streams specialize in different aspects and thus, a refined mask with sharper predictions around elongated parts can be obtained by fusing these two complementary information.

Our contributions are three-fold: 1) we identify the reasons behind the failure in segmenting objects with elongated thin structures; 2) we present a large-scale dataset, ThinObject-5K for interactive segmentation; and 3) we introduce a simple yet effective edge-guided segmentation baseline to tackle thin object selection. Extensive experimental results well validate the effectiveness of both ThinObject-5K and the proposed solution. To our best knowledge, this is the first interactive segmentation work to study the problem of segmenting objects with elongated thin structures under the context of deep learning.

2. Related Works

Deep interactive object segmentation. Xu *et al.* [51] made the first attempt to apply deep learning to interactive segmentation, in which user clicks first go through Euclidean distance transformation before concatenated with the image for training an FCN. Recent methods focus on better exploiting useful context from user-provided inputs [23, 15, 27] or deriving better representation for user clicks [32, 45, 19, 31]. Mahadevan *et al.* [30] proposed an iterative training strategy to reduce the train-test discrepancy. BRS [16] and f-BRS [41] enforce user-specified locations to have correct labels. Li *et al.* [21] and Liew *et al.* [22] addressed the ambiguity in interactive segmentation by enabling multiple hypothesis segmentation. Nonetheless, none of these methods addresses the problem of segmenting objects with elongated thin structures. As we will show in later sections, these approaches often fail when directly applied to thin object segmentation.

Interactive thin object segmentation. Starting with a segmented object part given some scribble inputs, Vicente *et al.* [43] proposed a connectivity prior that connects user-clicked pixels (on thin parts) to the main object. Inspired by

the observation that the color gradient remains uniform almost everywhere along the boundaries of thin parts, CoopCut [17] introduced discount for homogeneous boundaries in segmentation of long thin objects. COIFT [33, 34] incorporates a connectivity constraint on Oriented Image Foresting Transform (OIFT) to facilitate segmentation of connected objects with thin parts. Dong *et al*. [8] proposed a sub-Markov random walk algorithm with label prior. These non-learning-based approaches cannot capture semantics and other high-level priors, and thus are often slow and require extensive user efforts.

Context-edge disentanglement. Several prior works have also explored the idea of decoupling context and edge information to improve segmentation accuracy. For instance, Liu *et al*. [29] introduced an edge-prior branch into the segmentation network for indoor scene parsing. Takikawa *et al*. [42] proposed a new gating mechanism that encourages the shape stream to only focus on processing boundary-relevant information. Li *et al*. [20] explicitly decouple features into body and edge parts, and jointly optimize them in a unified framework. In our case, while more advanced design such as [42, 12, 20] might also work, we purposely opt for simpler design as the main goal of this work is to demonstrate the effectiveness of context-edge disentanglement for thin object segmentation task.

3. Method

3.1. Preliminaries

We first examine performance of existing interactive image segmentation models applied to thin objects. Specifically, we evaluate four state-of-the-art algorithms, including DIOS [51], Latent Diversity [21], DEXTR [32] and f-BRS on a subset of HRSOD [52] dataset which mainly consists of objects with elongated thin parts. We use the official released pre-trained weights (except DIOS) for testing. The performance is evaluated based on the Intersection-over-Union (IoU) at 4th click[1] for fair comparison with DEXTR which requires 4 extreme points as input. We further introduce a new metric IoU_{thin} by evaluating performance only on regions surrounding thin pixels (see Section 4). The results are summarized in Fig. 3.

As shown in the histogram plot, despite the overall good segmentation, the elongated thin parts are not well segmented ($\text{IoU}_{\text{thin}} < 45\%$). Similar observations can be made from qualitative comparison, where the models struggle even given 10 user clicks. These results reveal simply applying state-of-the-art interactive segmentation models to thin objects does not work. Therefore, at below we first introduce a new large-scale dataset for thin object segmentation, and then present a three-stream network for effective segmentation of elongated thin parts with only four clicks.

[1]We employ the standard iterative testing protocol in interactive segmentation literature [51, 23, 21, 41] by iteratively adding clicks to the center of the largest erroneous regions until the 4th click.

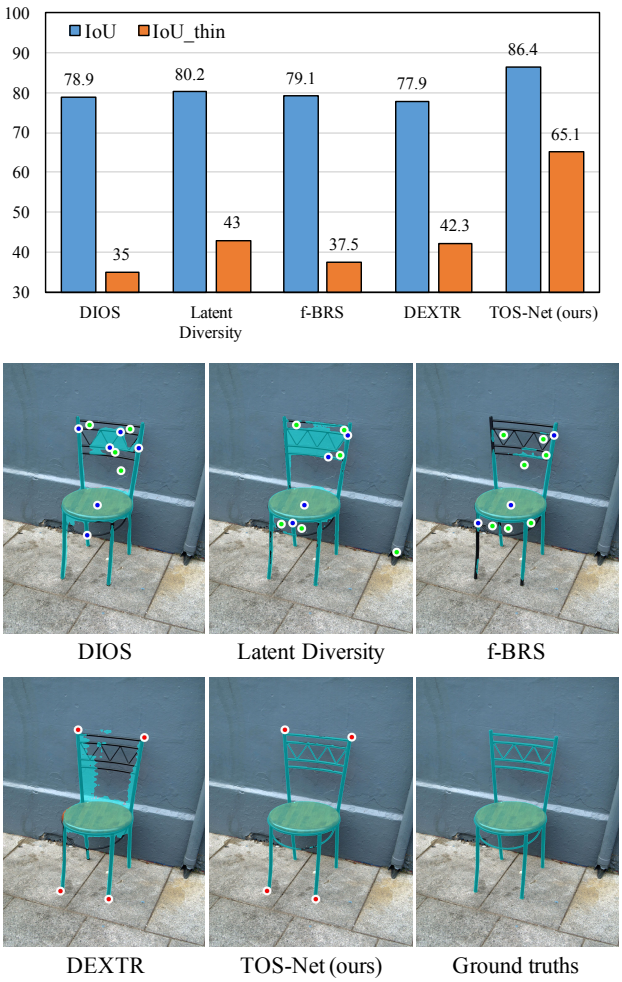


Figure 3: Comparison between our TOS-Net and existing state-of-the-art methods, including DIOS [51], Latent Diversity [21], f-BRS [41] and DEXTR [32]. **Top:** IoU and IoU_{thin} at 4-th click on HRSOD [52] dataset are chosen as the metrics. **Bottom:** The blue, green and red clicks denote the foreground, background and extreme clicks, respectively. We see that these models still struggle to segment elongated thin parts even when given 10 user clicks.

3.2. ThinObject-5K Dataset

Existing segmentation datasets (*e.g*. PASCAL VOC [9], MS COCO [26]) are mostly coarsely annotated, with fine-grained details (including the elongated thin parts *e.g*. bicycle spokes in Fig. 1 and 2) sacrificed during the annotation process for labeling efficiency. Hence they are not suitable for training and testing thin object segmentation models.

We thus collect images with thin structures from the In-

Figure 4: Example images from our ThinObject-5K dataset. Please zoom-in for more details.

ternet and construct a new dataset specially for thin object segmentation, which is named ThinObject-5K. We first source objects with transparent background from http://pngimg.com/ and only retain those with elongated thin parts. Foreground masks can thus be easily extracted from the alpha channel. Despite their high quality annotations in general, we observe some of the images contain labeling noise (*e.g*. halo effects) or incorrect annotations (*e.g*. missing parts). We then manually screen through the images and refine the masks using Photoshop. After that, we have 5,743 unique foregrounds in total.

Similar to [49], we composite the foreground objects on the background images taken from different datasets, leading to a total of 5,743 images with varying size ranging from 32 to 10K pixels. They are split to 4,743, 500 and 500 images for training, validation and testing respectively. Since compositing foreground objects on low-resolution background images such as PASCAL [9] or COCO [26] would lead to poor transferability to real image domain, we leverage two sources of background images, *i.e*., a high-resolution and a low-resolution background for composition depending on the resolution of the foreground object. Specifically, for training and validation set, we use HRSOD [52] and DIV2K [3] as high-resolution backgrounds, and COCO [26] as low-resolution backgrounds. Similarly, we use Flickr2K [24] and PASCAL [9] as high- and low-resolution backgrounds respectively for compositing test set. Some example images are shown in Fig. 4. It can be seen the thin structures, such as ant's legs, racket strings, soccer goal nets, computer mouse cable, harp strings *etc*. are finely annotated.

3.3. Thin Object Selection Network (TOS-Net)

To tackle the extremely imbalanced distribution of thin versus non-thin pixels within an object, we convert the learning target (segmentation mask) to an edge-based representation, such that the interior of the object which mainly consists of non-thin pixels are ignored. Moreover, we also consider better maintaining high-resolution boundary features in order to preserve more visual evidence needed for segmentation of thin structures. However, this would inevitably lead to a trade-off problem between capturing semantic context and retaining fine details. With downsampled inputs, the model captures semantics better but loses high-frequency details. On the other hand, with high-resolution inputs, the receptive field would be too small compared to the image context to capture object-level semantics. Our key idea for solving such an issue is using two separate streams to specialize for each aspect and then fuse them to make the best of both.

We therefore adopt a three-stream design for addressing thin object segmentation, termed Thin Object Selection Network (TOS-Net). As shown in Fig. 5, its three separate streams include: 1) context stream which accepts a *fixed-resolution* input image to extract global context and estimate a rough object segmentation; 2) high-resolution edge stream that processes the input of *high resolution* to delineate the object contours; and 3) fusion stream that fuses the information from the preceding two streams to produce the final mask. Please refer to the supplementary material for more details of our network architecture.

Context Stream. The context stream exploits and aggregates the semantic contextual information describing the object-of-interest for segmentation. For this purpose, a fixed-resolution input is used for better capturing semantics. In particular, we employ DEXTR [32] with minor modifications for this stream due to its simplicity. Similar to [32], our context stream takes four extreme points (top, bottom, leftmost and rightmost pixels) as inputs and converts them to Gaussian heatmaps before concatenating with the image. Then, the bounding box determined from the extreme points is relaxed to include some contextual information before used to crop the input. However, using a fixed value for relaxation as in [32] (50 pixels) is found harmful when tested on images of resolutions that are very different from training data, resulting in almost no context as shown in Fig. 6. We therefore replace the original fixed relaxation with an adaptive alternative as follows. Given a bounding box enclosing the object with box height h and width w, the relaxation r is set to $r = r_{ori} \cdot s_{box}/s_{ave}$, where $s_{box} = \frac{h+w}{2}$ is the average size of the bounding box while $r_{ori} = 50$ refers to the default relaxation value. s_{ave} denotes the average box size in the training set used in [32], which is 428 pixels.

The input image is cropped by the relaxed box and re-

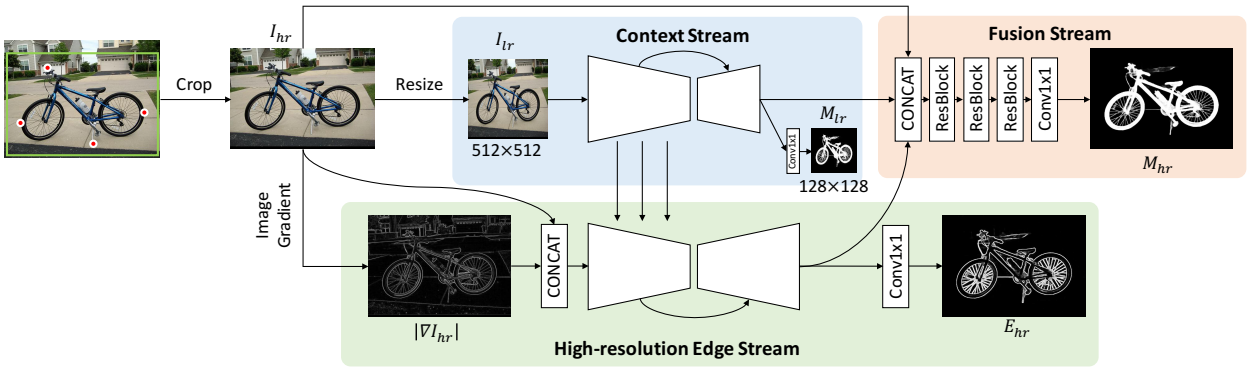


Figure 5: **Overall architecture of our TOS-Net.** Our network consists of three streams. The context streams processes the fixed-resolution cropped input I_{lr} to obtain a rough object segmentation M_{lr}. Taking the high-resolution image I_{hr} and its gradient map ∇I_{hr} as input, the edge stream progressively incorporates multi-level features from the context stream to extract a high-resolution boundary map E_{hr} which preserves image details including elongated thin structures. Finally, both regional and high-resolution boundary features are fused to produce the final segmentation mask M_{hr}.

Figure 6: Fixed *vs*. adaptive relaxation scheme.

sized to a fixed resolution of 512×512. We denote the cropped input before and after resizing as $I_{hr} \in \mathbb{R}^{H\times W\times 4}$ and $I_{lr} \in \mathbb{R}^{512\times 512\times 4}$, respectively, where the 4th channel is a 2D Gaussian heatmap encoding the locations of each extreme point. We then pass the input I_{lr} into a fully convolutional network to produce a rough segmentation mask. In this work, we employ a ResNet-50 [14] variant of DeepLabv3+ [5] as the backbone, which is proven successful across various segmentation tasks [44, 6]. DeepLabv3+ employs an Atrous Spatial Pyramid Pooling (ASPP) module for aggregating multi-scale context, followed by a decoder for refining its segmentation predictions. We append a 1×1 convolution layer with sigmoid activation at the end of the decoder to produce a fixed-resolution binary segmentation mask $M_{lr} \in [0,1]^{\frac{512}{k}\times\frac{512}{k}}$, where $k = 4$ is the network output stride.

High-resolution Edge Stream. The goal of edge stream is to extract object boundaries for guiding the segmentation of thin structures. To prevent too much loss of image details due to input downsampling, particularly for elongated thin parts, the edge stream processes the high-resolution input I_{hr} directly without resizing it to a fixed resolution as the context stream. To facilitate edge learning, we additionally append image gradient ∇I_{hr} as input, which can be easily computed using Sobel filter.

We adopt an FPN-style [25] approach for the edge stream. Specifically, the encoder progressively incorporates multi-level semantic information from context stream while the decoder gradually fuses high-level semantics from the deeper layers with low-level details from earlier layers in the encoder via lateral connection. Injecting semantic information in this way, edge stream can adopt a relatively lightweight and shallow architecture for processing image at high resolution. Similar to context stream, we also append a 1×1 convolution layer with sigmoid activation at the end to generate a boundary map $E_{hr} \in [0,1]^{H\times W}$, which has the same resolution as its input.

Although class-imbalance between thin and non-thin pixels can be mitigated with the proposed edge-based representation, this leads to another class-imbalance between edge (foreground) and non-edge (background) pixels. Fortunately, this issue has been extensively studied in edge detection literature [7, 1]. In this work, we adopt a combination of balanced binary cross-entropy loss [48] and Dice loss [7] for training the edge stream.

Fusion Stream. Fusion stream fuses semantic context from context stream with high-resolution boundary features from edge stream to produce a refined segmentation mask. In particular, it consists of a 1×1 convolution layer for dimension reduction and a series of bottleneck layers [14], followed by a 1×1 convolution and a sigmoid layer to output the final segmentation mask $M_{hr} \in [0,1]^{H\times W}$. Note we do not employ ground truth boundaries as the learning target here, because the high-resolution edge stream already well addresses the imbalanced thin/non-thin pixels distribution and also encodes fine-grained details for segmentation

of elongated thin parts here. Furthermore, in this way post-processing is also unnecessary for reverting the predicted boundaries back to the object mask.

Training Losses. We train our context stream with a balanced binary cross-entropy loss following DEXTR [32]. For edge stream, we employ a combination of balanced binary cross-entropy loss and Dice loss for training. We use a combination of bootstrapped cross-entropy loss [37] and Dice loss to supervise the training of the fusion stream.

Training and Inference. In order to encourage the network to focus on learning the segmentation of elongated thin structures, we randomly crop patches covering thin parts for training. However, context stream which requires access to the full input image is incompatible with such a patch-wise training scheme. To address this problem, we employ an `RoIAlign` [13] layer to extract semantic features from the corresponding patches in context stream when fusing its output with boundary features in edge stream. Similarly, the same `RoIAlign` layer is used when fusing both region and boundary features in fusion stream. During inference, although one can replace the `RoIAlign` layer with a bilinear interpolation operation, we observe this would cause feature misalignment, which subsequently results in significant performance drop, particularly along object boundaries. Instead, we use the `RoIAlign` layer with RoI covering the full input image during testing to mitigate the train-test disparity.

4. Experiments

4.1. Datasets and Settings

Implementation Details. We train our TOS-Net on ThinObject-5k `train` split which consists of 4,743 images. All the images are resized to have a shorter side of at least 512 pixels and a longer side that does not exceed 1,980 pixels. The same resizing operation is applied during inference. We sample 5 cropped patches per image for training, among which 4 patches are sampled from the regions covering thin structures and the remaining patch is randomly sampled from the whole image. We employ a crop size of 512 pixels for training. Random horizontal flip augmentation is applied during training. The network is initialized from ResNet-50 [14] pre-trained on ImageNet [39]. The new layers including the additional channel in the first convolutional layer are randomly initialized from a Gaussian distribution with standard deviation of 0.01. All models are trained with batch size of 1, learning rate of 1×10^{-3} with "poly" learning rate policy, momentum of 0.9 and weight decay of 5×10^{-4}. In order to accommodate the batch size of 1, we follow [10] by replacing all the batch normalization (BN) layers with group normalization (GN) [47] with weight standardization [38]. We train our model using stochastic gradient descent for 50 epochs. All our experiments are conducted on the PyTorch framework [35]. Note that our method does not require any post-processing other than simple thresholding. On average, our TOS-Net takes $0.45s$ for a 1980×1980 cropped image on a PASCAL Titan Xp GPU, thus being suitable for real time applications.

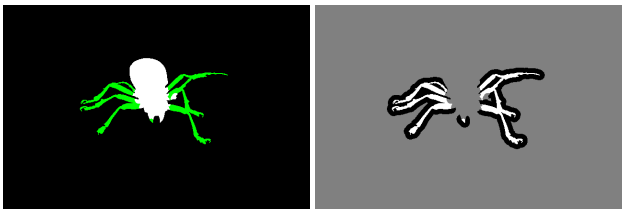


Figure 7: **Left:** Green pixels denote the extracted thin parts. **Right:** Mask used for IoU_{thin} metric where the gray pixels ("void" label) will be excluded from evaluation.

Datasets. We evaluate the performance of our proposed TOS-Net on the following benchmarks:

- **ThinObject-5K (test split).** This dataset contains 500 synthetic images with elongated thin structures.
- **HRSOD [52].** The HRSOD dataset was initially proposed for high-resolution salient object detection tasks. We extract a subset of 287 images which contains 305 objects with elongated and thin parts for evaluation.
- **COIFT [33, 34].** We combine 3 datasets of birds and insects from [33, 34] and denote this dataset as COIFT dataset, which contains 280 images. Note that the average image resolution is much smaller than the other two datasets.

Evaluation Metrics. For all the experiments, we evaluate segmentation performance in terms of IoU given 4 clicks (4 extreme points in our case). However, the actual segmentation performance on elongated thin parts is significantly over-estimated when using the full ground truth mask for evaluation, because the number of pixels belonging to thin regions is often too small compared to the whole objects to contribute meaningfully to the overall IoU. To address this problem, we propose a new metric that measures IoU only on the regions surrounding the thin structures, which is denoted as IoU_{thin}. The detailed steps for extraction of thin structures from ground truth masks for evaluation are explained at below.

Intuitively, for a region covering thin parts, even the innermost pixel should be close to its nearest boundaries. We therefore first compute a Euclidean distance transformation of the ground truth object mask. We then extract the local peaks in the distance map and only retain those peaks whose distance values are smaller than a threshold τ. Finally, we aggregate all the foreground pixels surrounding these peaks to obtain the thin parts. For evaluation, we also consider a thin strip of background pixels around the extracted thin regions to prevent a trivial solution that predicts the entire

No.	Train Data	Input Res	Edge Stream	Training Loss	DGF	ThinObject-5K			COIFT [33]			HRSOD [52]		
						IoU	IoU_{thin}	$\mathcal{F}$	IoU	IoU_{thin}	$\mathcal{F}$	IoU	IoU_{thin}	$\mathcal{F}$
M1	PASCAL-10K	Fixed	No	BBCE	No	61.8	43.5	49.0	70.6	36.8	74.4	69.5	35.4	66.1
M2	ThinObject-5K	Fixed	No	BBCE	No	88.8	74.0	89.3	88.2	68.3	93.4	82.5	57.7	84.8
M3	ThinObject-5K	Fixed	No	3-class balanced	No	79.9	61.6	76.5	79.8	51.0	86.7	75.3	46.8	77.3
M4	ThinObject-5K	Fixed	No	BootCE+Dice	No	91.0	77.9	91.8	90.3	72.1	94.3	83.3	59.0	85.9
M5	ThinObject-5K	High	No	BootCE+Dice	Yes	91.6	79.2	91.8	89.9	70.0	92.6	82.8	61.6	84.8
M6	ThinObject-5K	Fixed	Yes	BootCE+Dice	No	89.7	75.8	89.8	89.8	69.7	91.9	81.7	56.8	83.4
M7	ThinObject-5K	High	Mask	BootCE+Dice	No	93.9	85.5	93.6	**92.2**	**77.9**	94.2	84.4	63.6	82.9
M8 (ours)	ThinObject-5K	High	Yes	BootCE+Dice	No	**94.3**	**86.5**	**94.8**	92.0	76.4	**95.3**	**86.4**	**65.1**	**87.9**

Table 1: Quantitative results on ThinObject-5K test set, COIFT [33] and HRSOD [52] dataset. DGF: deep guided filter [46]; Input Res: input resolution; BBCE: balanced binary cross-entropy loss; BootCE: boostrapped cross-entropy loss.

image to be foreground. More details can be found in the supplementary material. An example is shown in Fig. 7 where the grey pixels denote "void" labels that will be excluded from evaluation of IoU_{thin}.

In addition, we also adopt the boundary measure $\mathcal{F}$ from video object segmentation literature [36] to better assess the quality of segmented edges.

4.2. Ablation Studies

We perform ablation experiments to justify various design choices for tackling the thin object segmentation task. The results are summarized in Table 1. All the models adopt the same backbone, *i.e.* ResNet-50-based DeepLabv3+ [5], for fair comparison. Note that the model M1 and M2 without edge stream (and fusion stream) simply reduce to DEXTR [32] except for using our proposed adaptive relaxation scheme; M8 denotes our full TOS-Net model.

Finely annotated training data. When trained on our ThinObject-5K dataset, we can see that M2 outperforms the counterpart trained on PASCAL-10K (M1) by a large margin, suggesting that finely annotated datasets offer much benefit to improving thin object segmentation.

Learning target. We train a baseline to output 3-class segmentation (background/thin/non-thin) using a balanced softmax cross-entropy loss that weighs each pixel with the inverse normalized frequency of labels within a minibatch. During testing, the predictions of thin and non-thin classes are combined to obtain the final output. However, this results in a significant performance drop (M3 *vs*. M2) possibly due to the heuristic definition of thin and non-thin pixels, which makes the learning difficult.

Training losses. When comparing M4 and M2, we can see that thin object segmentation is significantly benefited from using bootstrapped cross-entropy loss and Dice loss for training. Bootstrapped cross-entropy can be interpreted as Online Hard Example Mining (OHEM) [40] where only gradients for the examples leading to the top K highest loss (which are usually thin pixels) are back-propagated whereas Dice loss jointly optimizes the global similarity between the network prediction and ground truth. Altogether, these two losses are beneficial to the task of thin object segmentation.

High-resolution edge information. When comparing M8 to M4, we can see that edge information plays a crucial role in thin object segmentation. However, interestingly, the performance degrades when a fixed-resolution edge stream is used (M6 *vs*. M4). This is possibly because downsampled inputs suffer from loss of high-frequency details, and therefore are not suitable for thin object segmentation where fine-grained details particularly matter. Nevertheless, one may argue that the performance gain from M4 to M8 is partially due to the increased number of parameters. We therefore disentangle the two sources of performance boost by comparing with a baseline whose network architecture is the same as our TOS-Net, but edge supervision now replaced with mask supervision (M7). While performing better on the smaller COIFT dataset, we notice a significant performance drop on both ThinObject-5K and HRSOD datasets when removing edge supervision. This verifies our finding that high-resolution boundary information is essential to addressing the highly imbalanced thin/non-thin pixels distribution in mask-based representation.

High-resolution features. To further investigate the impact of high-resolution features upon thin object segmentation performance, we additionally compare our models with a baseline that incorporates high-resolution inputs for refinement. Specifically, we insert a deep guided filtering layer (DGF) [46] at the end of M2 to recover the details of elongated thin structures when upsampling back to its original resolution. We can see this model (denoted as M5) is still much inferior to our TOS-Net. This demonstrates the effectiveness of our proposed three-stream design for the thin object segmentation task.

4.3. Qualitative Results

We also present some qualitative comparison between our TOS-Net (M8) and DEXTR [32] (M2) given the same set of user inputs (Fig. 8). As compared to the baseline, we can see that our TOS-Net in general produces crisper seg-

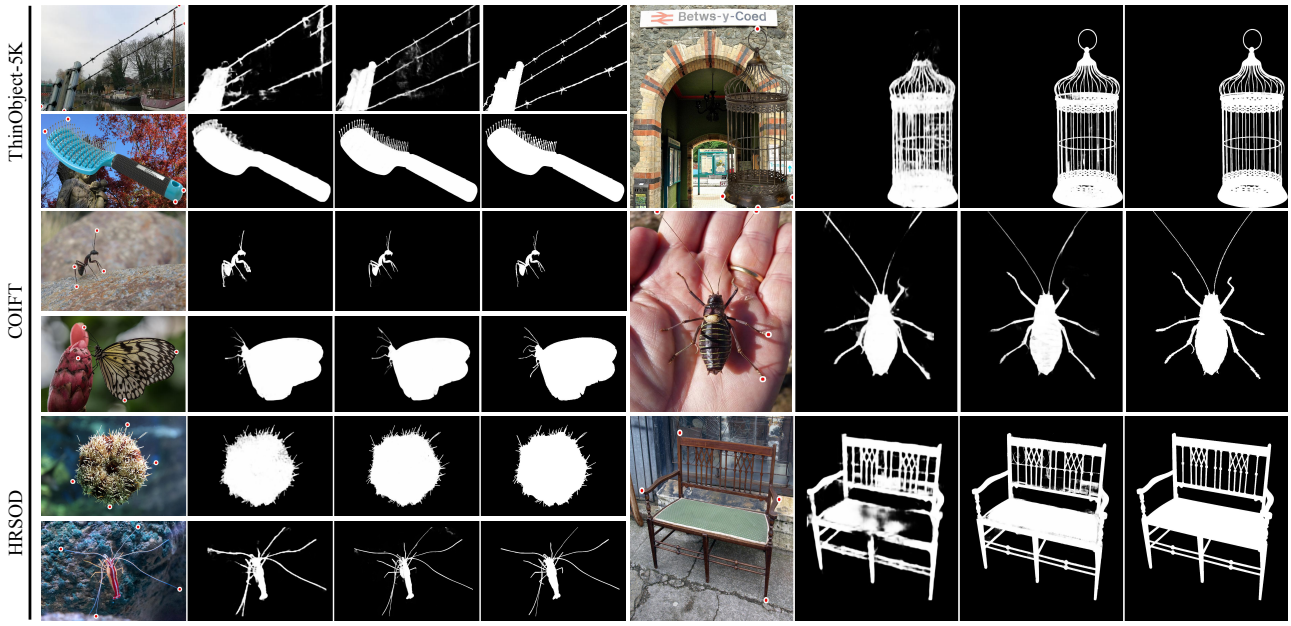


Figure 8: Qualitative results on ThinObject-5K, COIFT [33] and HRSOD [52] datasets. Each result is arranged in the order of: (i) input image, (ii) DEXTR [32] trained on ThinObject-5K dataset (M2), (iii) our TOS-Net (M8) and (iv) ground truths. The input extreme clicks are marked red.

mentation, especially along thin parts (*e.g*. hair brush, bird cage, bench). Moreover, our model performs well even on challenging scenes where the color distribution of thin parts and background significantly overlap, *e.g*. pole cables, butterfly and bug antenna. More qualitative examples can be found in the supplementary material.

4.4. Limitations

We also show some failure cases of our approach in Fig. 9. We notice that ThinObject-5K-trained models (both DEXTR and our TOS-Net) exhibit strong bias towards selecting the segmentation with elongated thin structures when the same set of extreme points allow multiple possible segmentations (*e.g*. sheep or fence in the first row of Fig. 9). However, depending on applications, this behaviour could be desirable. Another limitation lies in the incapability of tackling transparent objects, which has been an unsolved problem in computer vision.

Figure 9: Failure cases. Our model is **(top)** biased towards segmenting thin structures when the extreme points allow multiple possible segmentations and **(bottom)** unable to handle transparent object.

5. Conclusion

In this work, we first studied the critical factors behind the failure of segmenting objects with elongated thin structures. We find that lack of finely annotated training data and extremely imbalanced thin/non-thin pixels distribution are the main reasons for the poor performance of existing interactive segmentation models. To address these problems, we presented a large-scale dataset specifically for segmentation of thin elongated objects, named ThinObject-5K. In addition, we design a three-stream network called TOS-Net that integrates high-resolution boundary information with fixed-resolution semantic contexts for effective segmentation of thin parts. Extensive experimental results well demonstrate the effectiveness of our proposed solution.

Acknowledgement. Jiashi Feng was partially supported by AISG-100E-2019-035, MOE2017-T2-2-151, NUS_ECRA_FY17_P08 and CRP20-2017-0006. This work is supported in part by gifts from Adobe.

References

[1] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *CVPR*, 2019.

[2] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with Polygon-RNN++. In *CVPR*, 2018.

[3] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, 2017.

[4] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a Polygon-RNN. In *CVPR*, 2017.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[6] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas S Huang, Wen-Mei Hwu, and Honghui Shi. SPGNet: Semantic prediction guidance for scene parsing. In *ICCV*, 2019.

[7] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *ECCV*, 2018.

[8] Xingping Dong, Jianbing Shen, Ling Shao, and Luc Van Gool. Sub-markov random walk for image segmentation. *TIP*, 2015.

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 2010.

[10] Marco Forte, Brian Price, Scott Cohen, Ning Xu, and François Pitié. Getting to 99% accuracy in interactive segmentation. *arXiv preprint arXiv:2003.07932*, 2020.

[11] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[12] Ali Hatamizadeh, Demetri Terzopoulos, and Andriy Myronenko. End-to-end boundary aware networks for medical image segmentation. In *MLMI*, 2019.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Yang Hu, Andrea Soltoggio, Russell Lock, and Steve Carter. A fully convolutional two-stream fusion network for interactive image segmentation. *Neural Networks*, 2019.

[16] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *CVPR*, 2019.

[17] Stefanie Jegelka and Jeff Bilmes. Cooperative cuts for image segmentation. Technical report, Technical Report 2010-0003, University of Washington, 2010.

[18] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[19] Hoang Le, Long Mai, Brian Price, Scott Cohen, Hailin Jin, and Feng Liu. Interactive boundary prediction for object selection. In *ECCV*, 2018.

[20] Xiangtai Li, Xia Li, Li Zhang, Cheng Guangliang, Jianping Shi, Zhouchen Lin, Yunhai Tong, and Shaohua Tan. Improving semantic segmentation via decoupled body and edge supervision. In *ECCV*, 2020.

[21] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, 2018.

[22] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, Sim-Heng Ong, and Jiashi Feng. MultiSeg: Semantically meaningful, scale-diverse segmentations from minimal user input. In *ICCV*, 2019.

[23] Jun Hao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *ICCV*, 2017.

[24] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017.

[25] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[27] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *CVPR*, 2020.

[28] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with Curve-GCN. In *CVPR*, 2019.

[29] T. Liu, Y. Wei, Y. Zhao, S. Liu, and S. Wei. Magic-wall: Visualizing room decoration by enhanced wall segmentation. *TIP*, 28(9):4219–4232, 2019.

[30] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. In *BMVC*, 2018.

[31] Soumajit Majumder and Angela Yao. Content-aware multi-level guidance for interactive instance segmentation. In *CVPR*, 2019.

[32] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 2018.

[33] Lucy AC Mansilla and Paulo AV Miranda. Oriented image foresting transform segmentation: Connectivity constraints with adjustable width. In *SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2016.

[34] Lucy AC Mansilla, Paulo AV Miranda, and Fábio AM Cappabianco. Oriented image foresting transform segmentation with connectivity constraints. In *ICIP*, 2016.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison,

Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019.

[36] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[37] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *CVPR*, 2017.

[38] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight standardization. *arXiv preprint arXiv:1903.10520*, 2019.

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[40] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016.

[41] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-BRS: Rethinking backpropagating refinement for interactive segmentation. In *CVPR*, 2020.

[42] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-SCNN: Gated shape cnns for semantic segmentation. In *ICCV*, 2019.

[43] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008.

[44] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. FEELVOS: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019.

[45] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L Divid, Jan Deprest, Sébastien Ourselin, et al. DeepIGeoS: a deep interactive geodesic framework for medical image segmentation. *TPAMI*, 2018.

[46] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Fast end-to-end trainable guided filter. In *CVPR*, 2018.

[47] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

[48] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015.

[49] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *CVPR*, 2017.

[50] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep GrabCut for object selection. In *BMVC*, 2017.

[51] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *CVPR*, 2016.

[52] Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In *ICCV*, 2019.

[53] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.