doi:10.1017/S1431927620018243
Microsc. Microanal. 26 (Suppl 2), 2020
© Microscopy Society of America 2020

Component Detection and Evaluation Framework (CDEF): A Semantic Annotation Tool

Nathan Jessurun[1], Olivia Paradis[1], Alexandra Roberts[2] and Navid Asadizanjani[1]

[1]University of Florida, Gainesville, Florida, United States, [2]University of Florida, Bradenton, Florida, United States

Abstract

Image labeling is the process of manually assigning a class to subregions within an image for machine learning applications. When these subregions are complex shapes, this process is known as semantic segmentation. We propose a new software application, the Component Detection and Evaluation Framework (CDEF), for creating such semantic labels. The benefits of CDEF over existing tools are highlighted, and further improvements are proposed.

Introduction

In multiple domains of image processing and computer vision, labeled image data is essential for tuning and evaluating the performance of machine learning applications. Such labels are typically provided as enclosing shapes, i.e. rectangles, circles, or simple polygons. While this streamlines the labeling process, it misrepresents more complex components. When high accuracy is required, labels must be specified at the pixel-level – a process known as *segmentation labeling* or *semantic segmentation.* A detailed description of this process is explained in [1].

A wide variety of applications require segmentation-level accuracy, ranging from hardware assurance to medical imaging. Examples include Bill-of-Material (BoM) extraction [2], quality control during manufacturing [3]–[6], manuscript restoration / digitization [7]–[11], and effective patient diagnosis [12]–[15]. In all these cases, imprecise annotations severely limit the development of automated solutions.

Unfortunately, semantic segmentation suffers from a heavy tradeoff between accuracy and labeling rate. More so than simple shape annotation, pixel-level annotation typically requires consensus among multiple human annotators. In cases where accuracy is paramount, freehand boundary drawing rapidly becomes time-prohibitive without a large manual workforce and a set of objective segmentation criteria. These drawbacks drastically decrease throughput and raise the cost associated creating a large dataset to train and test a robust machine learning algorithm.

While a host of software solutions address the need for bounding box annotation, few are designed to assist with pixel-level annotation. Moreover, products fitting this domain often lack features conducive toward a streamlined workflow. For this paper, several products were tested on their ability to create pixel-level boundaries around surface-mount components on a PCB (e.g. resistors, capacitors, etc). The evaluated tools either could not easily process high-resolution images, suffered degraded performance when hundreds of distinct bounding boxes were drawn on the same image, or kept crucial annotation features behind an enterprise paywall. Such software solutions can be found at [16].

https://doi.org/10.1017/S1431927620018243 Published online by Cambridge University Press

To optimize the accuracy/labeling speed tradeoff, we propose the Component Detection and Evaluation Framework (CDEF). This tool is designed with ease, extensibility, and efficiency in mind. Its graphical user interface (GUI) is shown in Figure 1.

Component Detection and Evaluation Framework (CDEF) Contributions

Customizable interface. Major annotation functions are accessible through a configurable list of shortcuts and keystroke macros. Moreover, users can easily add custom functions to this list by utilizing the CDEF application programming interface (API). This contrasts with most other annotation software, where functionality is not user extendable. Importantly, these features easily accommodate left-handed users or those with international keyboard layouts.

Through its configurable interface, CDEF provides time-saving utilities during manual annotation – directly reducing the cost of segmentation labeling.

High reconfigurability. CDEF is intended as a general-purpose tool for various applications which require semantic segmentation of large images containing many objects. Toward this end, the CDEF API enables users to easily incorporate their own algorithms for preprocessing, feature extraction, object detection, etc. This allows users to adapt CDEF to suit their needs for various, highly specialized purposes.

A demonstration of CDEF's adaptability is shown in Fig. 1, where CDEF is used to annotate PCB components.

Online parameter tuning. Most notably, the annotation tool provides an editor interface for all significant algorithm parameters, shown in Figure 2. This eliminates the need to reload the application each time a parameter is adjusted; instead, real-time performance evaluation is achievable as these values are altered. Users inserting their own algorithms can also specify which parameters they wish to be editable from the GUI.

Contour control caters well to situations where localized image variations prevent a single set of algorithm parameters from applying to the entire image.

Future Work

With the application still under active development, multiple features will be incorporated in the coming months. Some significant improvements are listed below.

Improved annotation intelligence. Object with the same classification (e.g. resistors in Figure 1) often share similar characteristics, but CDEF does not currently account for this. When users edit the boundary of one component, the application should apply similar edits to other objects of the same type. This decreases the required amount of manual input. Also, the tool should account for contour shape to remove irregularities.

Intelligent contour refinement will increase the annotation accuracy without compromising speed or throughput.

https://doi.org/10.1017/S1431927620018243 Published online by Cambridge University Press

Built-in evaluation tools. Future CDEF iterations will include performance analysis tools to compare a given set of labels (manual and automatic). This evaluation suite will contain methods of boundary comparison, offering metrics for determining algorithm strengths and weaknesses. Examples include component-wise boundary comparison, pixel-wise segmentation comparison, and component statistics (e.g. shape, size and other characteristics per label).

These evaluation tools will enable robust algorithm analysis and assist users in choosing the optimal parameters for a given set of input images.

Future developments constitute further ways to reduce the manual labor involved in segmentation labeling. With a larger user base, the FICS lab can more effectively determine which developments will best improve the workflow. If readers are interested in beta testing, they can contact the lead author for more information.

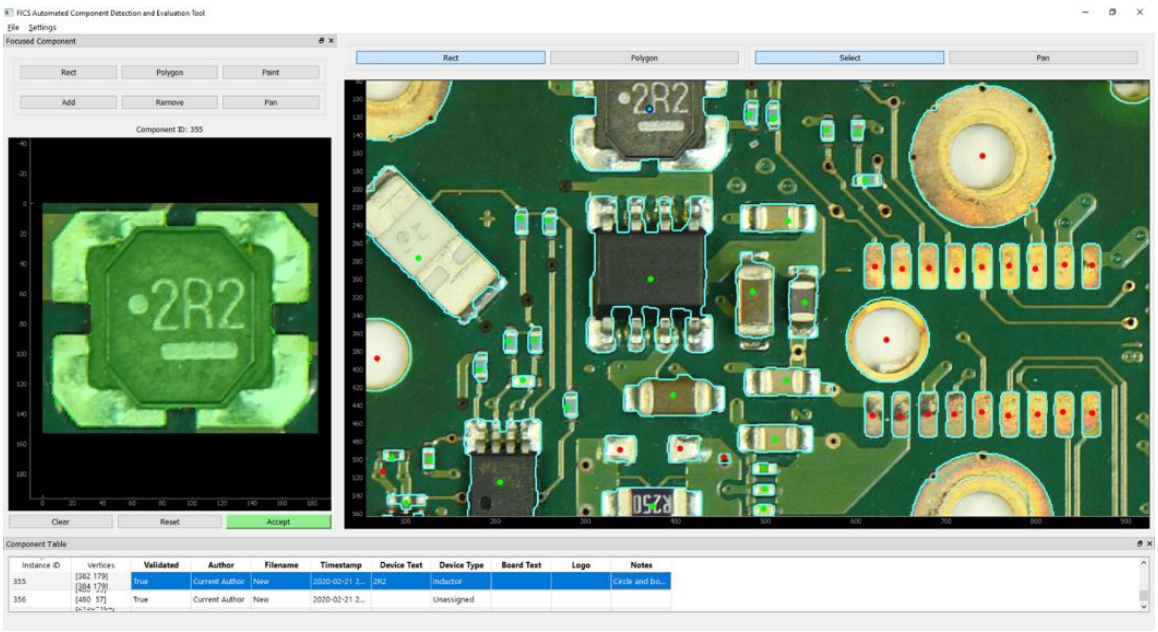

Figure 1. Overview of the annotation tool. Main UI components include an overview of the whole image, a zoomed-in view of the currently selected component, and a table of current component properties.

https://doi.org/10.1017/S1431927620018243 Published online by Cambridge University Press

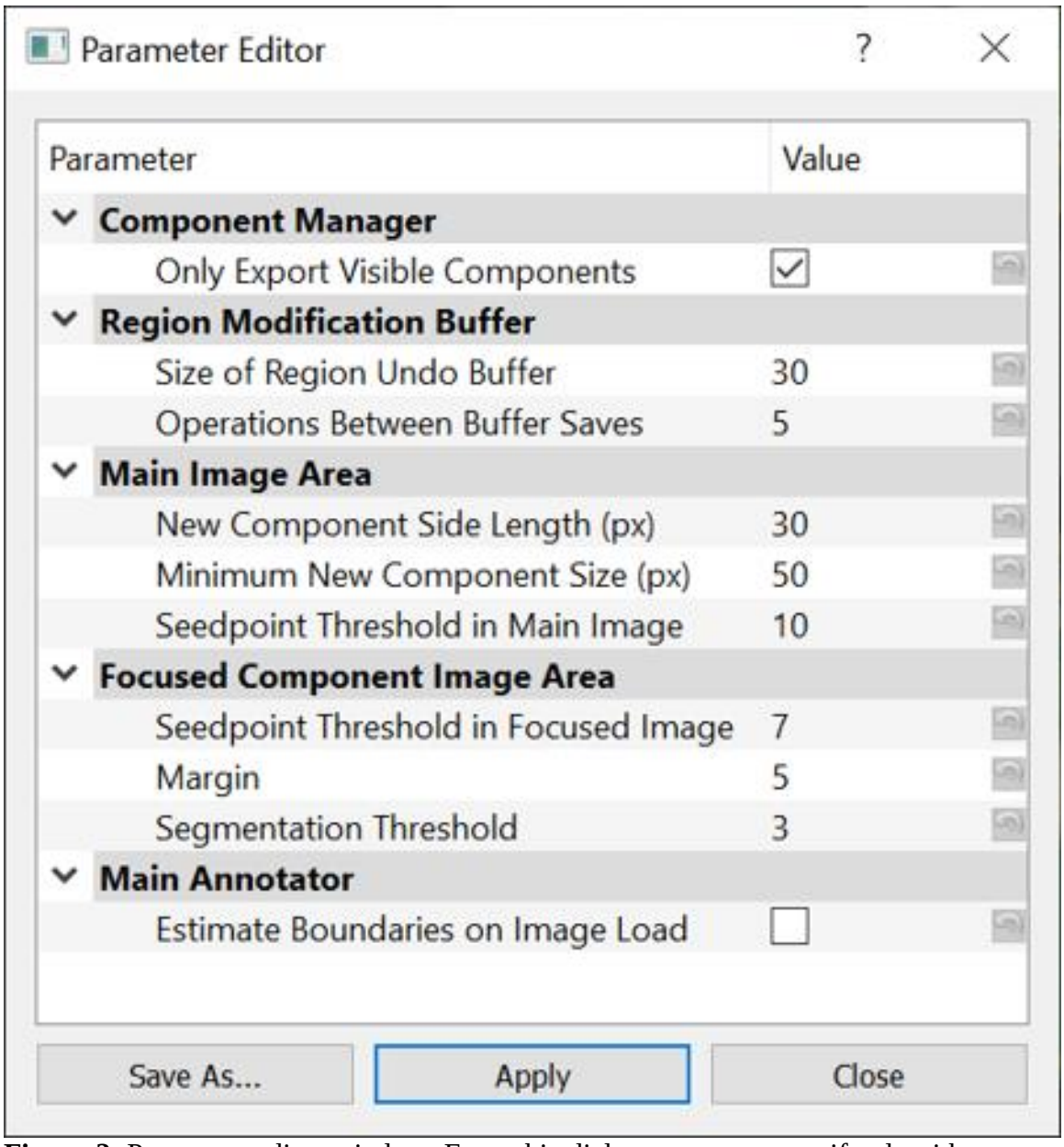

Figure 2. Parameter editor window. From this dialog, users can specify algorithm parameters and save their states on-the-fly.

References

[1] A. Hanbury, *J. Vis. Lang. Comput.*, vol. 19, no. 5, pp. 617–627, Oct. 2008

[2] M. Azhagan *et al.*, *in ISTFA 2019: Proceedings of the 45th International Symposium for Testing and Failure Analysis*, 2019, p. 256.

[3] M. K. Ferguson, A. Ronay, Y.-T. T. Lee, and Kincho. H. Law, *Smart Sustain. Manuf. Syst.*, vol. 2, 2018

[4] C. Anagnostopoulos, D. Vergados, E. Kayafas, V. Loumos, and G. Stassinopoulos, *J. Vis. Comput. Animat.*, vol. 12, no. 1, pp. 31–44, 2001

[5] C. Anagnostopoulos *et al.*, *Math. Comput. Simul.*, vol. 60, no. 3, pp. 389–400, Sep. 2002

[6] K. Adem, U. Orhan, and M. Hekim, *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3785–3789, May 2015

https://doi.org/10.1017/S1431927620018243 Published online by Cambridge University Press

[7] B. Gatos, K. Ntzios, I. Pratikakis, S. Petridis, T. Konidaris, and S. J. Perantonis, in *Document Analysis Systems VI*, Berlin, Heidelberg, 2004, pp. 63–74
[8] M. W. A. Kesiman, J.-C. Burie, and J.-M. Ogier, in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 325–330
[9] A. K. Jain and S. Bhattacharjee, *Mach. Vis. Appl.*, vol. 5, no. 3, pp. 169–184, Jun. 1992
[10] T. Taxt, P. J. Flynn, and A. K. Jain, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 12, pp. 1322–1329, Dec. 1989
[11] H. Fujisawa, Y. Nakano, and K. Kurino, *Proc. IEEE*, vol. 80, no. 7, pp. 1079–1092, Jul. 1992
[12] S. Seifert et al., Proc. SPIE vol. 7628, p. 762808, Mar. 2010
[13] M. Rajchl *et al.*, *IEEE Trans. Med. Imaging*, vol. 36, no. 2, pp. 674–683, Feb. 2017
[14] P. A. Yushkevich *et al.*, *NeuroImage*, vol. 31, no. 3, pp. 1116–1128, Jul. 2006
[15] D. K. Iakovidis, T. Goudas, C. Smailis, and I. Maglogiannis, *The Scientific World Journal*, 2014.
[16] https://hackernoon.com/the-best-image-annotation-platforms-for-computer-vision-an-honest-review-of-each-dac7f565fea

https://doi.org/10.1017/S1431927620018243 Published online by Cambridge University Press