TerraVerum

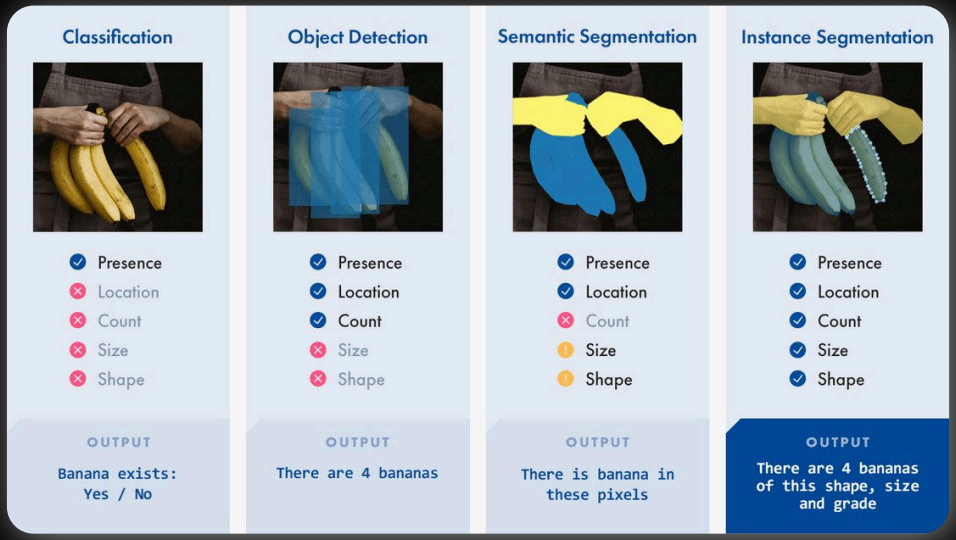
Annotationem ex Nihilo

Image annotation from scratch using `PyQtGraph` and `FastSAM`

Nathan Jessurun
Founder, TerraVerum

Introduction

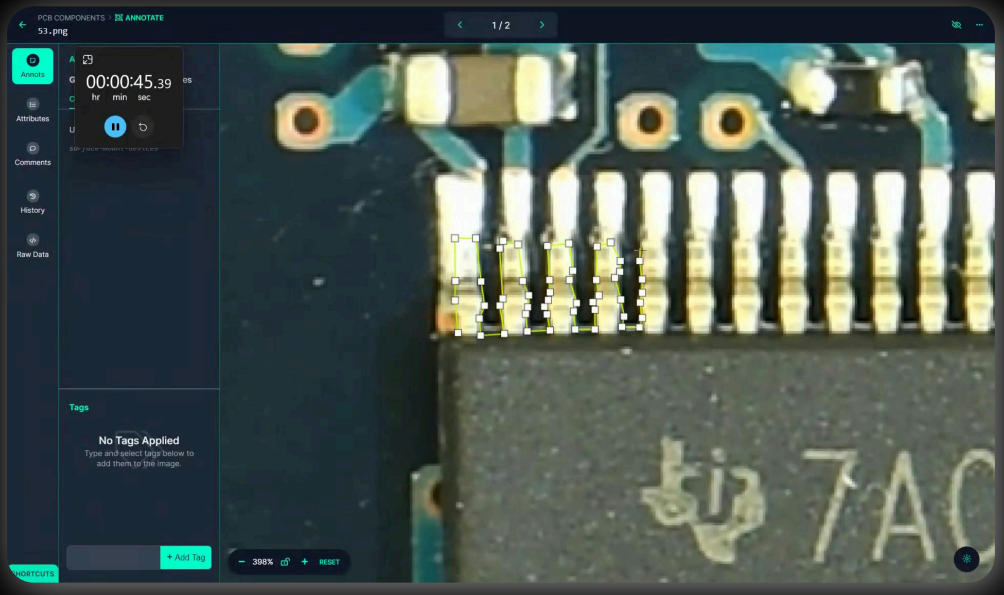
Several types of annotation, most time consuming and informative is **instance segmentation**



Introduction

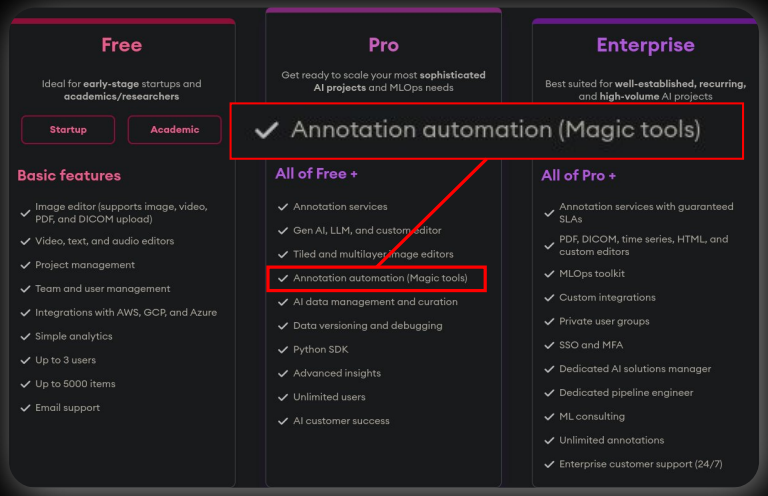
But ヽ(ಠ_ಠ)ノ **detailed annotation is time consuming!**

(manual timelapse)

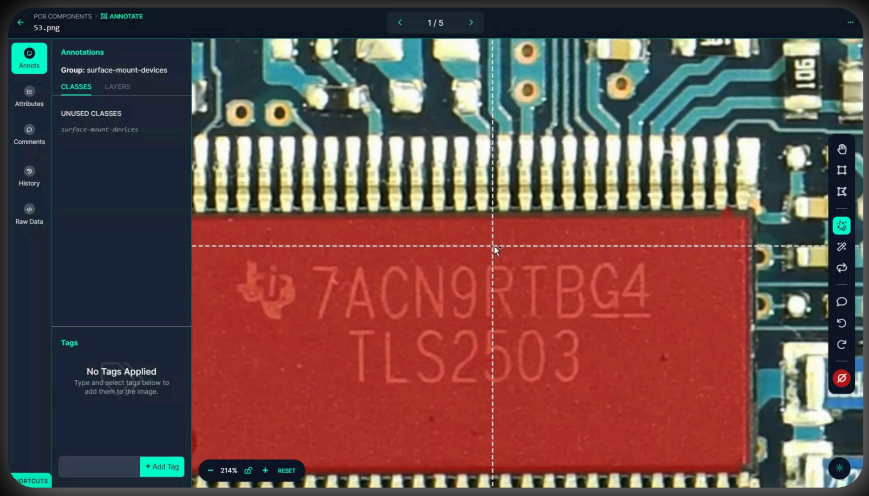


Introduction

Assistance tools help, but are often **paywalled**, **perform poorly**, or expect **detailed project workflows**



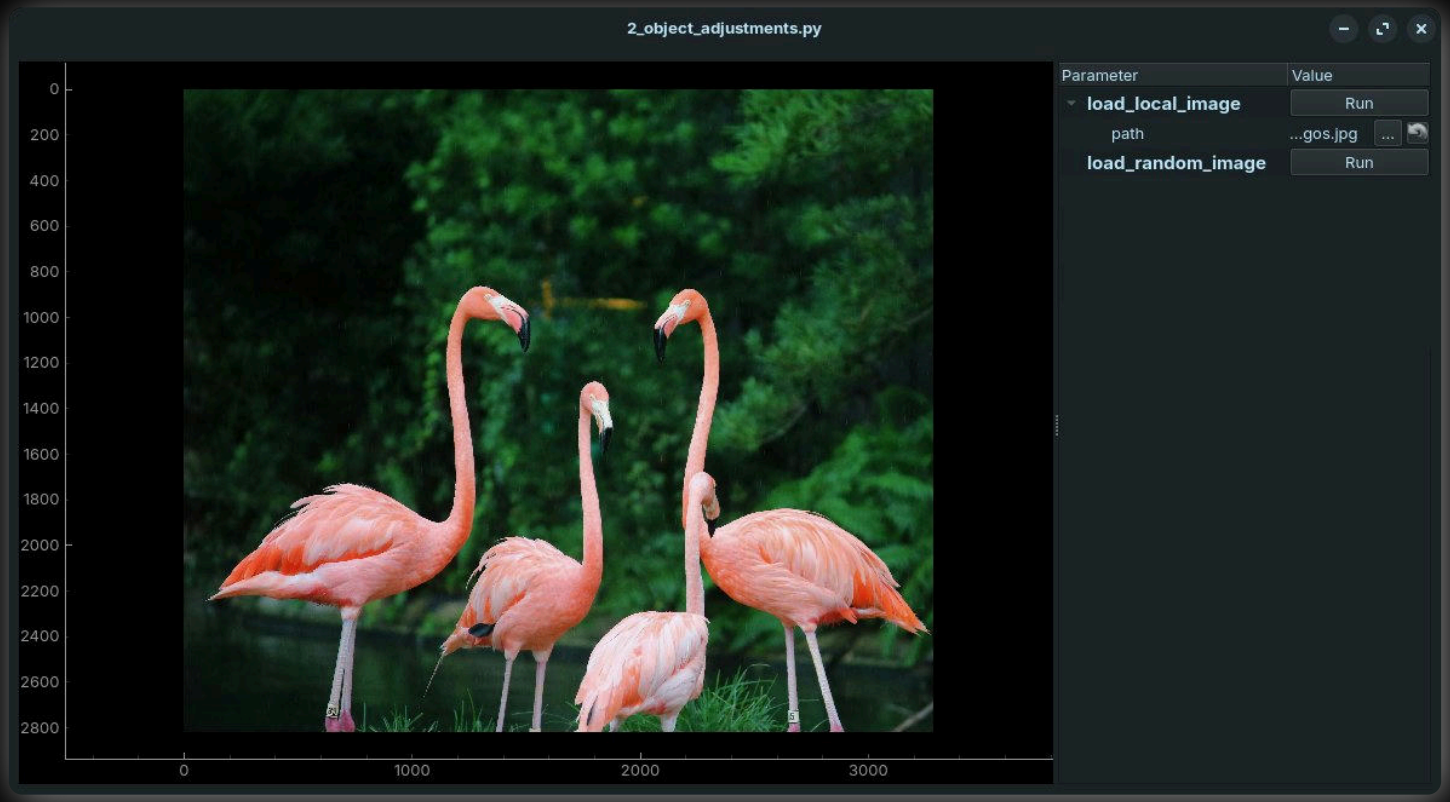
(Pretrained SAM)



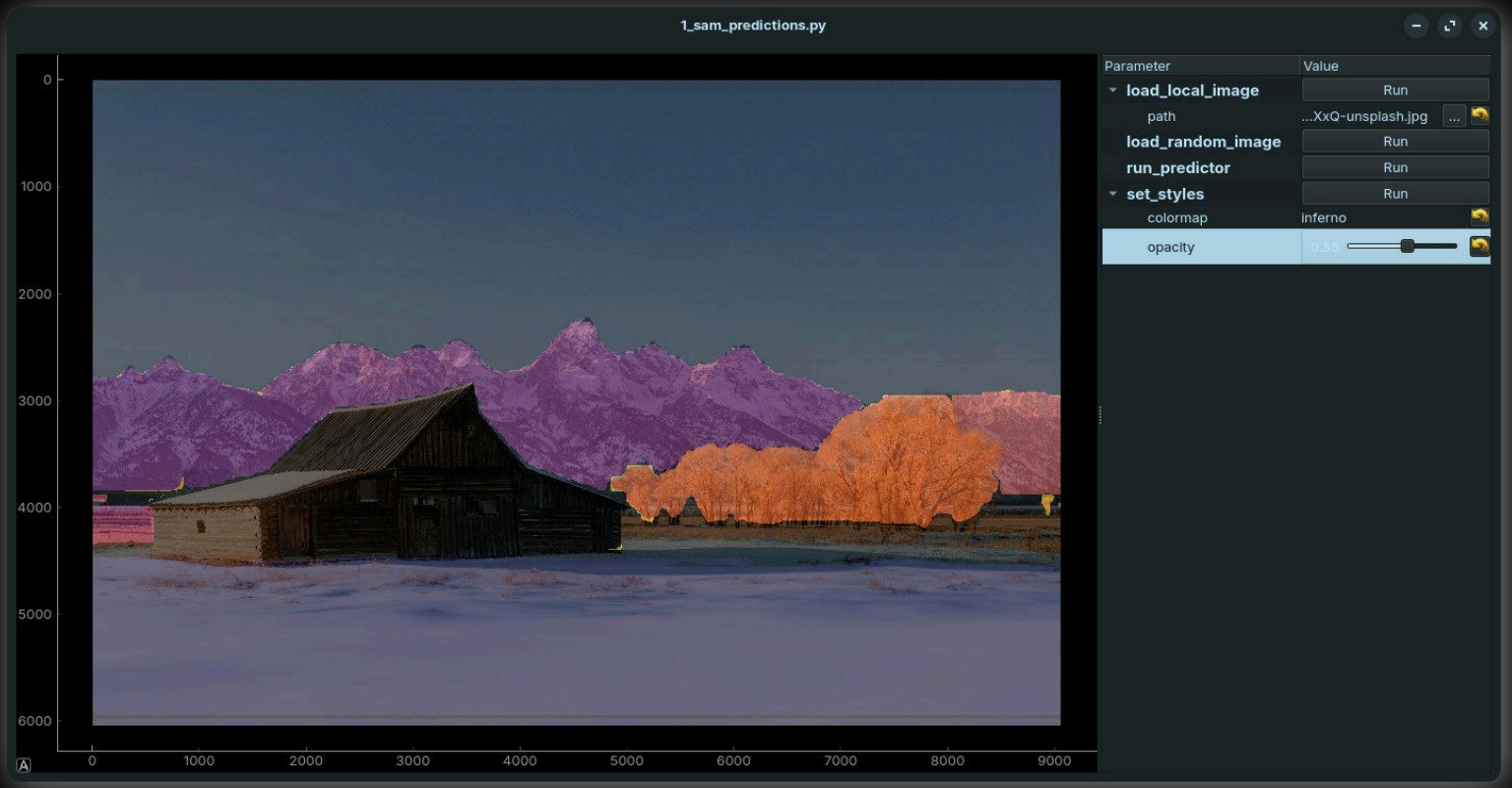
Tutorial Components

1. Create a window that loads an image from a file
2. Segment the current image using `FastSAM`
3. Enable object-by-object adjustments
4. Save annotations and edit history
5. Enable manual adjustments using a brush
6. Future work: incorporate metadata & postprocessing

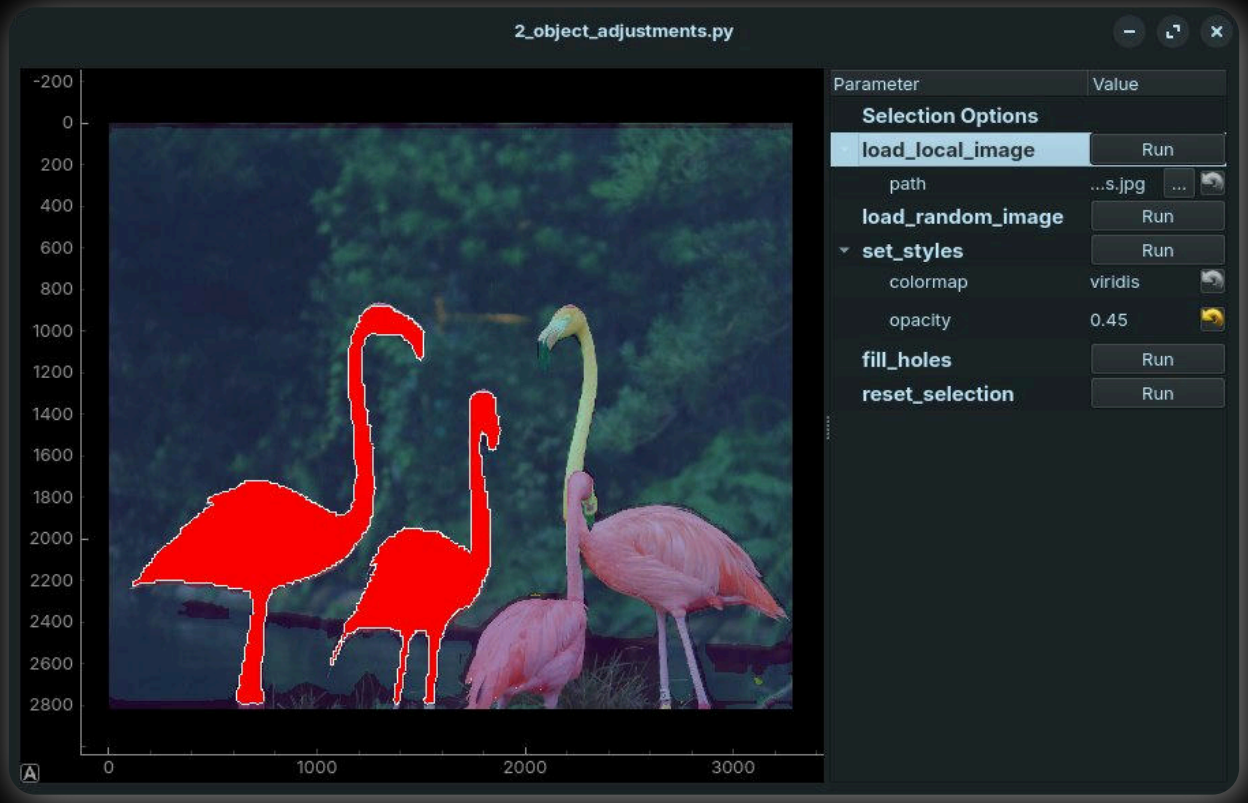
Window Loading an Image



Predictions using Ultralytics `FastSAM`



Enabling User-Specified Regions



Persisting User Edits

In progress...

Enabling Brush Adjustments

In progress...

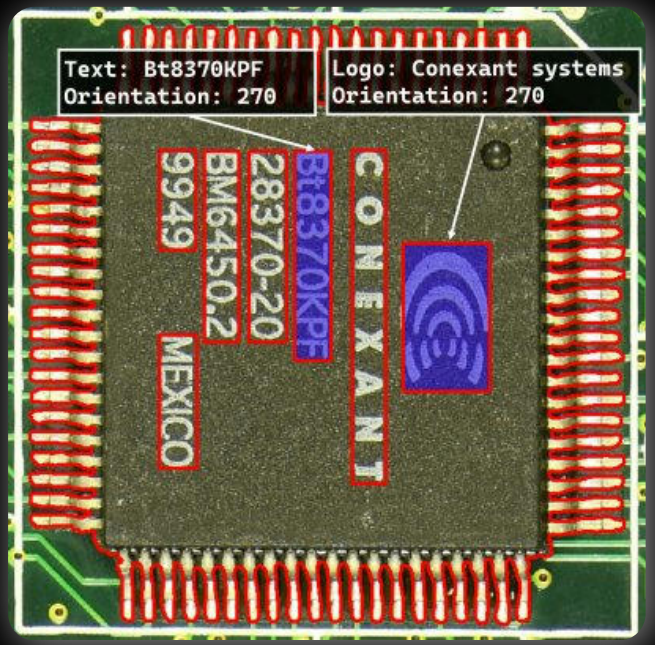
Future Work

Regions are only part of the story. We also need:

Metadata (class, tags, comments, ...)

Postprocessing

Annotation management



Future Work

Regions are only part of the story. We also need:

Metadata

Postprocessing (Filtering, grouping, ...)

Annotation management

Future Work

Regions are only part of the story. We also need:

Metadata

Postprocessing

Annotation management (copy/delete, bulk edits, ...)

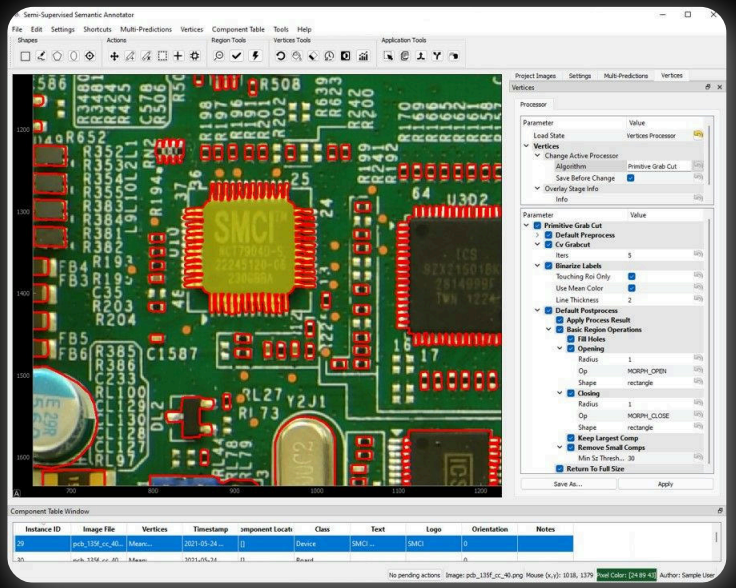
tor	Class	Text	Orientation	Notes
	Board	R1089	0	CL: Missing ...
	Board	R1110	0	
	Board	R1114	0	CL: Missing ...
	Board	R1112	0	
	Board	R1115	0	CL: Missing ...
	Board	C771	90	
	Board	C769	90	

Future Work

These are available within S3A: an open-source labeling tool

Integrate `FastSAM` (or any algorithm you wish) in **3* lines of code!**

```
def wrap_fast_sam(image: np.ndarray):
  label_mask = fast_sam(image)
  return dict(labels=label_mask)
```



Conclusion

In progress...