Interactive Image Segmentation: From Graph-based Algorithms to Feature-Space Annotation

Jordão Bragantini
Chan Zuckerberg Biohub
San Francisco, California, USA
Email: jordao.bragantini@czbiohub.org

Alexandre Xavier Falcão
University of Campinas
Campinas, São Paulo, Brazil
Email: afalcao@ic.unicamp.br

***Abstract*—In recent years, machine learning algorithms that solve problems from a collection of examples (*i.e.* labeled data), have grown to be the predominant approach for solving computer vision and image processing tasks. These algorithms' performance is highly correlated with the abundance of examples and their quality, especially methods based on neural networks, which are significantly data-hungry. Notably, image segmentation annotation requires extensive effort to produce high-quality labeling due to the fine-scale of the units (*pixels*) and resorts to interactive methodologies to provide user assistance.**

Therefore, improving interactive image segmentation methodologies with the goal of improving data labeling problems is of paramount importance to advance applications of computer vision methods. With this in mind, we investigated the existing literature on interactive image segmentation, contributing to it by introducing novel algorithms that perform the segmentation from markers, contours, and finally proposing a new paradigm for image annotation at scale.

I. Introduction

Image segmentation concerns splitting an image into segment (*i.e.* regions) with similar characteristics. It is a significantly difficult problem due to the number of units being analyzed (*e.g.* a mobile image contains more than a million pixels), the dependency between neighboring regions, and the ambiguity of precisely defining a segment (*e.g.* a whole tree can be considered a segment or it can be partitioned into branches and leaves). Due to these obstacles, interactive methods are indispensable tools that allows achieving the most accurate results to provide examples for learning-based methodologies or when they fail.

Moreover, with the advancement of automatic methods based on Convolutional Neural Networks (CNN) [1]–[3], the necessity of high-quality annotated data increased drastically, especially in the context of segmentation, which requires significantly more effort than other image-related tasks. For example, the ImageNet dataset [4] for image classification had surpassed more than 10 million labeled images by 2012, while a much more recent dataset, LVIS [5] from 2019, contains annotations of 2.2 million high-quality segmentation instances (around 164 thousand images), being a magnitude larger than other segmentation datasets [6], [7], but at the same time a fraction of the older ImageNet dataset. Hence, interactive segmentation techniques are of utmost importance to assist the production of high-quality labels with low effort so that users can annotate several images quickly.

Interactive image segmentation techniques combine the complementary competencies of humans and machines [8]. Humans can quickly identify objects and machines can process a large amount of data in well-defined tasks. Thus, in this work and most of the literature [8]–[14], the interactive image segmentation paradigm employs the user for detecting the object or region of interest and the machine for segmenting (*i.e.* partitioning, grouping, delineating) the images.

Furthermore, CNN methods also contributed to the development of novel techniques for interactive image segmentation [13], [15]–[20], significantly reducing the annotation burden, but introducing new problems, as they can greatly approximate the desired object's shape but fail at responding to user interactions, displaying significant bias towards results seen during training.

In contrast, classical graph-based approaches are very responsive to user input but require extensive interaction to perform accurately, and it is an ongoing challenge, to find the most effective way of combining these methodologies to fully exploit their complementary advantages.

Accordingly, this work's contributions start from graph-based techniques, which were the most popular and successful techniques preceding CNN-based approaches. With the CNN's progress, we shifted our goals to identifying limitations on the current annotation procedures and proposing a novel methodology for large-scale annotation.

In summary, our work contributions are:

- [14]: We developed a novel methodology for graph-based interactive image segmentation that dynamically estimates the arc-weight between pixels during the region growing process of the Image Foresting Transform (IFT) operator, thus providing a more robust estimate less sensitive to noise — without requiring training data or transfer learning.

This work derives from Jordão's M.Sc. Dissertation from University of Campinas

- [21]: To complement CNN-based methods for interactive segmentation, where user control is sacrificed for the networks' predictive power, we developed a new technique, called Grabber, to accurately correct segment without ruining the correct regions.
- [22]: We proposed a novel methodology for annotating images' segments that allow labeling multiple images at once on their feature space, speeding up the annotation process when redundant information is present.

While these contributions were developed independently in a sequence as our study progressed, they complement each other. For example, the methodology for large-scale annotation results in coarse segments that can be corrected with the graph propagation methodology or the contour-based method. Additionally, the contour-based method provides greater control and can be used to refine the graph algorithm results.

The implementation of all the methodologies described in this work are publicly available at:

- https://github.com/PyIFT/pyift is the library with the algorithm proposed in Section II, [14], and the back-end of III's methodology, [21].
- https://github.com/LIDS-UNICAMP/grabber the napari plugin [23] of the tool presented in Section III, [21].
- https://github.com/LIDS-UNICAMP/rethinking-interactive-image-segmentation contains Section IV tool/methodology, [22].

This article is organized such that, Sections II- IV describes briefly the methodologies presented in the author's M.Sc. dissertation [24] and respectively, their major results. Next, we conclude with the primary conclusions from this study. The necessary theoretical background for this work can be found in Chapter 2 of [24].

II. Dynamic Trees Image Foresting Transform

Image segmentation is challenging and often requires users' assistance for correction, among the many established approaches to solving the interactive segmentation problem, graph-based algorithms from user-defined markers have showed to be quite effective [11], enjoying a developed theoretical background [25], [26], and being easily extendable (sometimes no change is necessary) to the semi-supervised classification domain (*i.e.* transductive learning) for non-image data [27].

However, most of these approaches resolve the segmentation on static graphical models [9], [11], [28]–[30] starting from user defined markers (*i.e.* labeled nodes in the graph), and the unlabeled data information is only partially used during the segmentation process, providing a pathway to propagate labels over the graph, but without updating the propagation's strength as additional labels are estimated. Other techniques [10], [31] update each pixels' distribution function (*i.e.* **node weight**) through multiple executions of the same algorithm.

Our novel approach dynamically estimates the graph's **arc weight** as the segmentation is computed, in a region-growing fashion, on a single execution. Thus, improving the model with unlabeled data information as the segmentation advances.

A. Methodology

Existing algorithms can solve interactive segmentation in real-time only on limited scenarios; for example, the maxflow-mincut algorithm [32] is NP-hard beyond the binary case, the number of linear systems required to solving the Random-Walk [29] increases with the number of distinct labels, and on both cases the final label of any pixel is only fixed upon the algorithm's convergence. Moreover, they can only be executed on graphs with static weights, restricting the objective functions that they can optimize.

In contrast, the IFT framework [9], performs the segmentation in a region-growing manner, fixing nodes with immutable labels as the optimum-path propagates, and obtaining satisfactory results even when its original assumptions [26] are violated [33], [34].

We noticed that additional information beyond the labeled nodes can be used to improve the optimum-path routing between nodes with weak connectivity. For that, we proposed to explore the information of nodes with strong connectivity to provide this additional data. A simple, yet effective criterion to achieve this goal is to measure how much a weakly connected node (*i.e.* pixel) differs from the average of strongly connected components, this difference is computed as their Euclidean distance on the feature space (*e.g.* color), and as the segmentation progresses the strongly connected components grows and its distribution changes. To our knowledge, this is the first time dynamic arc weights have been explored for image segmentation.

Using dynamic programming the moving average of the components, and the arc weights can be computed without increasing the time complexity of the original algorithm, a in-depth description can be found in [14], [24]. We called this algorithm Dynamic Trees (DT).

From the DT algorithm, multiple variants of the moving average arc weight were proposed given different assumptions over the regions of interest properties (*i.e.* pixels' feature). The main functions being, a single average per label (*i.e.* object, DT_L), a single average per optimum-path tree (*i.e.* root, DT_R), and moving average with exponential decay along the path (DT_{exp}). The later, enjoying the property of being differentiable [35]. Refer to [14], [24] for additional details.

B. Experiments

We compared the proposed approaches with classical algorithms that have stood the test of time, GraphCut (GC) [32], Random Walks (RW) [29], Watershed Cuts (WS) [36], IFT [9], Power Watershed with $q = 2$ (PW) [11], and more recent approaches, One Cut (OC) [31] and Laplacian Coordinates (LP) [37], in two datasets: GrabCut [10] dataset that contains 50 images with the markers provided by Andrade and Carrera [38]; DAVIS dataset [7] of foreground segmentation on videos, following [16], 10% of the frames were sampled resulting on 345 images. The images were evaluated on the RGB color space and the arc weights of all methods are a function of the pixels' color Euclidean distances. Results presented in Table I.

Method	Intersection over Union	
	Grabcut Andrade [38]	DAVIS [7]
RW [29]	0.727 ± 0.159	0.784 ± 0.148
GC [32]	0.746 ± 0.156	0.761 ± 0.148
WS [36]	0.800 ± 0.138	0.787 ± 0.143
OC [31]	0.728 ± 0.207	0.601 ± 0.218
LP [37]	0.764 ± 0.158	0.809 ± 0.140
PW [11]	0.800 ± 0.138	0.788 ± 0.143
IFT [9]	0.798 ± 0.137	0.788 ± 0.143
DT_L [14]	0.691 ± 0.192	0.676 ± 0.145
DT_{exp} [14]	0.816 ± 0.132	0.798 ± 0.142
DT_R [14]	$\mathbf{0.832 \pm 0.133}$	$\mathbf{0.822 \pm 0.136}$

TABLE I: Interactive segmentation quantitative results.

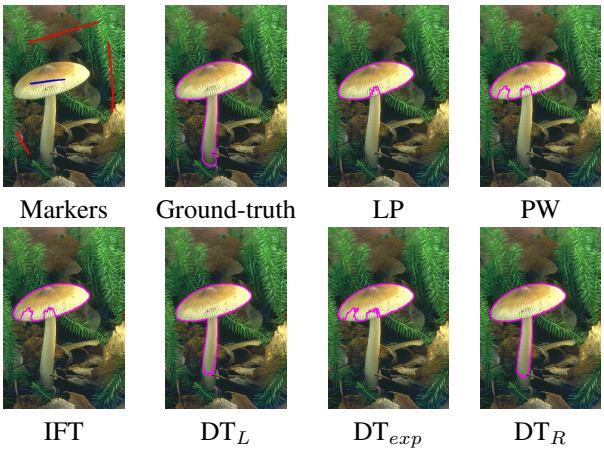

Fig. 1: Qualitative results on Microsoft's dataset. Foreground markers are blue and background are red. Segmentation contour in magenta. Where the presented DTs variants are regarding the labels (L), exponential decay (exp) and root (R).

Figure 1 presents the segmentation results of the best performing methods on an image from the Grabcut dataset. The DT_L obtained satisfactory result due to the homogeneous characteristics of the object, which can be effectively summarized in a single mean, the same can be said to DT_R, which extends it to multiple means, the variant with exponential decay (exp), produced results similar to the watershed. Hence, an increase in the moving average autocorrelation parameter might yield results more similar to the other variants.

III. Grabber

As described previously, CNN-based interactive segmentation methods are the state-of-the-art and significantly reduced the amount of user interaction. However, when faced with challenging scenarios they neglect the user constraints (*i.e.* input).

With *Grabber* we addressed the above problem, providing a tool to improve the user control, assisting the conclusion of segmentations from any method, automatic or interactive.

Grabber estimates anchor points from a user-provided segmentation mask and sorts the points in one boundary orientation, rather than requiring the user to provide a sequence of anchor points in a given order along the boundary as in Live-Wire [8]. The object's delineation is obtained by an optimum contour constrained to pass through the anchor points. It

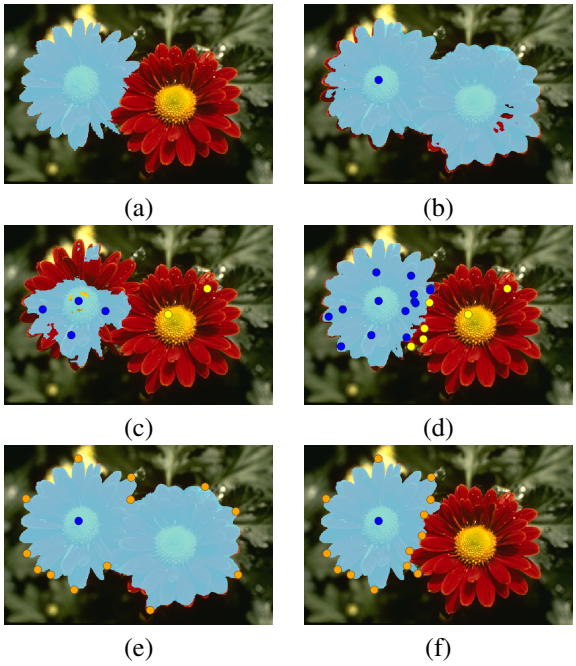

Fig. 2: (a) Original image with ground-truth segmentation. (b), (c), (d) Results of fBRS [17] after 1, 6, and 25 iterations, respectively, of a robot user that inserts internal (blue) and external (yellow) points. One can see that corrections in (c) ruin correct parts of the result from (b). (e) Grabber can improve the result from iteration 1 of fBRS, delineating an optimum contour constrained to pass through estimated anchor points (orange) by letting the user move, add, and remove anchor points, (f); Grabber converged faster with higher accuracy and with 13 fewer user interventions.

can also leverage object's properties from the initial coarse segmentation (*e.g.* internal and external probability density maps) to improve boundary delineation. The user can control object delineation in any part of the boundary by adding, removing, and moving anchor points.

In order to demonstrate the impact of Grabber to increase convergence in interactive segmentation, we integrate it with two recent approaches, a CNN-based method, fBRS [17], and a graph-based method, named DT [14] presented in the previous Section. Figures 2(e)-(f) illustrate its potential to improve interactive segmentation when integrated with fBRS.

A. Methodology

Given a segmentation mask, we estimate anchor points along the boundary of the mask and let the user manipulate them (*e.g.* add, remove, and move points). The initial sorted anchor points are obtained using the Douglas-Peucker algorithm [39] with a curvature-approximation threshold ϵ. When a user manipulate the anchors, it estimates the object's boundary as an optimum contour constrained to pass through that and the adjacent anchors, resulting in an delineation that adheres to the object's contour.

At each interaction, the boundary adherent contour is obtained by solving a minimum cut problem, where we want to

split the graph (*i.e.* image) into two disconnected components (*i.e.* object and background), such that the removed edges have minimum sum, with the additional constraints that the untouched anchors and their respective contour segments must remain the same. Given these constraints, the optimization problem can be solved using dynamic programming, as proposed in [8], allowing it to work in real time even in large images, an requirement for interactive algorithms.

Additionally, we proposed an arc weight that can leverage priors over the object of interest,

$$w(p,q) = e^{-\frac{\|I(l_{p,q}) - I(r_{p,q})\|}{\sigma_I}} e^{-\frac{\|f(l_{p,q}) - f(r_{p,q})\|}{\sigma_f}}$$

where p, q are adjacent pixels (*i.e.* nodes), $l_{p,q}$ is the pixel at their left side and $r_{p,q}$ at their right, $I(.)$ access a pixel color space and $f(.)$ their density map, feature space or any other kind of prior on the pixel grid, σ_I and σ_f balances how much the node should adhere to the color's (feature, density) gradient, if both σ's tends to ∞ the resulting cut is a straight line connecting the anchors and if one the σ's tends to zero the path is the route with minimum $\max_{p,q} w(p,q)$ and more susceptible to noise. The algorithm is shown in details in [21].

B. Experiments

This section evaluates Grabber combined with two methods, fBRS [17] and DT [14] and their standalone versions. The combined approaches are called DT-w_I, fBRS-w_I, and fBRS-w_f. In w_I there is no prior and only the $I(.)$ term is included, in w_f the prior is the CNN prediction, Equation III-A.

We adopted a stress experiment similar to the convergence analysis from [17]. It measures the number of interactions required to achieve a fixed threshold of Intersection over Union (IoU) — *i.e.* the number of clicks/anchor manipulations required to achieve 0.95 IoU (NoC@0.95) limited to 50 interactions. And the total of images which did not achieve the desired score given the threshold is also reported. The experiments used 100 images of the testing set of Berkeley [40] from [41], and the DAVIS dataset as described in the Section II-B.

To simulate a user, competing methods used [13] robot user. Since Grabber operates along the contours, we implemented a new robot, it simulates a user by locating the largest erroneous component for correction, and then it decides to insert remove, add, or drag the component's anchor to the closest point the ground-truth border given a set of predefined rules.

Table II show that the integration of fBRS and DT with Grabber can increase the mean IoU and decrease the number of user interactions because it provides greater control. Additional results and details of experiments, our robot user implementation, parameters and the fBRS network setup can be found in [21].

Method	Dataset	# Img $\geq$ 50	NoC@0.95	Grabber (%)
fBRS	Berkeley	23	16.77	-
fBRS-w_I	Berkeley	**12**	14.53	42.0
fBRS-w_f	Berkeley	**12**	**14.02**	43.0
DT	Berkeley	31	27.71	-
DT-w_I	Berkeley	**22**	**26.77**	71.0
fBRS	DAVIS	133	24.83	-
fBRS-w_I	DAVIS	**93**	**24.49**	65.8
fBRS-w_f	DAVIS	100	24.74	65.8
DT	DAVIS	163	39.07	-
DT-w_I	DAVIS	**139**	**37.85**	91.6

TABLE II: For each method and dataset, the number of images which it could not achieve 0.95 IoU in 50 interactions (bold indicates better), average NoC@0.95, and percentage of images that required Grabber.

IV. Feature Space Annotation

While our previous work and most of the literature [12], [13], [15]–[20], [42]–[44] focus on the *microtask* of segmenting a single object, the big picture in today's segmentation labeling is that thousands of images with multiple objects require annotation. While these objects might not share the same appearance, their semantics are most likely related. Hence, thousands of interactions to obtain thousands of segments with similar contexts do not sound as appealing as before.

Therefore, we proposed a scheme for interactive large-scale image annotation that allows labeling of many similar segments at once. It starts by defining segments from multiple images and computing their features with a neural network pre-trained in another domain. The user annotation is done on a projection of the data feature space, Figure 3, and as it progresses, the similarities between segments are updated with metric learning, increasing the discrimination among classes, and further reducing the labeling burden.

To our knowledge, this was the first interactive image segmentation methodology that does not receive user input on the image domain. Hence, our goal was not to beat the state-of-art of interactive image segmentation but to demonstrate that other forms of human-machine interaction, notably feature space interaction, can benefit the interactive image segmentation paradigm and can be combined with existing methods to perform more efficient annotation.

A. Methodology

The proposed methodology is summarized in Figure 4, the user interface is composed of two primary components, the Projection View and the Image View. Red contours in Figure 4 delineate which functionalities are present in these widgets. The Projection View is concerned with displaying the segments arranged in a canvas (Figure 3), enabling the user to interact with it: assigning labels to clusters, focusing on cluttered regions, and selecting samples for correction in the image domain. Image View displays the image containing a selected segment from the canvas. The selected segment is highlighted to allow fast component recognition among the other segments' contours. Samples already labeled are colored by class. This widget allows further user interaction to fix incorrect delineation, like DT from Section II.

The colored rectangles in Figure 4 represent data processing stages: yellow represents fixed operations that are not updated during user interaction, red elements are updated as the user

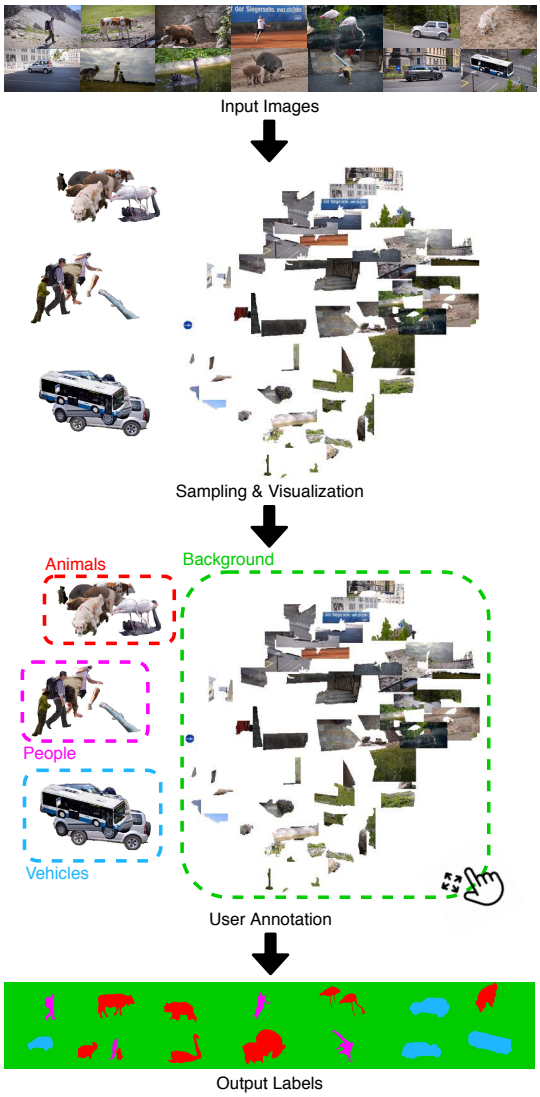

Fig. 3: Our approach to interactive image segmentation: candidate segments are sampled from the dataset and presented in groups of similar examples to the user, who annotates multiple segments in a single interaction.

annotation progresses, and the greens are the user interaction modules. Arrows show how the data flows in the pipeline.

Our implementation of the methodology works as follows, starting from a collection of images, their boundaries are computed using an off-the-shelf edge-detection CNN [45], from this, we partition each image into segments using watershed [46] — these segments are the units that will be processed and annotated in the next stages.

The next step concerns with representing the notion of similarity between segments as perceived by the user. We propose communicating this information to the user by displaying samples with similar examples in the same neighborhood. Hence, we extract deep features [3] from the segments and reduce the features dimensionality using UMAP [47] to embed the units into a 2D plane while preserving, as best as possible, their relative feature space distances.

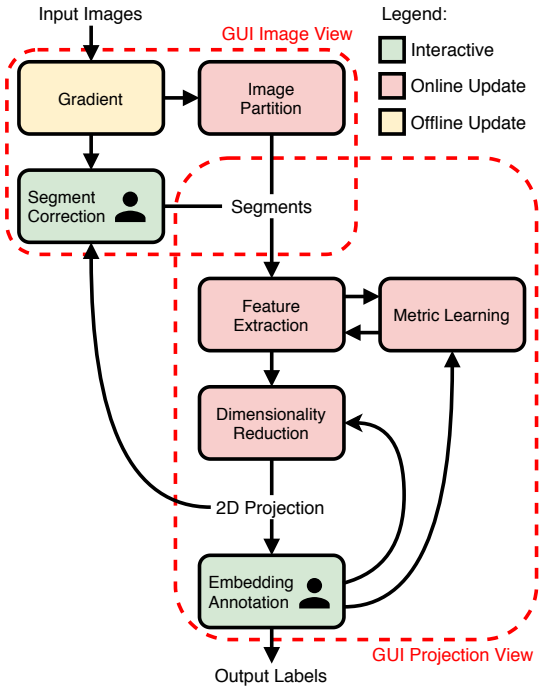

Fig. 4: The proposed feature space annotation pipeline.

The user labeling process is executed in the 2D canvas by defining a bounding-box and assigning the selected label to the segments inside it. As the labeling progresses, their deep features are updated using metric learning [48], improving class separability, enhancing the 2D embedding, thus, reducing the annotation effort.

This pipeline relies only upon the assumption that it is possible to find meaningful candidate segments from a set of images and extract discriminant features from them to cluster together similar segments. Even though these problems are not solved yet, existing methods can satisfy these requirements. Refer to [22] for additional details.

B. *Experiments*

We quantitatively compare our method with existing baselines on three foreground segmentation datasets: iCoSeg [49], 643 natural images that belong to the 38 different context (*e.g.* same location or event); DAVIS, as described in II-B; Rooftop [50], a remote sensing dataset with 63 images that contains 390 instances of disjoint rooftop polygons.

We executed our own experiments according to the code availability of the state-of-the-art methods; Them being, fBRS-B [17] and FCANet [20]. We are not comparing with IOG [19] because we could not reproduce their results (subpar performance) with the available code and weights, and [18] is not publicly available.

Table III report the average IoU and the total time spent in annotation. Click-based methods used [13]'s robot user, the interaction time was estimated as 2.4s for the initial click and 0.9s for additional clicks, as measured in [51].

We achieve comparable accuracy results with state-of-the-art methods while employing less sophisticated segmentation

Dataset	iCoSeg		DAVIS		Rooftop	
Method	IoU	Sec.	IoU	Sec.	IoU	Sec.
fBRS (3)	79.82	4.2	79.87	4.2	62.57	4.2
fBRS (5)	82.14	6	82.44	6	74.53	6
FCANet (3)	84.63	4.2	82.44	4.2	65.99	4.2
FCANet (5)	**88.00**	6	**86.63**	6	**81.38**	6
Ours	84.29	5.96	84.53	8.74	77.28	7.02

TABLE III: Average IoU and time over images, except for Rooftop, where time is computed over instances. For robot user experiments, with multiple budgets (3 and 5 clicks), time was estimated according to this study [51]. Our method obtains comparable accuracy, but it requires additional time to annotate foreground and background.

procedures. Despite this, existing methods require less time to annotate these datasets; this is due to them being specialized in the foreground annotation microtask, while our approach wastes time annotating the background — this is exacerbated on the DAVIS dataset where a background object might be of the same nature as the foreground.

The following experiment evaluated our performance on the semantic segmentation dataset Cityscapes [52], where labeling the whole image is the final goal, not just the microtask of delineating a single object. Since the true labels of the test set are not available, we took the same approach as [12], by testing on the validation set. Furthermore, the annotation quality was evaluated on 98 randomly chosen images (about 20% of the validation set). The boundary prediction network was optimized on the training set boundaries.

The original article reports an agreement (*i.e.* accuracy) between annotators of 96%. We obtained an agreement of 91.5% with the true labels of the validation set (Figure 5), while spending less than 1.5% of their time — *i.e.* our experiment took 1 hour and 58 minutes to annotate the 98 images, while to produce the same amount of ground-truth data took approximately 6.1 days (average of 1.5 hour per image [52]) — about 74.75 times faster than the original procedure. These 98 images contain about 6500 segmentation instances. Thus, with the estimate of 6 secs per instance, FCANet would take 10 hours and 50 minutes to label them.

Additional results and a study of individual parts of the pipeline can be found at [22].

V. Conclusion and Reflections

In this work, we studied a diverse set of interactive image segmentation methodologies, following along the progress of this research area, starting from graph-based methods to the recent deep learning-based techniques, and proposing a novel alternative for large-scale annotation.

The two first contributions push the boundaries of image segmentation by providing methodologies that assist the user in obtaining higher-quality segmentation labels. Moreover, the accomplishments of DT (Section II) have yet to be explored in the context of semi-supervised learning on the feature domain.

Additionally, we noticed that ourselves and a large portion of the research community were caught up in existing benchmarks or established procedures for solving the proposed problem. Given that, we took a step back and noticed that scaling existing approaches to the magnitude of existing datasets is a significant and challenging problem.

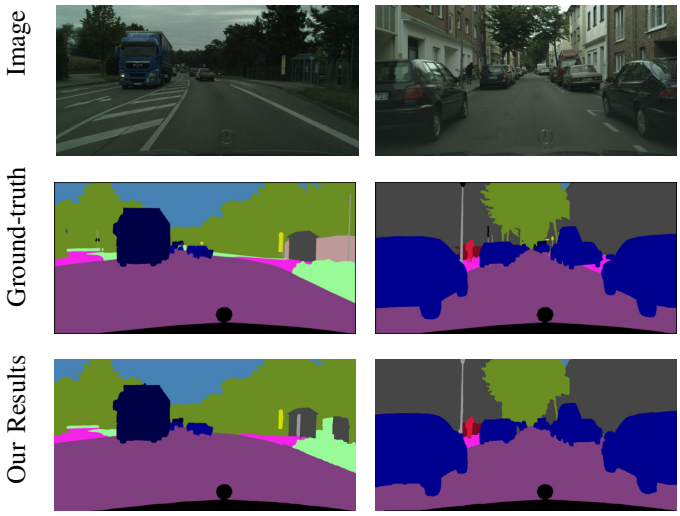

Fig. 5: Cityscapes result, each column is a different image, row indicates which kind.

To tackle that, we proposed a new framework for annotating multiple segments at once, Section IV. Our implementation showed comparable results to existing deep learning techniques for foreground and background annotation and significantly reduced the annotation time of data for semantic segmentation tasks, where samples belong to the same context, at a small cost over the final accuracy.

We think this latter method is a starting point for novel methodologies for annotating segments at scale. Notably, active learning could be inserted into the system, predicting the classes of segments where the labeling is trivial and recommending annotation of samples with high classification uncertainty, improving the classifier at each interaction. Additionally, the approach could be explored in scenarios with simultaneous annotation from multiple users.

By making all of our code available we hope to assist the advancement of ML research and its applications, especially in the domains where labeled data is lacking or expensive.

VI. Acknowledgments

This work was supported by CNPq [#303808/2018-7]; and FAPESP research grants [#2014/12236-1, #2019/11349-0, and #2019/21734-9]; We thank Chan Zuckerberg Biohub for assisting with the conference expenses.

VII. Publications

The author's dissertation contains the following articles: [14], [21], [22]. Other works [53], [54] were also developed in parallel but were not part of the dissertation.

REFERENCES

[1] Y. LeCun, L. Bottou, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] A. Krizhevsky, I. Sutskever, *et al.*, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[3] K. He, X. Zhang, *et al.*, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016, pp. 770–778.

[4] J. Deng, W. Dong, *et al.*, "Imagenet: A large-scale hierarchical image database," in *IEEE CVPR*, IEEE, 2009, pp. 248–255.

[5] A. Gupta, P. Dollar, *et al.*, "Lvis: A dataset for large vocabulary instance segmentation," in *IEEE CVPR*, 2019, pp. 5356–5364.

[6] M. Everingham, L. Van Gool, *et al.*, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[7] F. Perazzi, J. Pont-Tuset, *et al.*, "A benchmark dataset and evaluation methodology for video object segmentation," in *IEEE CVPR*, 2016, pp. 724–732.

[8] A. X. Falcão, J. K. Udupa, *et al.*, "User-steered image segmentation paradigms: Live wire and live lane," *Graphical models and image processing*, vol. 60, no. 4, pp. 233–260, 1998.

[9] A. X. Falcão, J. Stolfi, *et al.*, "The image foresting transform: Theory, algorithms, and applications," *IEEE TPAMI*, vol. 26, no. 1, pp. 19–29, 2004.

[10] C. Rother, V. Kolmogorov, *et al.*, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Trans. on Graphics*, vol. 23, 2004, pp. 309–314.

[11] C. Couprie, L. Grady, *et al.*, "Power watershed: A unifying graph-based optimization framework," *IEEE TPAMI*, vol. 33, no. 7, pp. 1384–1399, 2011.

[12] L. Castrejon, K. Kundu, *et al.*, "Annotating object instances with a polygon-rnn," in *IEEE CVPR*, 2017, pp. 5230–5238.

[13] N. Xu, B. Price, *et al.*, "Deep interactive object selection," in *IEEE CVPR*, 2016, pp. 373–381.

[14] J. Bragantini, S. B. Martins, *et al.*, "Graph-based image segmentation using dynamic trees," in *Iberoamerican Congress on Pattern Recognition*, Springer, 2018, pp. 470–478.

[15] Z. Li, Q. Chen, *et al.*, "Interactive image segmentation with latent diversity," in *IEEE CVPR*, 2018, pp. 577–585.

[16] W.-D. Jang and C.-S. Kim, "Interactive image segmentation via backpropagating refinement scheme," in *IEEE CVPR*, 2019, pp. 5297–5306.

[17] K. Sofiiuk, I. Petrov, *et al.*, "F-brs: Rethinking backpropagating refinement for interactive segmentation," in *IEEE CVPR*, 2020, pp. 8623–8632.

[18] T. Kontogianni, M. Gygli, *et al.*, "Continuous adaptation for interactive object segmentation by learning from corrections," in *IEEE ECCV*, Springer, 2020, pp. 579–596.

[19] S. Zhang, J. H. Liew, *et al.*, "Interactive object segmentation with inside-outside guidance," in *IEEE CVPR*, 2020, pp. 12 234–12 244.

[20] Z. Lin, Z. Zhang, *et al.*, "Interactive image segmentation with first click attention," in *IEEE CVPR*, 2020, pp. 13 339–13 348.

[21] J. Bragantini, B. Moura, *et al.*, "Grabber: A tool to improve convergence in interactive image segmentation," *Pattern Recognition Letters*, vol. 140, pp. 267–273, 2020.

[22] J. Bragantini, A. Falcão, *et al.*, "Rethinking interactive image segmentation: Feature space annotation," *Pattern Recognition*, 2022.

[23] N. Sofroniew, T. Lambert, *et al.*, *Napari/napari: 0.4.12rc2*, version v0.4.12rc2, Oct. 2021. DOI: 10.5281/zenodo.5587893. [Online]. Available: https://doi.org/10.5281/zenodo.5587893.

[24] J. Bragantini, "Interactive image segmentation: From graph-based algorithms to feature-space annotation," M.S. thesis, Universidade Estadual de Campinas, Instituto de Computação, 2021.

[25] C. Allène, J.-Y. Audibert, *et al.*, "Some links between extremum spanning forests, watersheds and min-cuts," *Image and Vision Computing*, vol. 28, no. 10, pp. 1460–1471, 2010.

[26] K. C. Ciesielski and et al., "Path-value functions for which dijkstra's algorithm returns optimal mapping," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 7, pp. 1–12, 2018.

[27] W. P. Amorim, A. X. Falcão, *et al.*, "Improving semi-supervised learning through optimum connectivity," *Pattern Recognition*, vol. 60, pp. 72–85, 2016.

[28] Y. Boykov, O. Veksler, *et al.*, "Fast approximate energy minimization via graph cuts," *IEEE TPAMI*, vol. 23, no. 11, pp. 1222–1239, 2001.

[29] L. Grady, "Random walks for image segmentation," *IEEE TPAMI*, vol. 28, no. 11, pp. 1768–1783, 2006.

[30] J. Cousty, G. Bertrand, *et al.*, "Watershed cuts: Thinnings, shortest path forests, and topological watersheds," *IEEE TPAMI*, vol. 32, no. 5, pp. 925–939, 2010.

[31] M. Tang, L. Gorelick, *et al.*, "Grabcut in one cut," in *IEEE CVPR*, 2013, pp. 1769–1776.

[32] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE TPAMI*, vol. 26, no. 9, pp. 1124–1137, 2004.

[33] P. A. V. Miranda and L. A. C. Mansilla, "Oriented image foresting transform segmentation by seed competition," *IEEE TIP*, vol. 23, no. 1, pp. 389–398, 2014.

[34] C. L. Demario and P. A. Miranda, "Relaxed oriented image foresting transform for seeded image segmentation," in *IEEE ICIP*, IEEE, 2019, pp. 1520–1524.

[35] A. X. Falcão and F. P. G. Bergo, "Interactive volume segmentation with differential image foresting transforms," *IEEE Transactions on Medical Imaging*, vol. 23, no. 9, pp. 1100–1108, 2004.

[36] J. Cousty, G. Bertrand, *et al.*, "Watershed cuts: Minimum spanning forests and the drop of water principle," *IEEE TPAMI*, vol. 31, no. 8, pp. 1362–1374, 2009.

[37] W. Casaca, J. P. Gois, *et al.*, "Laplacian coordinates: Theory and methods for seeded image segmentation," *IEEE TPAMI*, 2020.

[38] F. Andrade and E. V. Carrera, "Supervised evaluation of seed-based interactive image segmentation algorithms," in *Sym. on Signal Processing, Images and Computer Vision*, 2015, pp. 1–7.

[39] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the Int. Journal for Geographic Info. and Geovisualization*, vol. 10, pp. 112–122, 1973.

[40] D. Martin, C. Fowlkes, *et al.*, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE ICCV*, IEEE, vol. 2, 2001, pp. 416–423.

[41] K. McGuinness and N. E. O'connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.

[42] R. Benenson, S. Popov, *et al.*, "Large-scale interactive object segmentation with human annotators," in *IEEE CVPR*, 2019, pp. 11 700–11 709.

[43] D. Acuna, H. Ling, *et al.*, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *IEEE CVPR*, 2018, pp. 859–868.

[44] H. Ling, J. Gao, *et al.*, "Fast interactive object annotation with curve-gcn," in *IEEE CVPR*, 2019, pp. 5257–5266.

[45] J.-J. Liu, Q. Hou, *et al.*, "A simple pooling-based design for real-time salient object detection," in *IEEE CVPR*, 2019, pp. 3917–3926.

[46] J. Cousty, L. Najman, *et al.*, "Hierarchical segmentations with graphs: Quasi-flat zones, minimum spanning trees, and saliency maps," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 4, pp. 479–502, 2018.

[47] L. McInnes, J. Healy, *et al.*, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[48] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification.," *JMLR*, vol. 10, no. 2, 2009.

[49] D. Batra, A. Kowdle, *et al.*, "Interactively co-segmentating topically related images with intelligent scribble guidance," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 273–292, 2011.

[50] X. Sun, C. M. Christoudias, *et al.*, "Free-shape polygonal object localization," in *IEEE ECCV*, Springer, 2014, pp. 317–332.

[51] A. Bearman, O. Russakovsky, *et al.*, "What's the point: Semantic segmentation with point supervision," in *IEEE ECCV*, Springer, 2016, pp. 549–565.

[52] M. Cordts, M. Omran, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *IEEE CVPR*, 2016, pp. 3213–3223.

[53] A. X. Falcão and J. Bragantini, "The role of optimum connectivity in image segmentation: Can the algorithm learn object information during the process?" In *Int. Conf. on Discrete Geometry for Computer Imagery*, 2019, pp. 180–194.

[54] S. B. Martins, T. V. Spina, *et al.*, "A multi-object statistical atlas adaptive for deformable registration errors in anomalous medical image segmentation," in *SPIE on Medical Imaging: Image Processing*, 2017, 101332G–101332G.