Annolid Installation and Quick Start

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

Installation instructions version 0.1, April 2024

1 Installing the Anaconda Environment for Python

If you are not already comfortable in your chosen Python environment, we suggest that you install the Anaconda (or Miniconda) environment for Windows, Linux, or MacOS, with its Conda virtual environment and package manager. These instructions assume a Conda environment. If you already have Conda installed, proceed to section 2.

Installing Anaconda on Windows

1. **Download the Anaconda installer**
   - Go to the Anaconda website: `https://www.anaconda.com/products/distribution`.
   - Download the appropriate version of Anaconda for Windows (32-bit or 64-bit).
2. **Run the installer**
   - Once the installer is downloaded, double-click the executable file to launch the installer.
3. **Follow the installation wizard**
   - Follow the prompts in the installation wizard.
   - Select the installation location and (optionally) add Anaconda to your PATH environment variable. Follow the defaults unless you have a particular reason to do otherwise.
   - Once Anaconda is installed, you can open the Conda environment via the Anaconda Navigator (graphical interface) or the Anaconda Prompt (text window resembling the CMD.exe command line). Both are accessible via the Windows Start menu. We will use the Anaconda Prompt in this document.
4. **Verify installation**
   - Open the Anaconda Prompt from the Start menu.
   - In the Anaconda Prompt window, type `conda --version` and press Enter. This command should return the Conda version that is installed. If it works, you have successfully installed Anaconda.

Installing Anaconda on Linux

1. **Download the Anaconda installer**

   - Go to the Anaconda website: `https://www.anaconda.com/products/distribution`.
   - Download the appropriate version of Anaconda for Linux. Miniconda also should suffice if you prefer.

2. **Run the installer script**

   - Open a terminal window (Ctrl-Alt-T). We assume `bash` compatibility.
   - Use the `cd` command to navigate to the directory in which the Anaconda installer was downloaded. Use the `ls` command to identify the full filename of the downloaded shell script (.sh suffix).
   - Use the following command to run the Anaconda installer script:

     ```
     bash Anaconda3-<version>-Linux-x86_64.sh
     ```

     Replace `<version>` with the version number of Anaconda you downloaded (that is, exactly match the filename of the downloaded shell script).

   - Follow the prompts to agree to the license terms, select the installation location, and confirm the installation.

     The Anaconda installer may ask "Do you wish to update your shell profile to automatically initialize conda?". If you say yes, then whenever you open a new terminal you will be in the conda environment rather than in the default bash environment (that is, it will be like opening an Anaconda Prompt under Windows). This is fine, but optional.

     Alternatively, you can add the path to the conda executable to your `.bashrc` file manually. Use your favorite text editor to do this; if you don't have a favorite, `gedit` is user-friendly. Type the following into your terminal.

     ```
     gedit ~/.bashrc
     ```

     This will open `.bashrc` in `gedit`, a simple GUI-based text editor. (There should be substantial text in your `.bashrc` file already (>100 lines) – if there isn't, something is wrong; probably you are not editing your existing `.bashrc` but creating a new file, which won't work for present purposes). Append this line to the end of the `.bashrc` file, as shown in Figure 1.

     ```
     export PATH=~/anaconda3/bin:$PATH
     ```

     Save the file in `gedit` (via its GUI button) and then quit `gedit`.

3. **Activate the installation**

   - After installation, type the following into your terminal).

     ```
     source ~/.bashrc
     ```

Figure 1: The line to add to `.bashrc`, shown in the `gedit` text editor, that will add the conda directory to your system path, enabling conda to be easily run from any terminal.

4. **Verify and initialize installation**
   - Open a new terminal window.
   - Type `conda --version` and press Enter. This command should return the Conda version that is installed. If it works, you have successfully installed Anaconda.
   - Then, to initialize conda, type

     ```
     conda init
     ```

   - Close the current terminal window.

Installing Anaconda on MacOS

1. **Download the Anaconda installer:**
   - Go to the Anaconda website: `https://www.anaconda.com/products/distribution`.
   - Download the appropriate version of Anaconda for MacOS.
2. **Run the installer script**
   - Launch Terminal from the Applications/Utilities folder or by searching in Spotlight.
   - Use the `cd` command to navigate to the directory in which the Anaconda installer was downloaded (most likely the Download directory). Use the `ls` command to identify the full filename of the downloaded shell script (.sh suffix).
   - Use the following command to run the Anaconda installer script:

     ```
     bash Anaconda3-<version>-MacOSX-x86_64.sh
     ```

     Replace `<version>` with the version number of Anaconda you downloaded (that is, exactly match the filename of the downloaded shell script).
   - Follow the prompts to agree to the license terms, select the installation location, and confirm the installation.
   - After installation, source the `.bash_profile` file to activate the changes:

     ```
     source ~/.bash_profile
     ```

3. **Verify installation**

   - Open a new terminal window.
   - Type `conda --version` and press Enter. This command should return the Conda version that is installed. If it works, you have successfully installed Anaconda.

2 Creating a Conda virtual environment for Annolid

1. Open the Anaconda Prompt (Windows) or a Terminal window (Linux, MacOS).
2. Check the installed version of Conda.

   ```
   conda -V
   ```

3. Create a new virtual environment for Annolid (here we name this environment *annolid* and specify which version of Python to install in that virtual environment). As of this writing, Annolid requires Python 3.11; it is likely to also work with later versions.

   ```
   conda create -n annolid python=3.11
   ```

4. Activate the newly created environment.

   ```
   conda activate annolid
   ```

   You now can (optionally) install and test the packages necessary for GPU computation in Annolid (section 3), or else proceed directly to installing the Annolid application (section 5).

3 Installing packages for GPU computation

By default, if you 'pip install -e . ', it will install a CPU version of PyTorch. If you don't have a GPU on the device, you do not need to install PyTorch with the following steps. If you do have a GPU on your device, please install PyTorch for the correct CUDA version first and then install Annolid by 'pip install -e .'. Note that if you install 'pip install -e . ' first and then install PyTorch for the CUDA version, it may result in a QBuffer error, likely associated with the Pillow version, which might be updated during the PyTorch CUDA installation. To resolve this, simply run 'pip install -e .' again within your Annolid environment. This command ensures that Pillow is within the version range of '>=9.3.0,<=9.5.0', consequently resolving the QBuffer error.

Optionally, if you have an NVIDIA GPU card installed, you can install Python packages to take advantage of GPU computational resources from Annolid, and ensure that they are properly installed and functional. To do so, follow the directions below.

1. Activate the Annolid conda virtual environment if it is not already activated.

   ```
   conda activate annolid
   ```

2. **Verify CUDA Version**. The NVIDIA CUDA toolkit provides a development environment for creating GPU-accelerated applications on NVIDIA graphics cards. To check the current CUDA version installed in your system, open a terminal, Anaconda Prompt, or regular Windows command prompt and run the following command.

   ```
   nvidia-smi
   ```

   The current CUDA version will be listed in the top right corner of the output.

   NOTE: If you intend to update your CUDA version, do that first. We don't discuss that process here, but the gist is (1) update your NVIDIA drivers, (2) update CUDA, then (3) update other packages that may be relevant. This is all done at the operating system level, not within Python, Anaconda, or Annolid.

3. **Install PyTorch**. The PyTorch libraries provide access to CUDA resources from Python, and hence from Annolid as well.

   - First, determine which PyTorch versions are compatible with your current CUDA version (and Python version): `https://pytorch.org/get-started/locally/`.
   - Then, install an appropriate PyTorch version into the Annolid environment using Conda. From a Conda terminal, with the Annolid environment activated, run the following commands. Note that the argument to `pytorch-cuda` refers to the version of CUDA that you have installed, and the two -c arguments are not packages, but specify additional channels in which to search for the listed packages. The other line items – pytorch, torchvision, and torchaudio – are Python packages that will be installed (along with their dependencies).

     The following is a standard installation command for CUDA 12.1 (or greater), which will install the most recent compatible versions.

     ```
     conda install pytorch torchvision torchaudio \
     pytorch-cuda=12.1 -c pytorch -c nvidia
     ```

     (If you have CUDA 12.2 installed, for example, then specify `pytorch-cuda=12.1`, as 12.2 is not yet specified for PyTorch installation (as of this writing). Existing options are detailed at `https://pytorch.org/get-started/locally/`).

4 Choosing the Right PyTorch-CUDA Version

When installing Annolid, it's essential to ensure compatibility between PyTorch and CUDA versions. PyTorch offers pre-built packages tailored for specific CUDA versions to optimize performance and stability. Here's a quick guide to help you select the appropriate PyTorch-CUDA version:

(a) **Check Your CUDA Version:** Firstly, determine which CUDA version is installed on your computer. You can do this by running the following command in your terminal:

```
nvcc --version
```

This command will display information about the CUDA toolkit installed on your system, including the version number.

(b) **Match PyTorch-CUDA Version:** Once you've identified your CUDA version, you need to choose the corresponding PyTorch-CUDA package. PyTorch provides pre-built packages for specific CUDA versions, ensuring optimal compatibility and performance.
For example, if your CUDA version is 12.2 and you're installing PyTorch, you might encounter an issue as PyTorch-CUDA packages are available for CUDA 11.8 and 12.1 only. In such cases, you should specify the closest compatible version, which in this case would be 12.1.

(c) **Specifying the PyTorch-CUDA Version:** During installation, when specifying the PyTorch-CUDA version using package managers like pip, ensure to specify the correct version according to your CUDA installation. For instance:

```
conda install pytorch torchvision torchaudio \
pytorch-cuda=12.1 -c pytorch -c nvidia
```

In this command:
- `cu12.1` specifies the CUDA version 12.1.

(d) **Handling Unsupported CUDA Versions:** If you attempt to specify a PyTorch-CUDA version that does not exist (e.g., specifying CUDA 12.2 when it's not supported by PyTorch), you may encounter an error indicating that the package does not exist. In such cases, refer to the closest compatible PyTorch-CUDA version.

By following these steps, you can ensure a smooth installation process, minimizing compatibility issues between PyTorch and CUDA versions. If you encounter any difficulties or have further questions, don't hesitate to reach out to our support team for assistance. We're here to help you get started with Annolid hassle-free!

To be more conservative, the following exact versions are the same as the present Annolid development environment (as of this document's writing), and can be installed specifically.

```
conda install pytorch==2.2.0 torchvision==0.17.0 \
torchaudio==2.2.0 pytorch-cuda=12.1 -c pytorch -c nvidia
```

4. **Confirm PyTorch and CUDA functionality**. To confirm that PyTorch is installed and working with CUDA, first start Python:

```
python
```

You will get the Python `>>>` prompt. At this prompt, type

```
import torch
torch.cuda.is_available()
```

The latter command should return `True` if CUDA is working. You then can quit Python with

```
exit()
```

5 Installing the Annolid application

1. Activate the Annolid conda environment if it is not already activated.

   ```
   conda activate annolid
   ```

2. Install `git` and its dependencies (in the annolid environment).

   ```
   conda install git
   ```

3. Navigate to the directory (folder) within which you would like to install the Annolid application. (The `git clone` command will create a new directory called `annolid` inside this folder).

4. Clone the Annolid repository and its submodules using Git. This will install the most recent release of Annolid.

   ```
   git clone --recurse-submodules https://github.com/healthonrails/annolid.git
   ```

5. Navigate to the newly-created Annolid application directory.

   ```
   cd annolid
   ```

6. Install Annolid in editable mode using pip. (Don't forget to include the trailing space and period in this command).

   ```
   pip install -e .
   ```

7. Run Annolid. This will bring up the Annolid GUI, and you can start working with Annolid.

   ```
   annolid
   ```

8. Once you are finished working with To deactivate the Annolid environment and return to the base environment, run:

   ```
   conda deactivate
   ```

6 Updating Annolid

There are two strategies to updating Annolid.

1. The first strategy is to use git to update your existing Annolid application to the current version.

   - Activate the the Annolid conda environment if it is not already activated, and navigate to your Annolid application directory, which you selected in section 5.

     ```
     conda activate annolid
     cd annolid
     ```

   - To fetch the most recent updates from the Annolid repository, use the `git pull` command.

     ```
     git pull
     ```

   - Finally, update your Annolid installation to include these changes. (Don't forget to include the trailing space and period in this command).

     ```
     pip install -e .
     ```

2. The second option is to remove the Annolid environment (as described in section 7 below) and then install an updated version from scratch as described above.

In either case, you may want to update the conda environment itself as well. We don't discuss that process here in detail, but the basic command is `conda update conda`; additional details are on the Conda cheatsheet (`https://docs.conda.io/projects/conda/en/latest/user-guide/cheatsheet.html`) or in Conda documentation. If the conda version number (as reported by `conda -V` or `conda --version`) does not update as it should, several potential solutions are discussed at `https://stackoverflow.com/questions/73974735/conda-update-conda-does-not-update-conda`.

7 Removing Annolid

If you wish to remove Annolid, the Annolid environment within Conda, and all of its associated packages from your system, use the following command from within conda base (assuming that the environment is named *annolid*):

```
conda env remove --name annolid
```

After removing the Annolid environment, you may then need to go into the filesystem and manually delete the directory (folder) in which the Annolid conda environment was installed. Find your Anaconda installation folder (probably entitled anaconda3) and navigate to the `anaconda3/envs`

directory/folder therein. If there is a subfolder within `anaconda3/envs` called `annolid` (or whatever the name of your annolid environment was), delete it. This is necessary if you intend to create a new Annolid environment in conda (e.g., to install a newer version of Annolid).

To confirm the current list of Conda environments installed on your system, enter:

```
conda info --envs
```

Finally, you also will need to delete the Annolid application directory (folder) in which you originally installed the Annolid application (section 5).

8 A Quick Start to Tracking with Annolid

This short guide will get you started using Annolid to track animals or other objects in your videos. It is only a starting point, in which we illustrate the semiautomated segmentation of user-specified objects followed by Cutie-based automatic tracking predictions.

For further details about Annolid's new (2024) SAM/Grounding DINO-based automatic segmentation coupled with Cutie-based automatic tracking, consult [1]. For some of Annolid's diverse other behavioral analysis applications, consult [2, 3].

Step 1: Open your video and label instances in one frame

Start by opening Annolid and loading the video you want to work with. You can do this by clicking on the *Open Video* icon in the toolbar or selecting "Open Video" from the File menu.

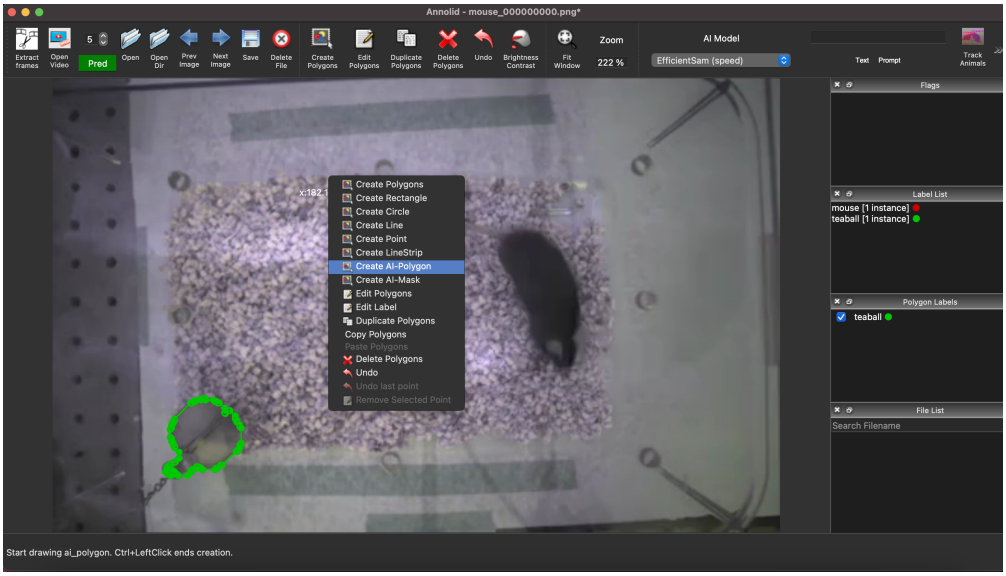

Figure 2: Opening a video and selecting the AI-Polygon tool

Once the video is loaded, navigate to the first frame in which all of your animals and other objects of interest appear. Right-click to bring up the floating menu and choose "AI-Polygon" (Figure 2). (This tool uses Meta's Segment Anything Model to automagically outline most common objects so that you don't have to manually draw the outlining polygon). Click on the object you want to track, and Annolid will generate a polygon around it. You can adjust the polygon as needed and assign a unique identifier to each instance. Clicking on an object while holding down the Control (Command) key will bring up a pop-up dialog (Figure 3) in which you can enter the information for each instance.

When labeling is complete, use Control-S (Command-S) or click on the toolbar Save button to save the labeled instances to a JSON file.

8.1 Step 2: Start automatic predictions (tracking) based on the first frame

After labeling the first frame, it's time to start tracking! Select "Cutie" from the AI Model pulldown menu and click on the green "Pred" button to initiate the tracking process. Annolid will run its

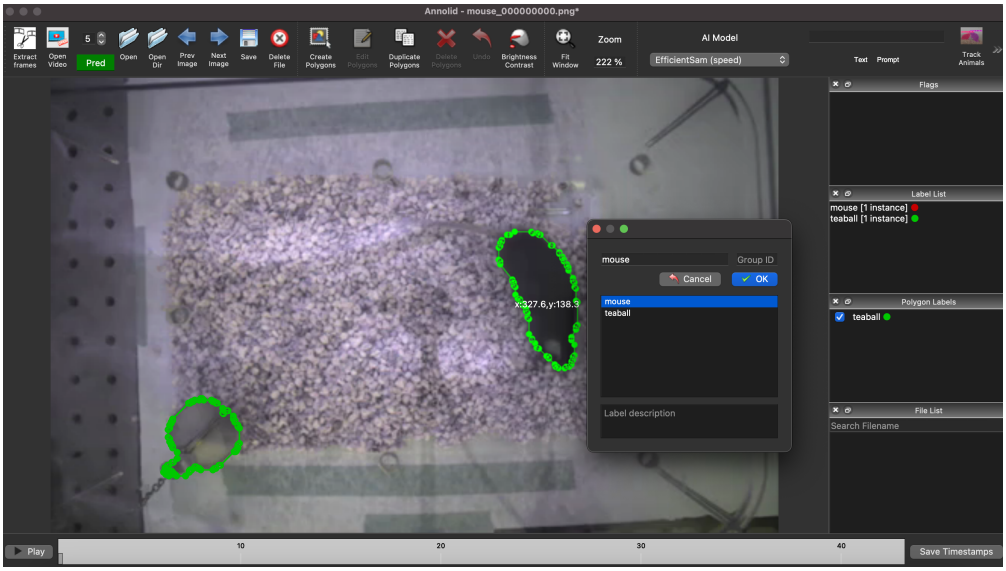

Figure 3: Labeling instances in Annolid (control-click)

model inference in the background, and you can check its predictions in real time by navigating through the completed frames using the slider at the bottom of the screen.

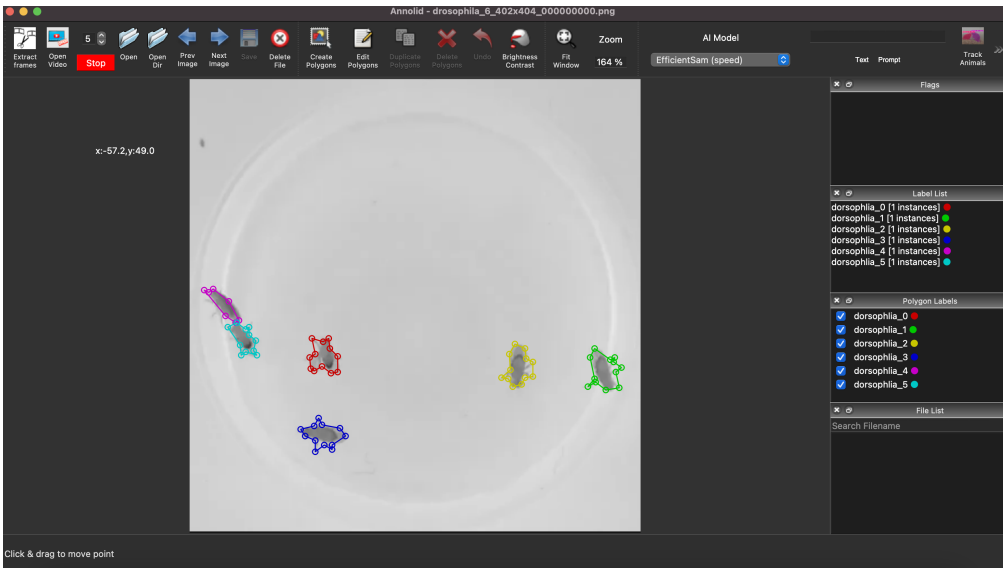

Figure 4: Tracking underway in Annolid

8.2 Step 3: Verify Annolid's predictions and correct if necessary

If you notice any inaccuracies in Annolid's predictions, you can stop the prediction process by clicking the red "Stop" button (Figure 4). Once prediction has been paused, navigate to the first incorrectly segmented frame, make corrections as needed (Figure 5), and restart predictions ("Pred" button). Annolid will make new predictions from that point forward.

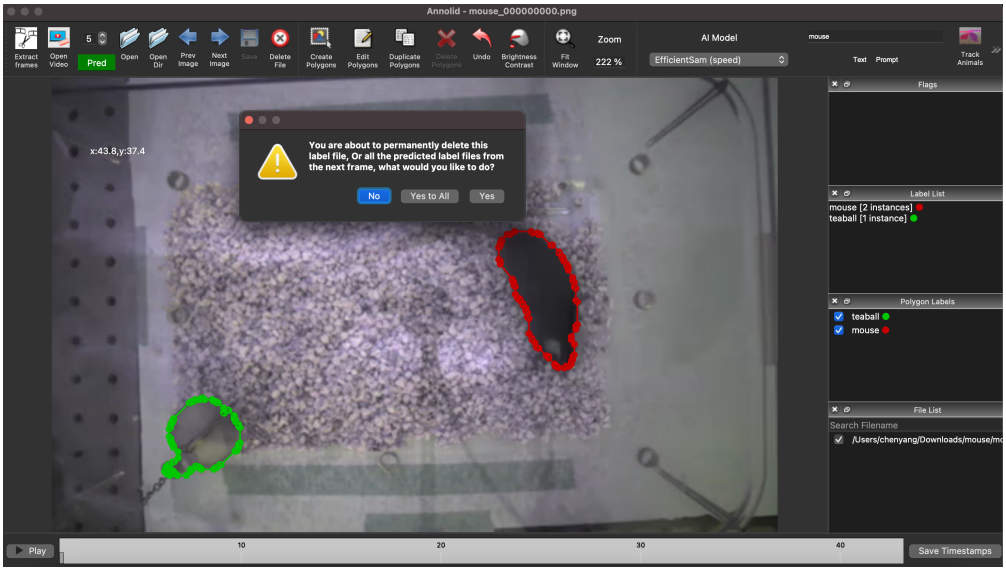

Figure 5: Pausing and correcting predictions in Annolid

8.3 Step 4: Finalize the predictions

Once you are satisfied with the tracking, you can finalize the predictions (Figure 6). Annolid makes it easy to review and complete the tracking task, ensuring accuracy and reliability.

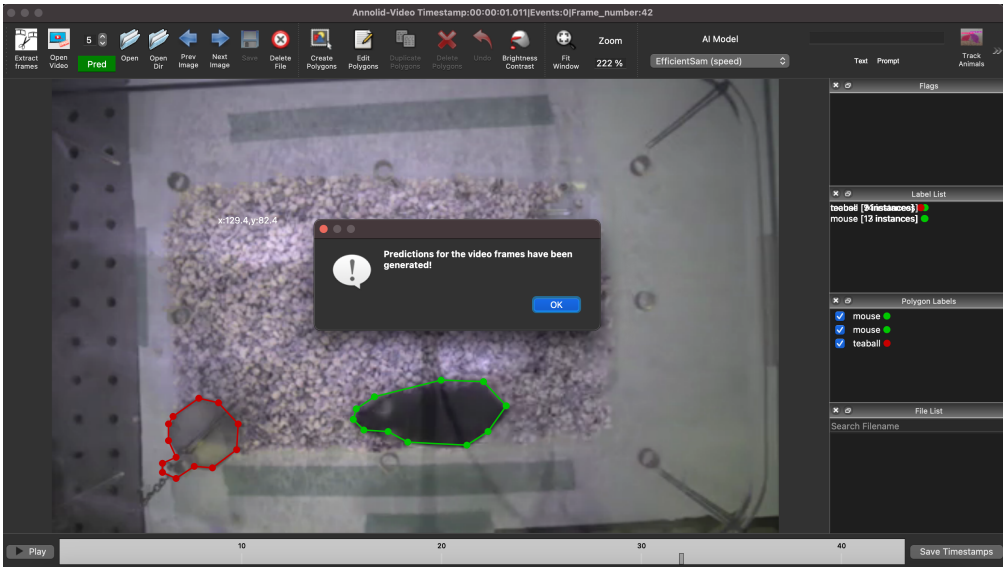

Figure 6: Completing the tracking process in Annolid

Congratulations! You've successfully used Annolid to track objects in your videos. Your tracking results will be saved to a folder with the same name as the video. This folder will contain all the JSON files following the convention 'video_name_frame_number.json', as well as two summary CSV (spreadsheet) files following the filename conventions 'video_name_tracking.csv' and 'video_name_tracked.csv'.

More details are available in our published works (see *References* below) and in the online

Annolid documentation at `https://annolid.com/README.html`.

References

[1] C. Yang and T. A. Cleland, "Annolid: annotate, segment, and track anything you need," *arXiv:2403.18690*, 2024.

[2] C. Yang, J. Forest, M. Einhorn, and T. A. Cleland, "Automated behavioral analysis using instance segmentation," *arXiv:2312.07723*, 2023.

[3] J. Fang, C. Yang, and T. A. Cleland, "Scoring rodent digging behavior with Annolid," *Soc. Neurosci. Abstr. 512.01*, 2023.