Annolid: an instance segmentation-based multiple-animal tracking and behavior analysis package

Chen Yang, Matthew Einhorn, Thomas A. Cleland
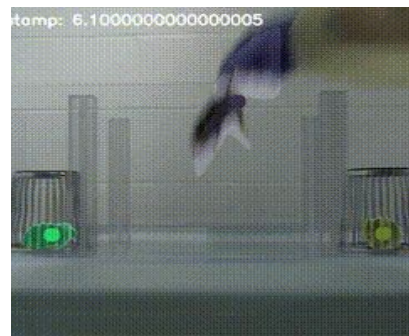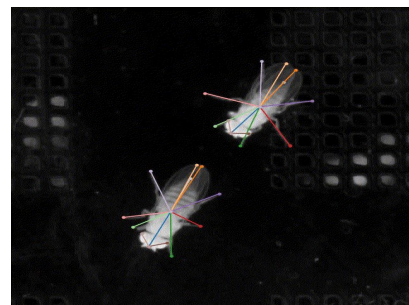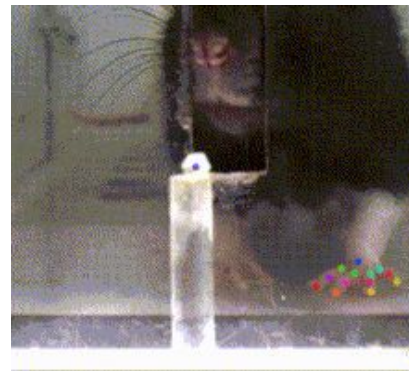Dept. of Psychology, Cornell University, Ithaca, NY 14853

Introduction

Background: Animal tracking with computer vision

1. DeepLabCut and SLEAP: animal pose estimation
2. SimBA (behavior recognition from pose)
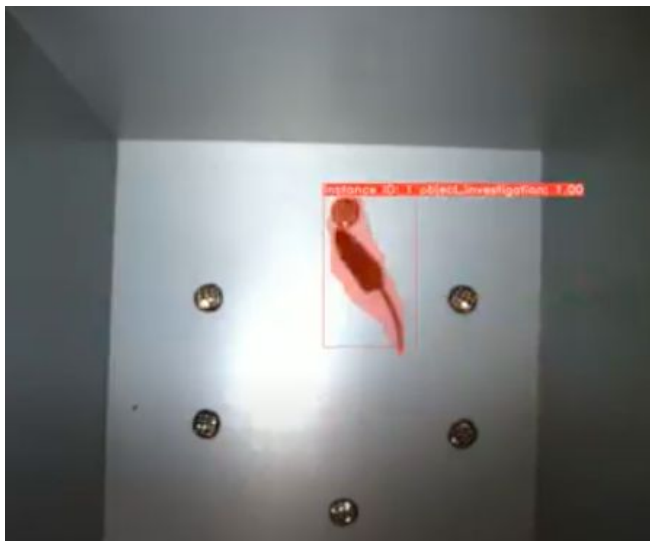3. Annolid: instance segmentation

https://github.com/DeepLabCut/DeepLabCut
https://github.com/murthylab/sleap

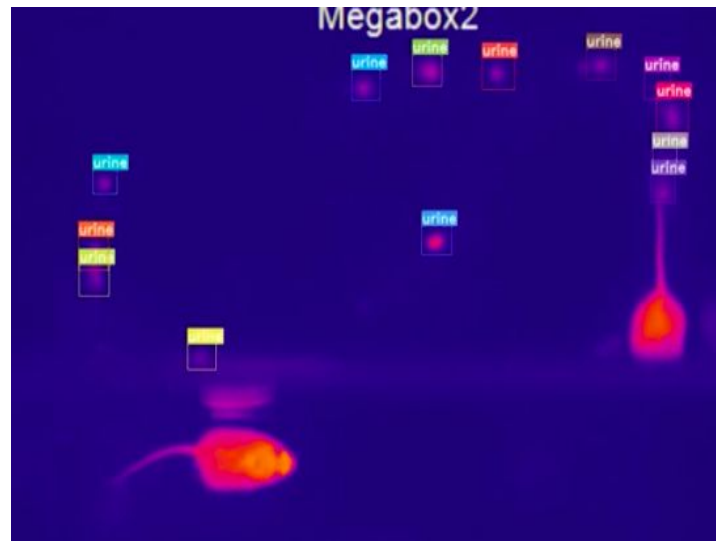# Why do we need instance segmentation?

Object Investigation

Area of ROIs

Annolid treats keypoints-based pose estimation as a special case of instance segmentation.

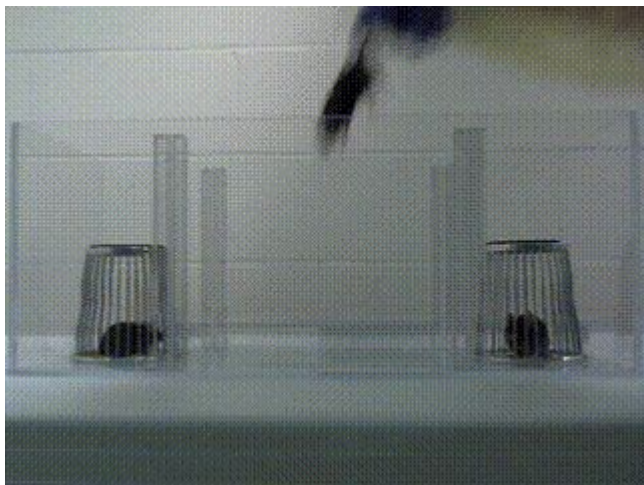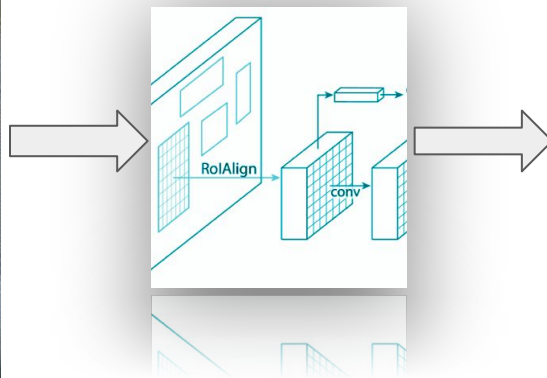Instance mask *area* and *perimeter* can be tracked, as well as *keypoints*.

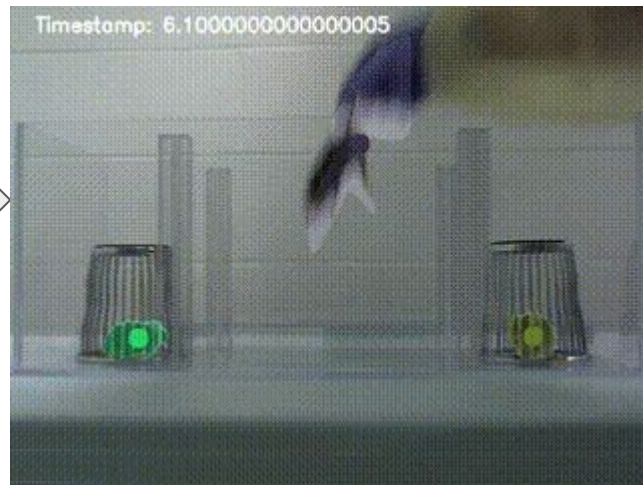# Annolid uses deep learning for animal tracking and behavior analysis

Annolid = Annotation + Annelid (segmentation)
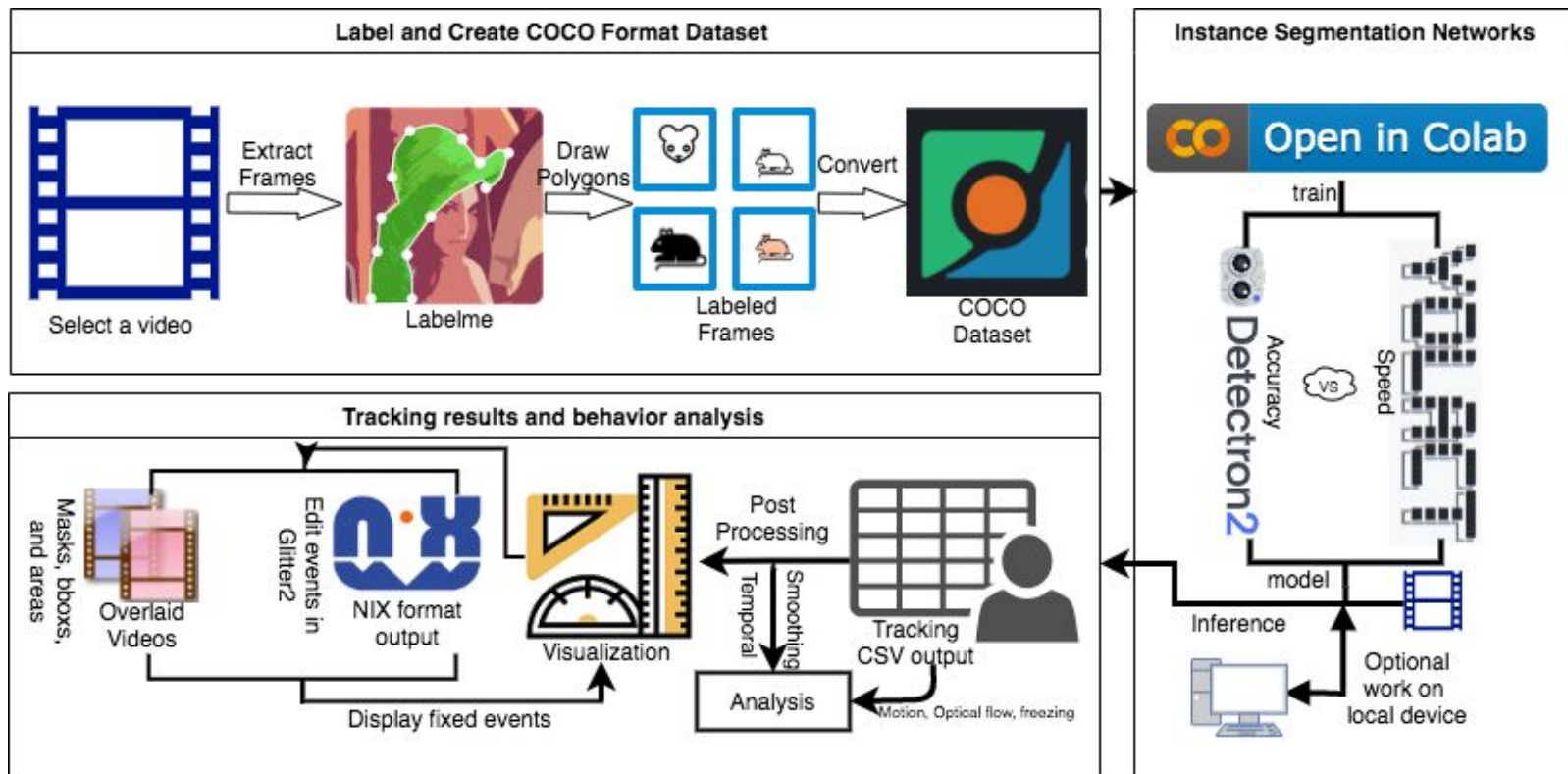

Original video

Mask

(Mask R-CNN or YOLACT)

Behavior identification

Annolid workflow

- Labeling of frames (annotation)
- COCO formatting
- Training and inference (local or Colab)
- Post-processing and analysis

Label and Create COCO Format Dataset

Select a video → Extract Frames → Labelme → Draw Polygons → Labeled Frames → Convert → COCO Dataset

Instance Segmentation Networks

Open in Colab → train

Accuracy Detectron2 vs Speed → model → Inference → Optional work on local device

Tracking results and behavior analysis

Masks, bboxs, and areas → Overlaid Videos → Edit events in Glitter2 → NIX format output → Visualization → Post Processing → Temporal Smoothing → Analysis → Tracking CSV output

Display fixed events

Motion, Optical flow, freezing

How do we label animals and behaviors?
Polygons & keypoints(e.g. Vole_1, nose, tail_base, rearing…...)
~100 labeled frames
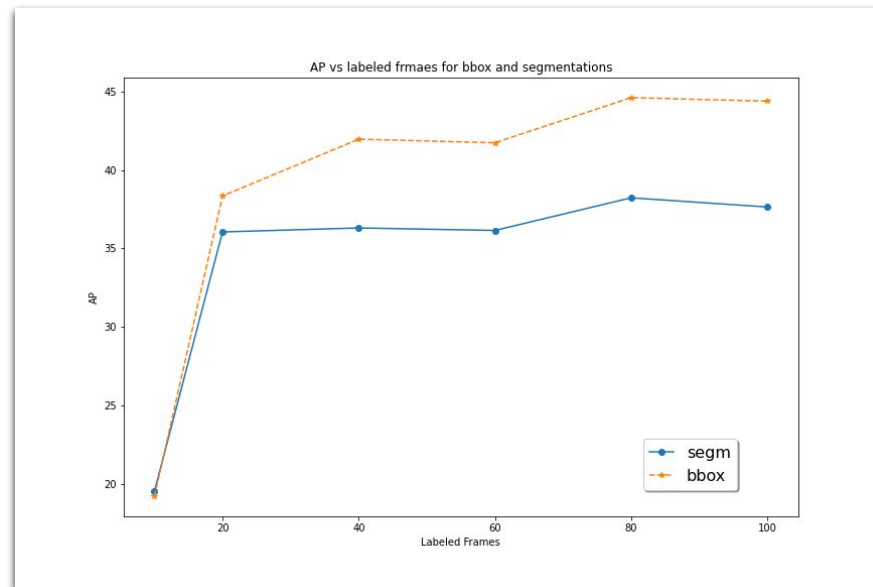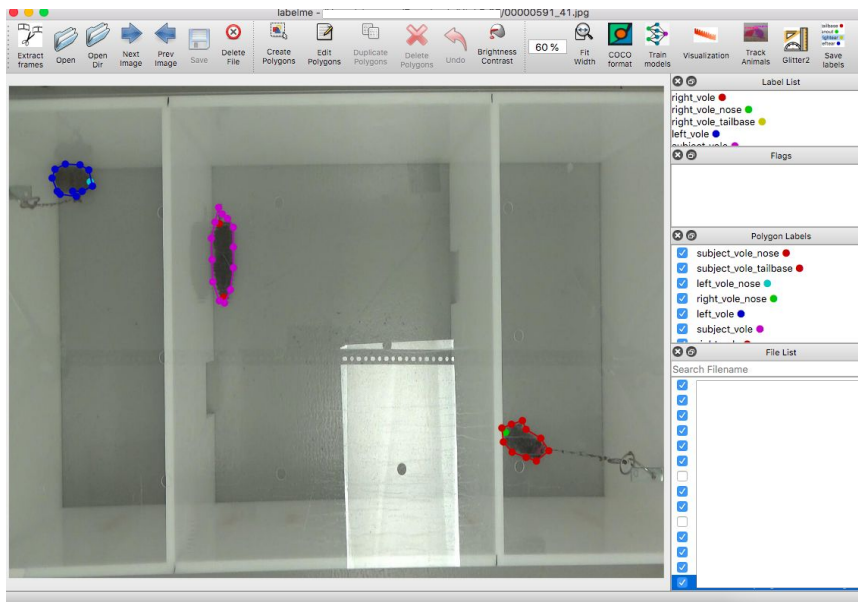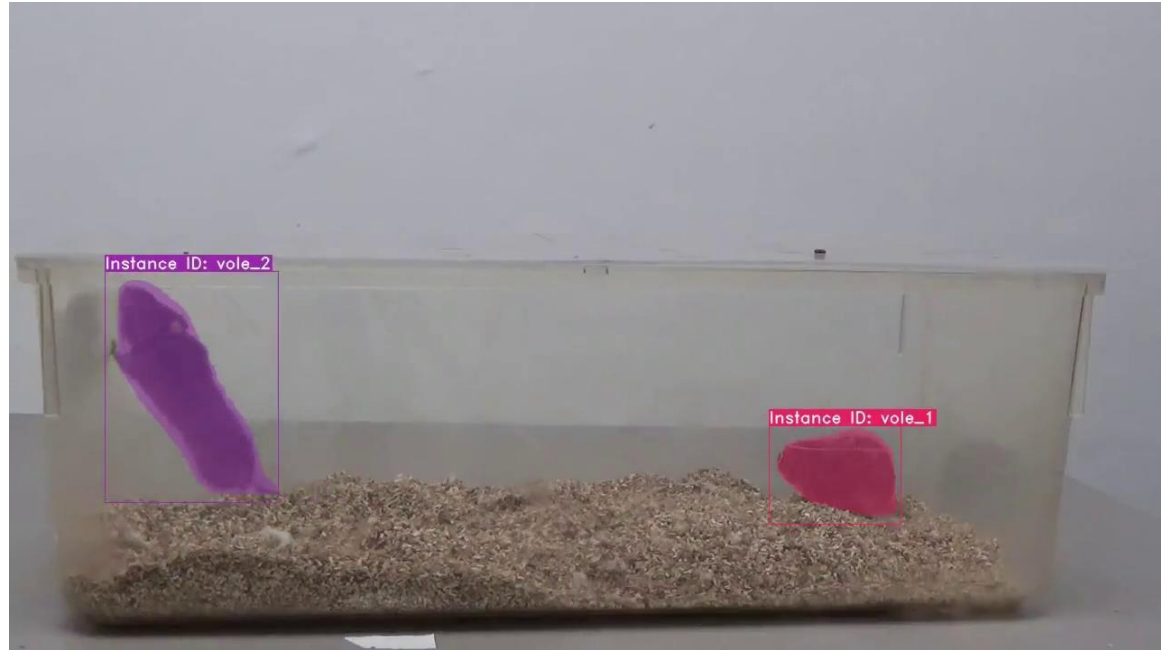
Tracking and re-identification

To link instances across frames, we treat each instance as its own class across frames.

To track multiple animals, label each animal as a separate instance.

To generalize across animals or videos, label multiple animals as examples of the same instance.
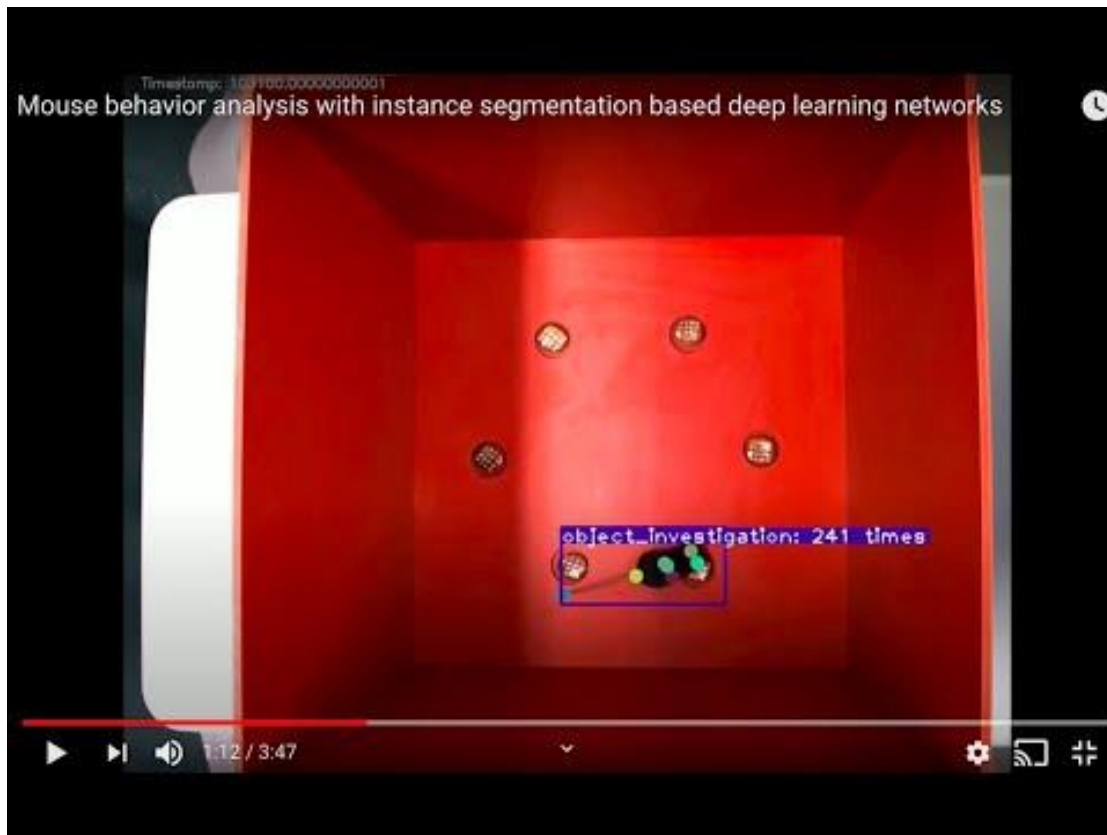
Applying Annolid to diverse end-user goals

Animal tracking

Keypoint tracking of body parts

Automated behavior identification (bounding box)

Applying Annolid to diverse end-user goals

Multiple animal tracking, including periods of partial body occlusion

Whole-body masking

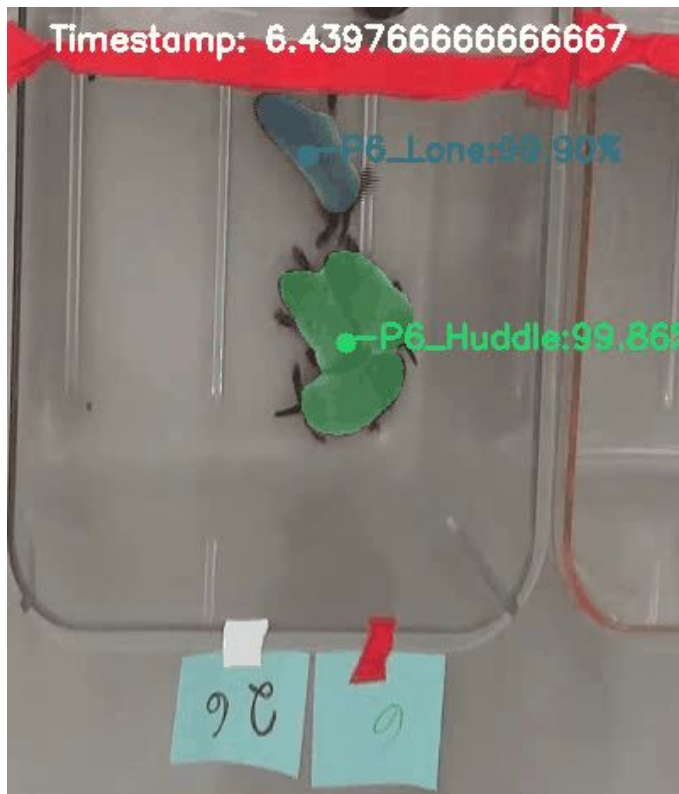Automated identification of interactions (bounding box)

Applying Annolid to diverse end-user goals

Animal tracking

Masking and automatic scoring of *lone* animals and *huddles* of multiple animals

Confidence of identification reported

Applying Annolid to diverse end-user goals

Animal and object tracking, including periods of occlusion

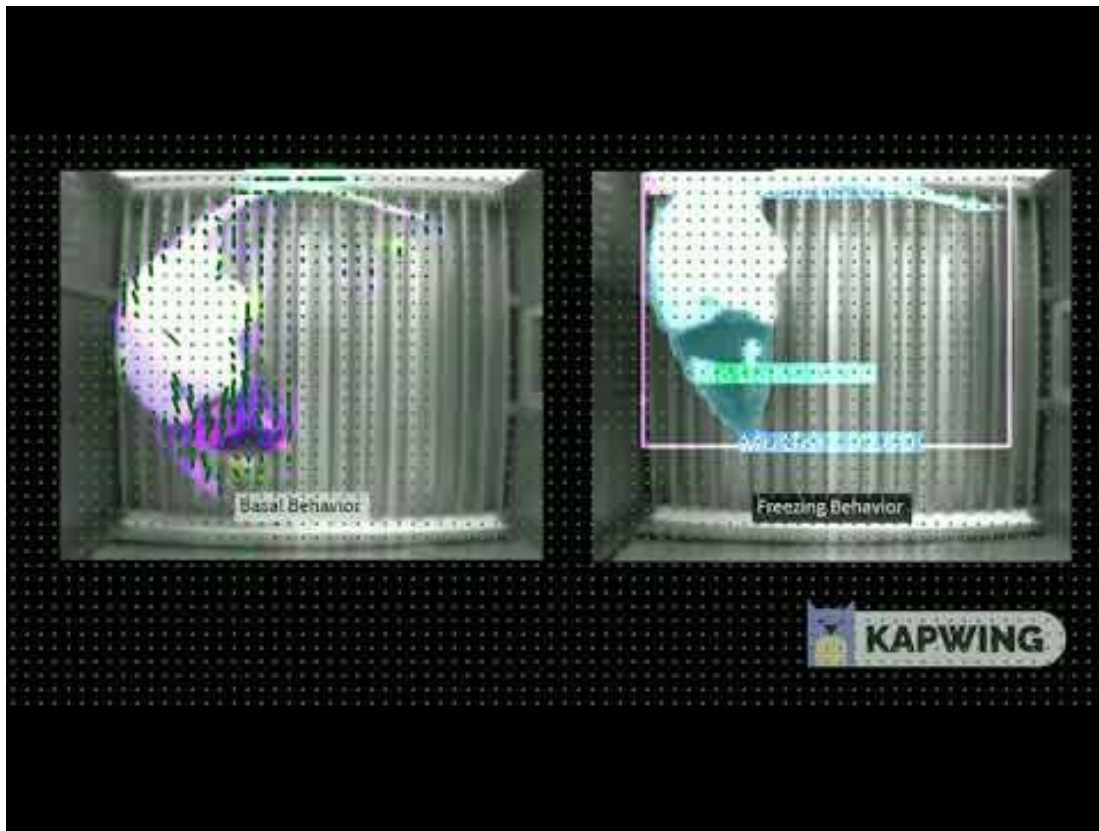Tracked objects automatically associated with user-defined zones

Robustness to noisy background

Applying Annolid to diverse end-user goals

Identification of freezing
behavior (e.g., from fear
conditioning)

Reporting of *motion
score* based on *optical
flow* measurements
applied selectively to the
body mask

https://www.youtube.com/watch?v=qFABuhoGr_E
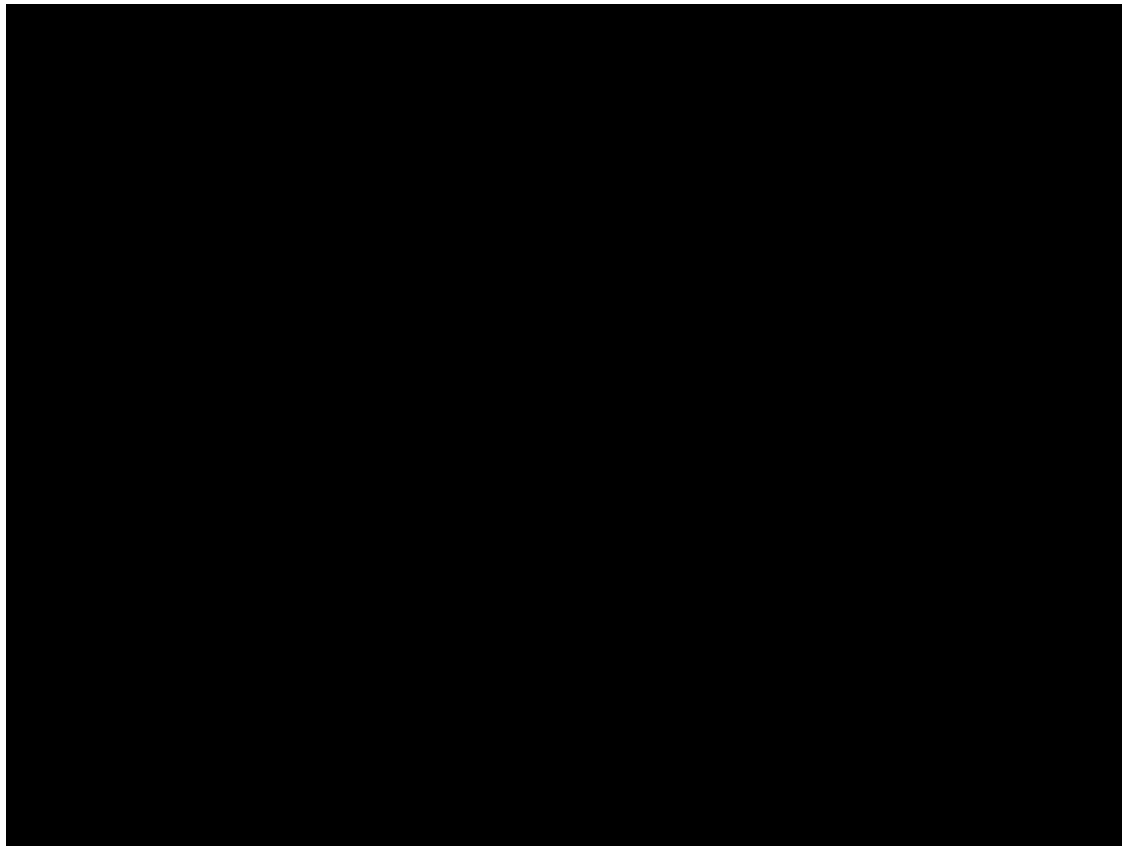
Applying Annolid to diverse end-user goals

Multiple animal tracking
on cryptic background

Video courtesy of Jessica Nowicki, Julia Cora-anne Lee, and Lauren O'Connell

Applying Annolid to diverse end-user goals

Multiple animal tracking
with a large field of view

Accessibility and efficiency

- Options for training on Google Colab (as well as on a local workstation)

- Fast training with quality- and speed-optimized options

  Model training
  - *200 labeled images*
  - *< 2 hours for 3000 iterations on Colab*
  - *30 min on NVidia 1080Ti*

  Inference (applying trained model to behavior videos)
  - *Mask R-CNN: ~7 FPS*
  - *YOLACT: ~30 FPS*

- Transfer learning based on existing models trained on COCO dataset improves performance

- Capacities for *autolabeling* and *human-in-the-loop iterative model training*
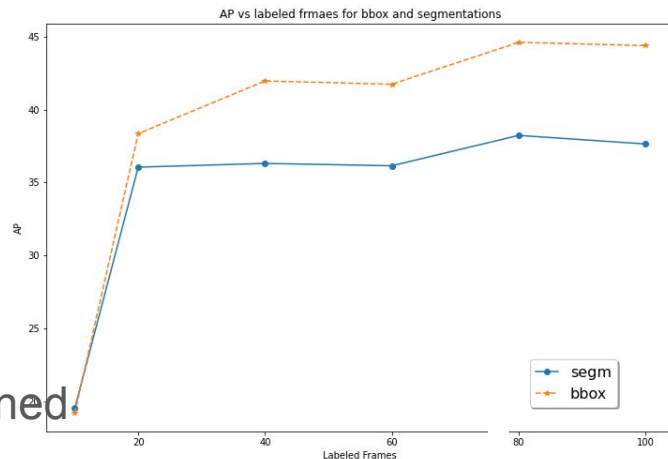
Conclusion and future directions

- Annolid is based on *instance segmentation* models, which facilitate the tracking of multiple animals along with flexible state identification (e.g., behavior classification, urine deposition, interactions among objects)

- Self-supervised, weakly-supervised, and unsupervised training options

- Incorporation of *optical flow* mechanics for improved performance

- Improved labeling efficiency via *autolabeling* and *iterative model training*

Annolid and Glitter2 are both freely available:

https://cplab.science/annolid
https://cplab.science/annolid-tutorial-det2
https://cplab.science/glitter2

Thanks for listening!