


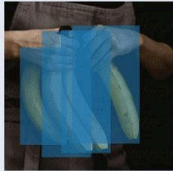
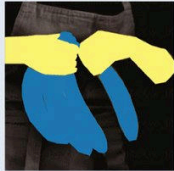

Annotationem ex Nihilo

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

Nathan Jessurun
Founder, TerraVerum

Introduction

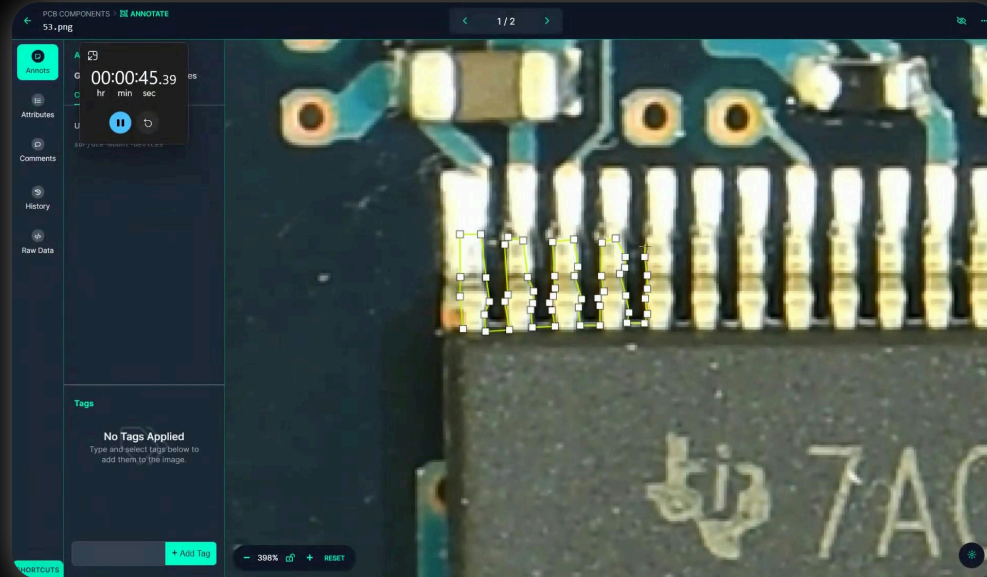
Several types of annotation, most time consuming and informative is **instance segmentation**

Classification	Object Detection	Semantic Segmentation	Instance Segmentation
			
<ul style="list-style-type: none">✓ Presence✗ Location✗ Count✗ Size✗ Shape	<ul style="list-style-type: none">✓ Presence✓ Location✓ Count✗ Size✗ Shape	<ul style="list-style-type: none">✓ Presence✓ Location✗ Count⚠ Size⚠ Shape	<ul style="list-style-type: none">✓ Presence✓ Location✓ Count✓ Size✓ Shape
<p>OUTPUT</p> <p>Banana exists: Yes / No</p>	<p>OUTPUT</p> <p>There are 4 bananas</p>	<p>OUTPUT</p> <p>There is banana in these pixels</p>	<p>OUTPUT</p> <p>There are 4 bananas of this shape, size and grade</p>

Introduction

But \rf_\rf_/ **detailed annotation is time consuming!**

(manual timelapse)

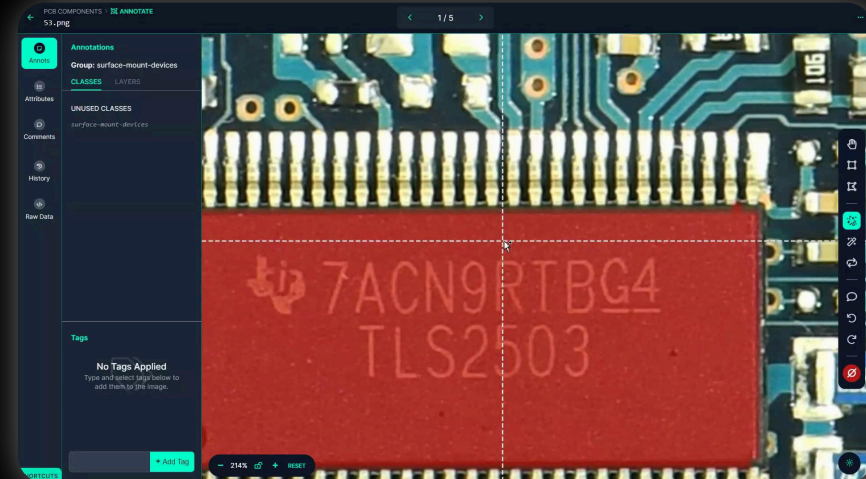


Introduction

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

(Pretrained SAM)

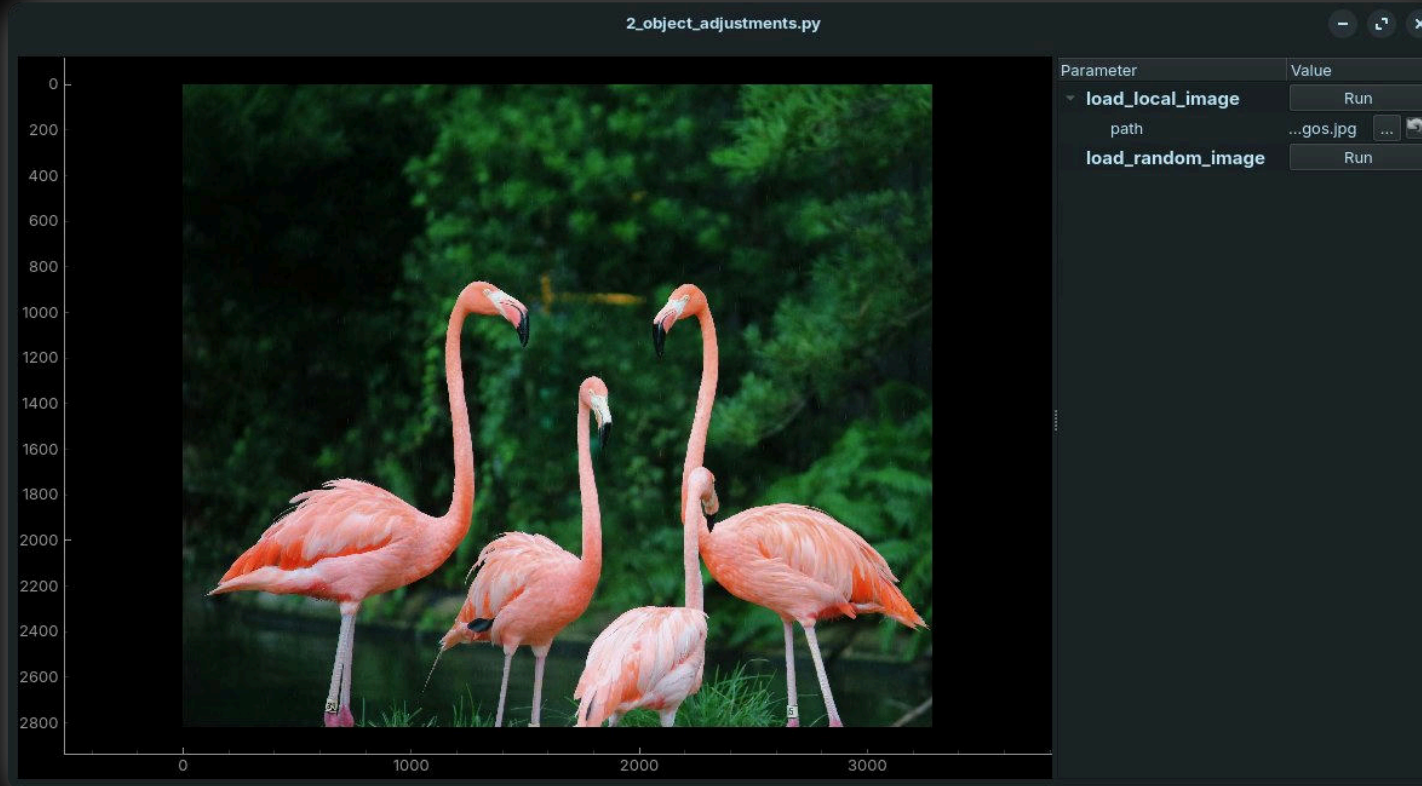
Free	Pro	Enterprise
Ideal for early-stage startups and academics/researchers	Get ready to scale your most sophisticated AI projects and MLOps needs	Best suited for well-established, recurring, and high-volume AI projects
Startup Academic	✓ Annotation automation (Magic tools)	
Basic features	All of Free +	All of Pro +
<ul style="list-style-type: none">✓ Image editor (supports image, video, PDF, and DICOM upload)✓ Video, text, and audio editors✓ Project management✓ Team and user management✓ Integrations with AWS, GCP, and Azure✓ Simple analytics✓ Up to 3 users✓ Up to 5000 items✓ Email support	<ul style="list-style-type: none">✓ Annotation services✓ Gen AI, LLM, and custom editor✓ Tiled and multilayer image editors✓ Annotation automation (Magic tools)✓ AI data management and curation✓ Data versioning and debugging✓ Python SDK✓ Advanced insights✓ Unlimited users✓ AI customer success	<ul style="list-style-type: none">✓ Annotation services with guaranteed SLAs✓ PDF, DICOM, time series, HTML, and custom editors✓ MLOps toolkit✓ Custom integrations✓ Private user groups✓ SSO and MFA✓ Dedicated AI solutions manager✓ Dedicated pipeline engineer✓ ML consulting✓ Unlimited annotations✓ Enterprise customer support (24/7)



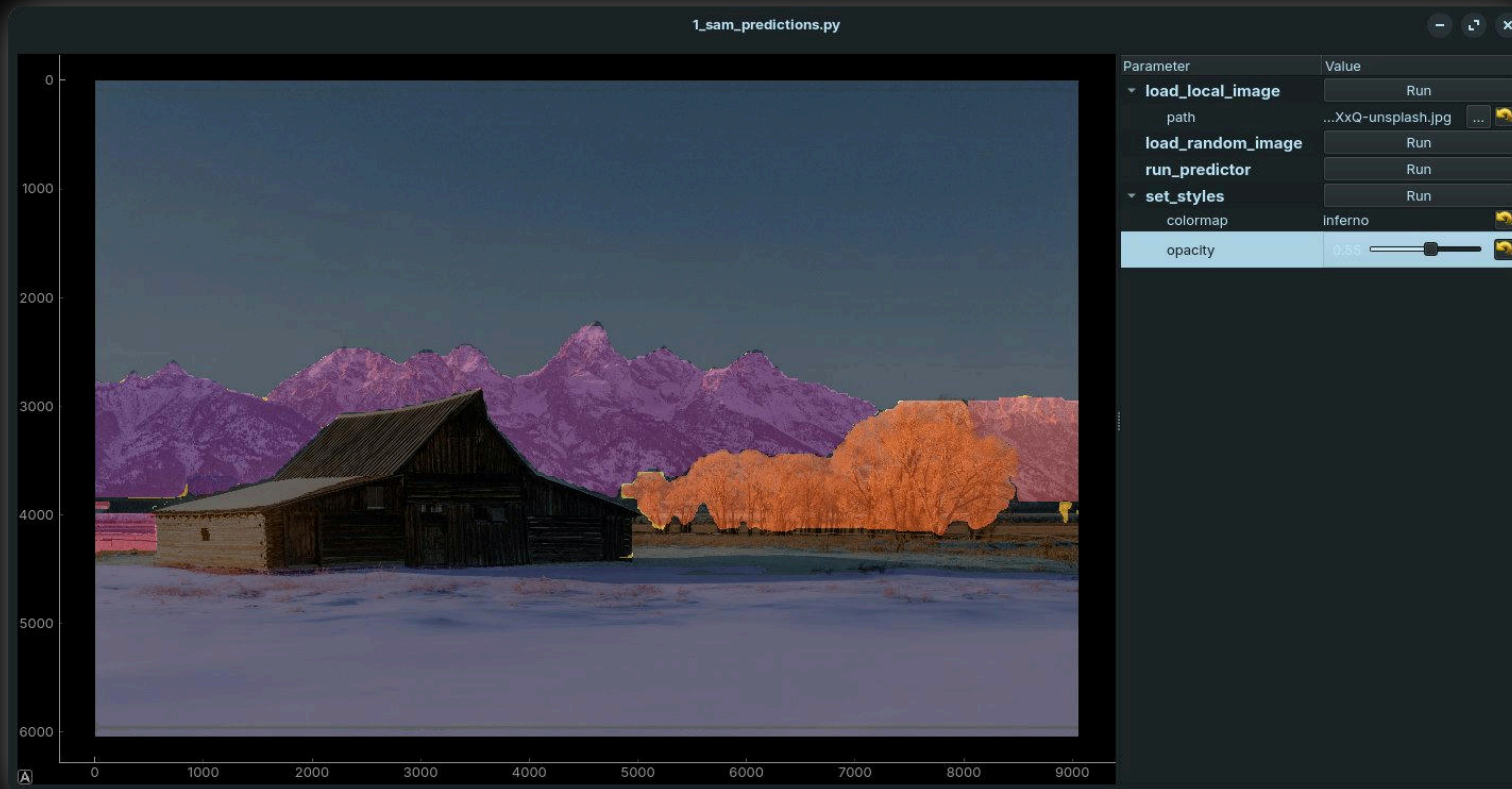
Tutorial Components

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

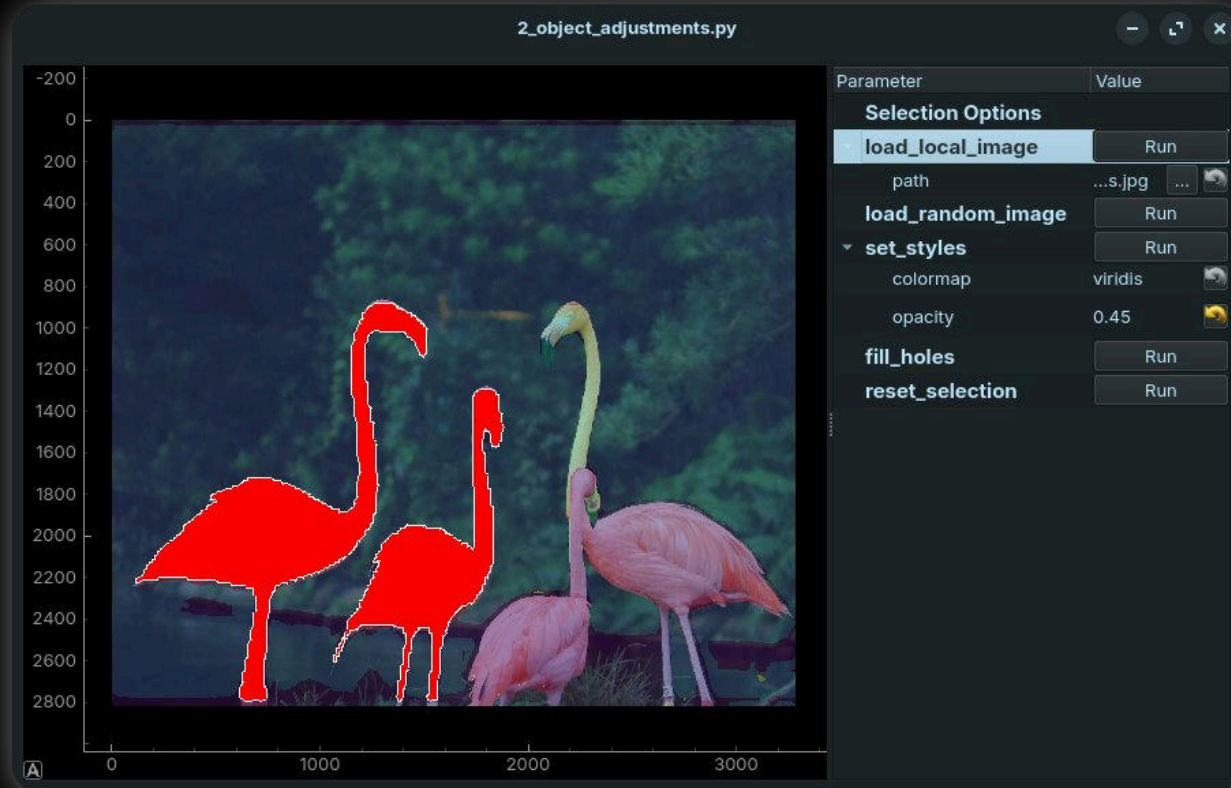
Window Loading an Image



Predictions using Ultralytics FastSAM



Enabling User-Specified Regions



Persisting User Edits

Enabling Brush Adjustments

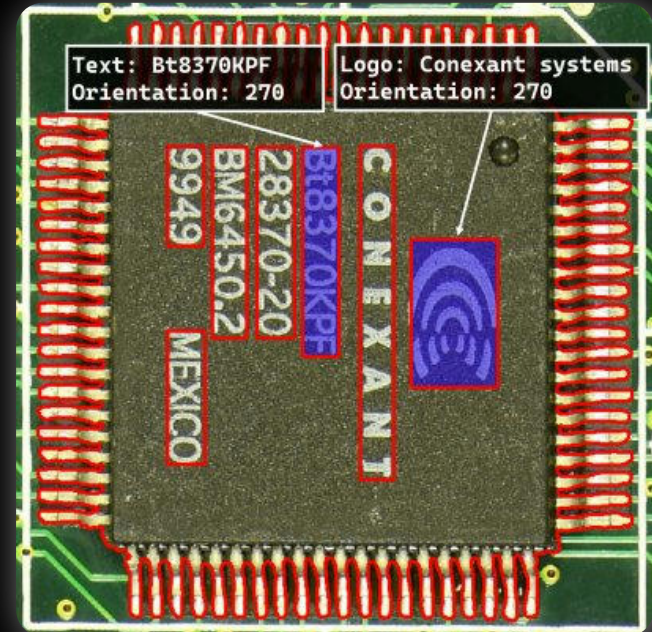
Future Work

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

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

Postprocessing

Annotation management



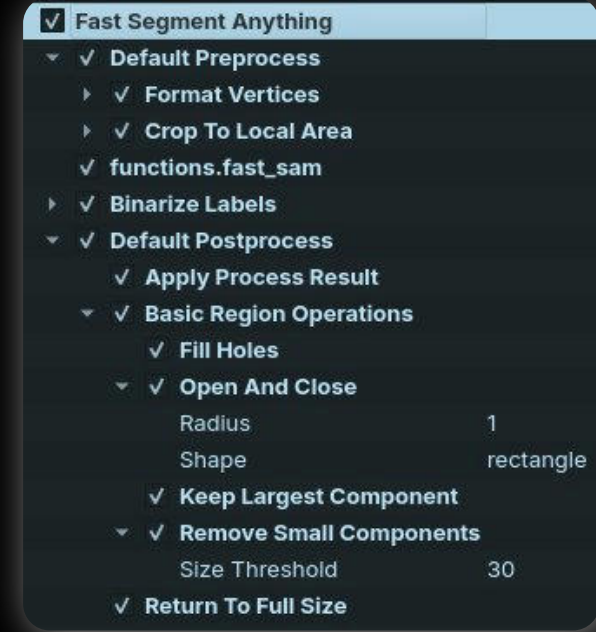
Future Work

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, ...)



The screenshot shows a top-down view of a green printed circuit board (PCB) layout. Several components are visible, each labeled with a blue box containing a text label: R1089, R1110, R1112, R1114, and R1115. The components are arranged in a grid-like pattern. Below the PCB image, there is a table with columns for 'tor', 'Class', 'Text', 'Orientation', and 'Notes'. The table contains data for the components shown in the image, including their class (Board), text labels (R1089, R1110, R1114, R1112, R1115), orientation (0 or 90), and notes (CL: Missing ...).

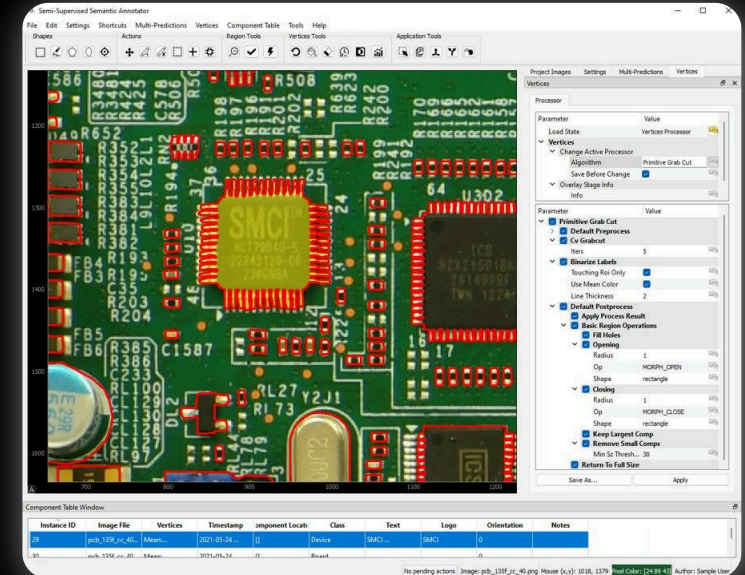
tor	Class	Text	Orientation	Notes
	Board	R1089	0	CL: Missing ...
	Board	R1110	0	
	Board	R1114	0	CL: Missing ...
	Board	R1112	0	
	Board	R1115	0	CL: Missing ...
	Board	C771	90	
	Board	C769	90	

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