DeepLabCut AI Residency

Day 2 Session 3: What could be done next after creating your videos?

July 30 & August 1, 2025

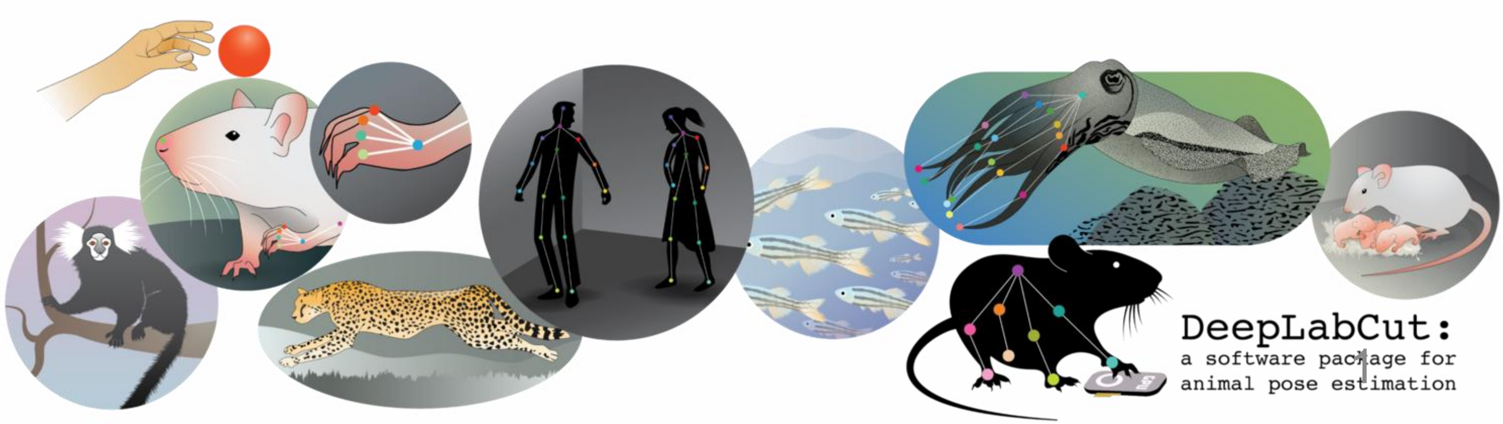
McGill University, Montreal

Jiayue Yang

Vic Shao-Chinh Chiang

DeepLabCut AI Residency
next-gen animal behavior

McGill



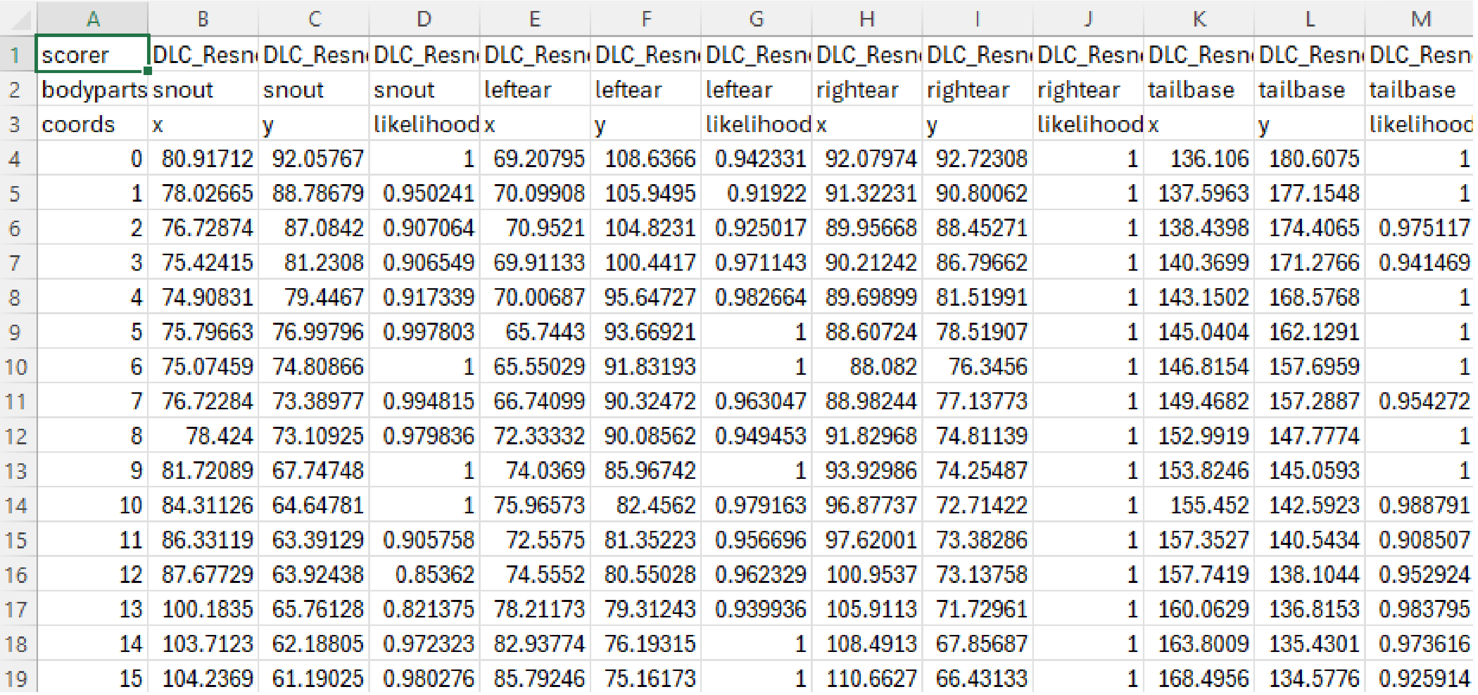
Github troubleshooting

https://github.com/DeepLabCut/DeepLabCut/issues

Post-processing analysis

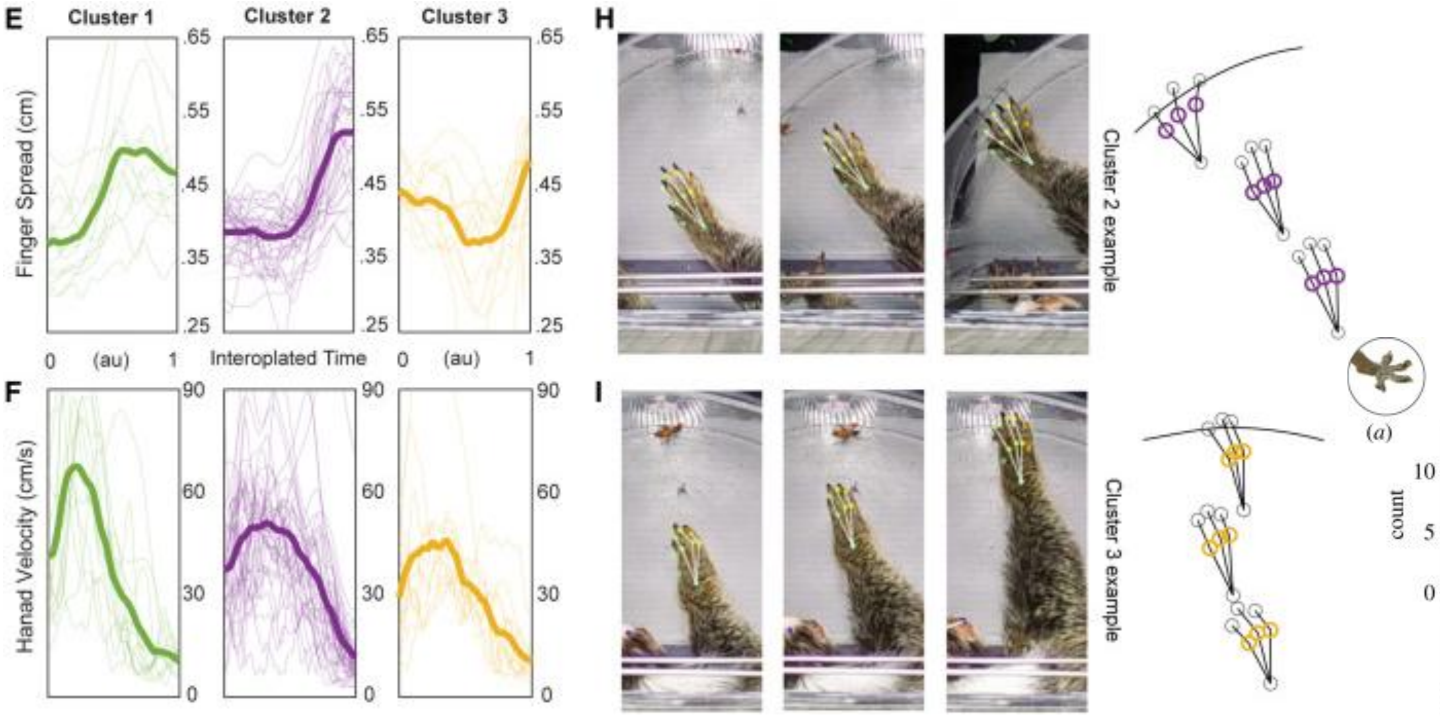
Predictions saved in the .csv file

Predicted labels at each frame and animals (in **pixel**)



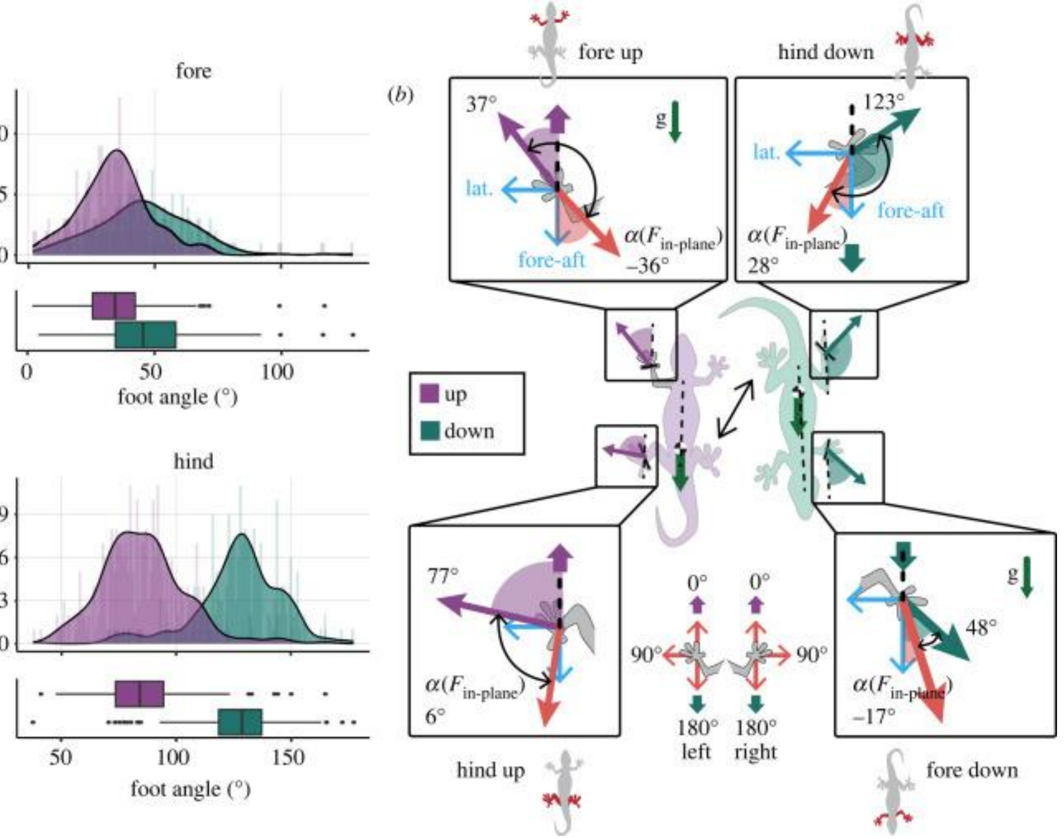
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	scorer	DLC_Resn	DLC_Resn	DLC_Resn	DLC_Resn	DLC_Resn	DLC_Resn	DLC_Resn	DLC_Resn	DLC_Resn	DLC_Resn	DLC_Resn	DLC_Resn
2	bodyparts	snout	snout	snout	leftear	leftear	leftear	rightear	rightear	rightear	tailbase	tailbase	tailbase
3	coords	x	y	likelihood	x	y	likelihood	x	y	likelihood	x	y	likelihood
4	0	80.91712	92.05767	1	69.20795	108.6366	0.942331	92.07974	92.72308	1	136.106	180.6075	1
5	1	78.02665	88.78679	0.950241	70.09908	105.9495	0.91922	91.32231	90.80062	1	137.5963	177.1548	1
6	2	76.72874	87.0842	0.907064	70.9521	104.8231	0.925017	89.95668	88.45271	1	138.4398	174.4065	0.975117
7	3	75.42415	81.2308	0.906549	69.91133	100.4417	0.971143	90.21242	86.79662	1	140.3699	171.2766	0.941469
8	4	74.90831	79.4467	0.917339	70.00687	95.64727	0.982664	89.69899	81.51991	1	143.1502	168.5768	1
9	5	75.79663	76.99796	0.997803	65.7443	93.66921	1	88.60724	78.51907	1	145.0404	162.1291	1
10	6	75.07459	74.80866	1	65.55029	91.83193	1	88.082	76.3456	1	146.8154	157.6959	1
11	7	76.72284	73.38977	0.994815	66.74099	90.32472	0.963047	88.98244	77.13773	1	149.4682	157.2887	0.954272
12	8	78.424	73.10925	0.979836	72.33332	90.08562	0.949453	91.82968	74.81139	1	152.9919	147.7774	1
13	9	81.72089	67.74748	1	74.0369	85.96742	1	93.92986	74.25487	1	153.8246	145.0593	1
14	10	84.31126	64.64781	1	75.96573	82.4562	0.979163	96.87737	72.71422	1	155.452	142.5923	0.988791
15	11	86.33119	63.39129	0.905758	72.5575	81.35223	0.956696	97.62001	73.38286	1	157.3527	140.5434	0.908507
16	12	87.67729	63.92438	0.85362	74.5552	80.55028	0.962329	100.9537	73.13758	1	157.7419	138.1044	0.952924
17	13	100.1835	65.76128	0.821375	78.21173	79.31243	0.939936	105.9113	71.72961	1	160.0629	136.8153	0.983795
18	14	103.7123	62.18805	0.972323	82.93774	76.19315	1	108.4913	67.85687	1	163.8009	135.4301	0.973616
19	15	104.2369	61.19025	0.980276	85.79246	75.16173	1	110.6627	66.43133	1	168.4956	134.5776	0.925914

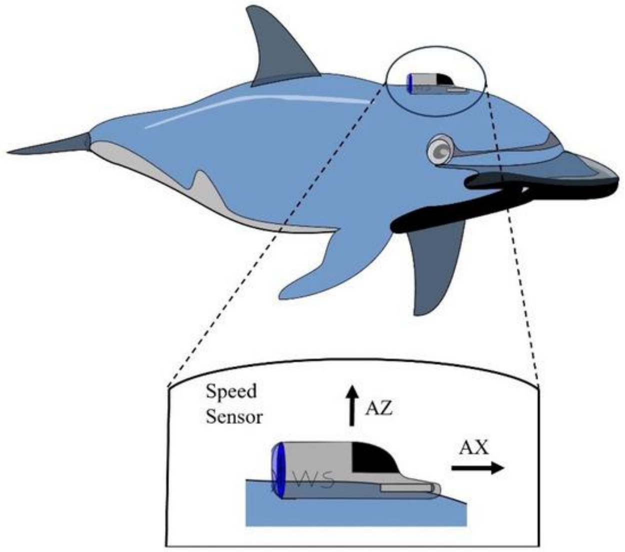
Post-processing analysis



Shaw, et al. (2023). Current Biology.

Schultz, et al. (2023). J R Soc Interface.





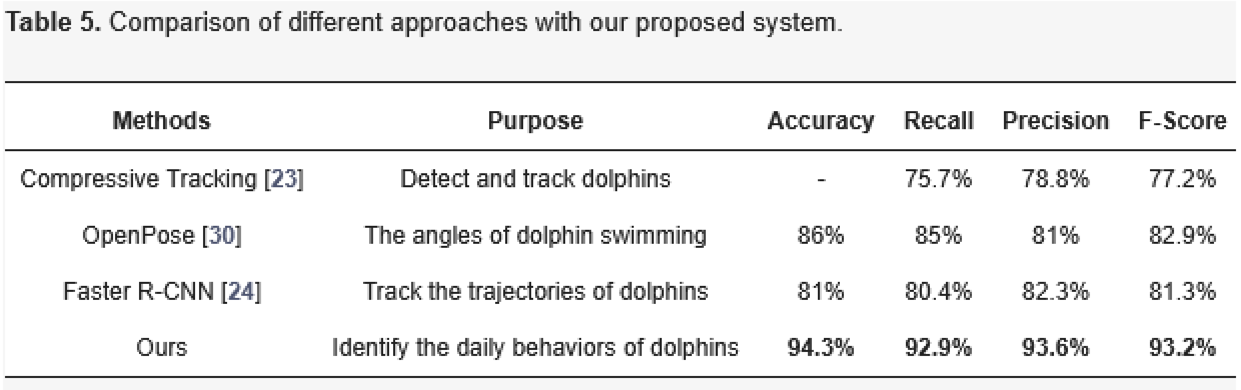
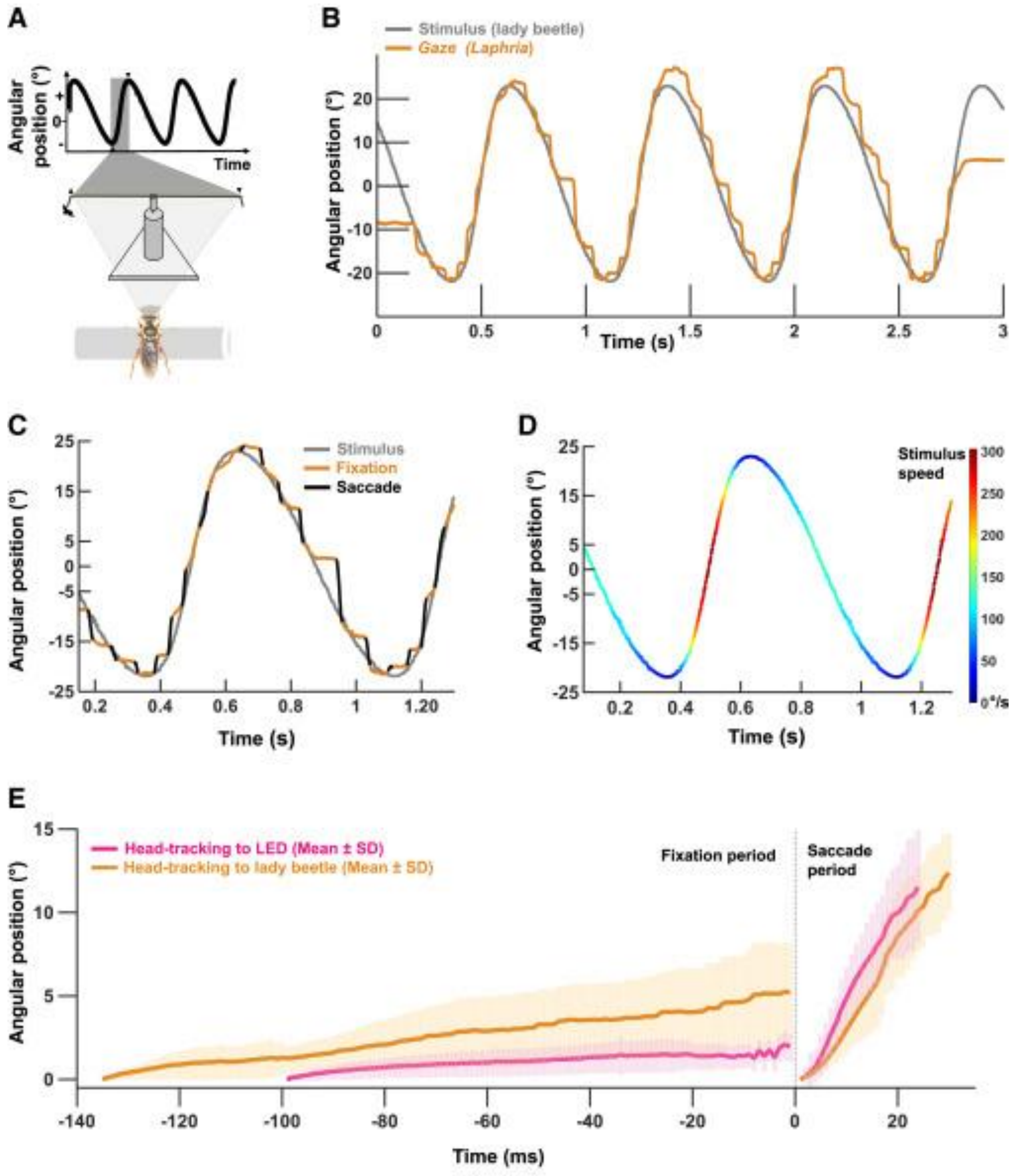


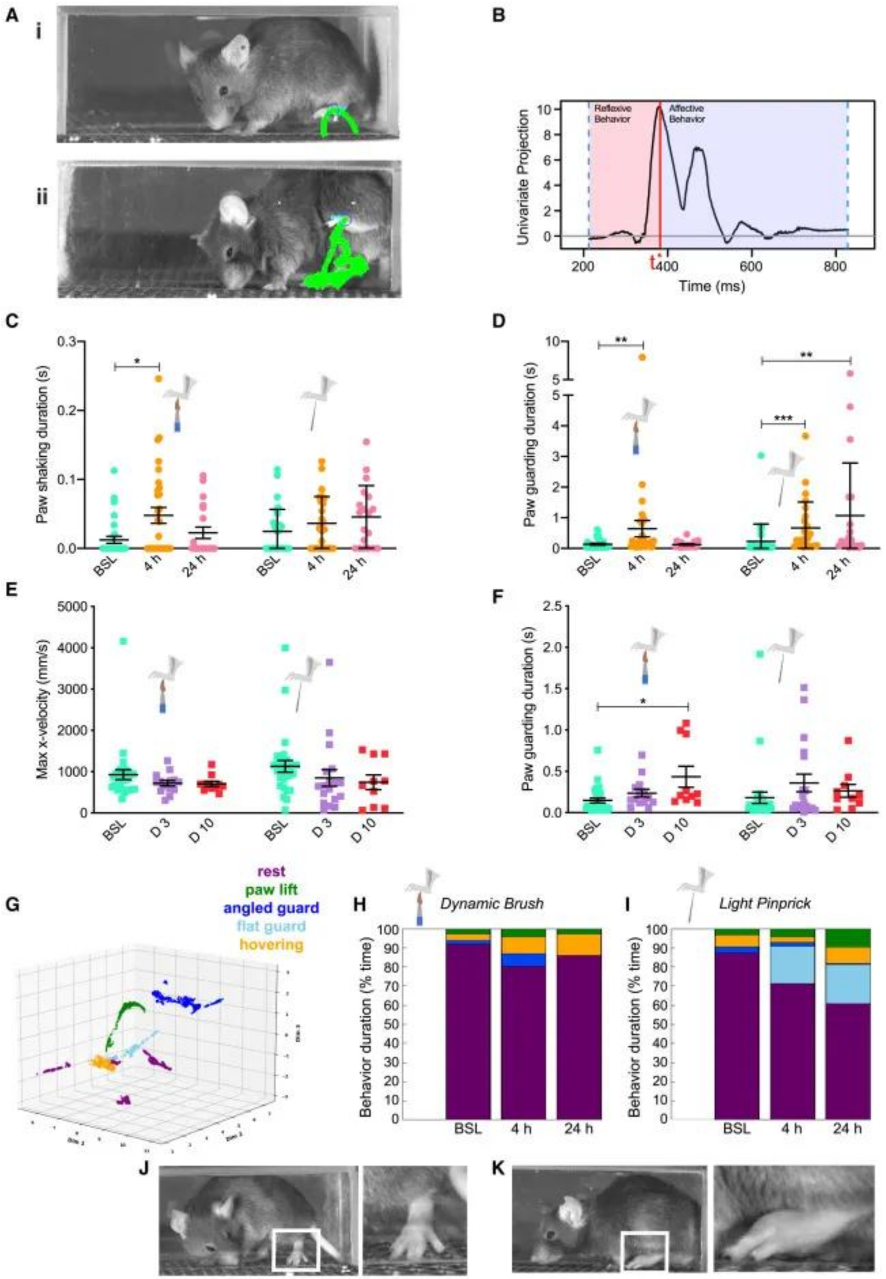
Table 5. Comparison of different approaches with our proposed system.

Methods	Purpose	Accuracy	Recall	Precision	F-Score
Compressive Tracking [23]	Detect and track dolphins	-	75.7%	78.8%	77.2%
OpenPose [30]	The angles of dolphin swimming	86%	85%	81%	82.9%
Faster R-CNN [24]	Track the trajectories of dolphins	81%	80.4%	82.3%	81.3%
Ours	Identify the daily behaviors of dolphins	**94.3%**	**92.9%**	**93.6%**	**93.2%**

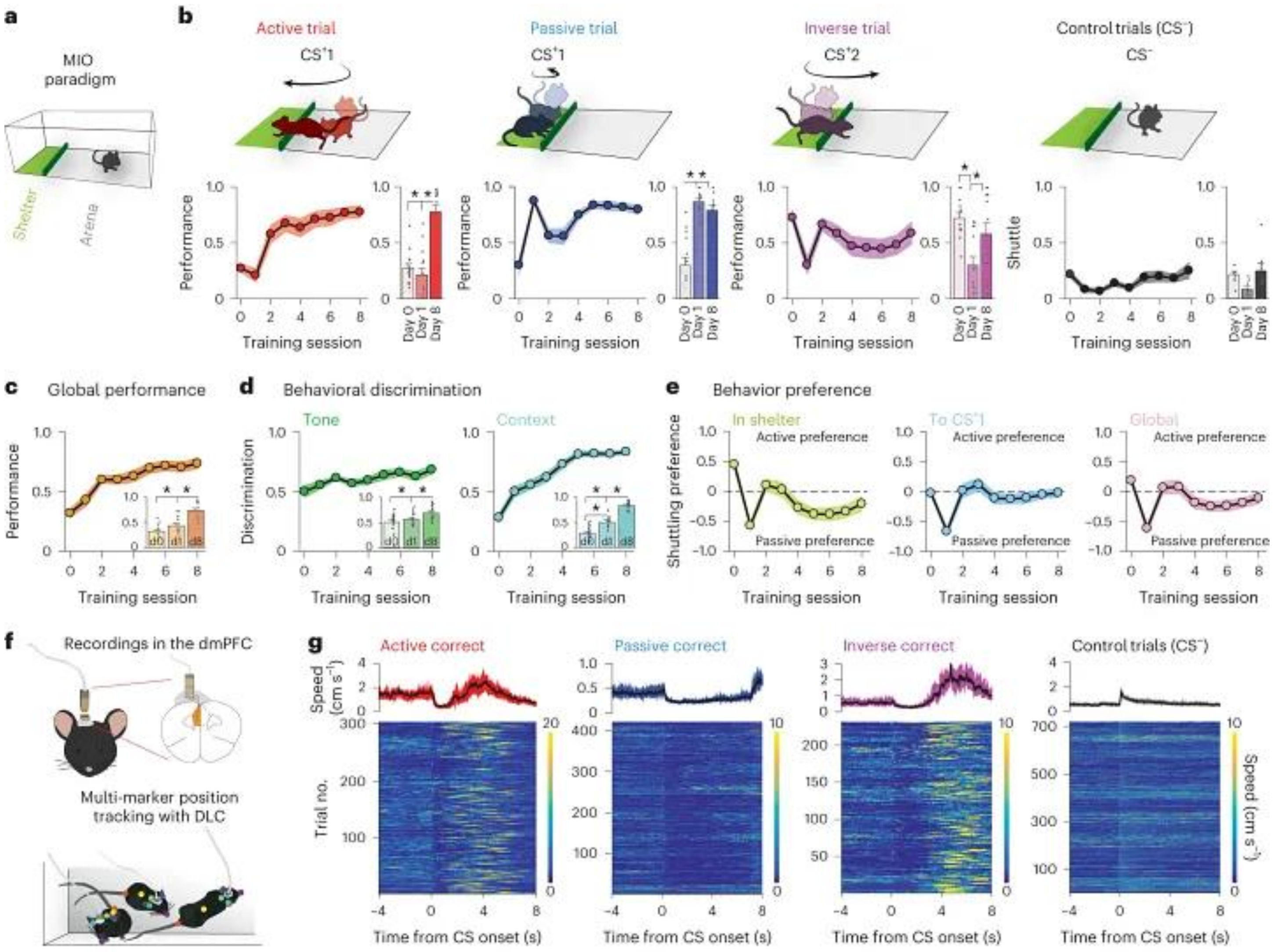
Tseng, et al. (2024). J. Mar. Sci. Eng.



Talley, et al. (2023). Current Biology.

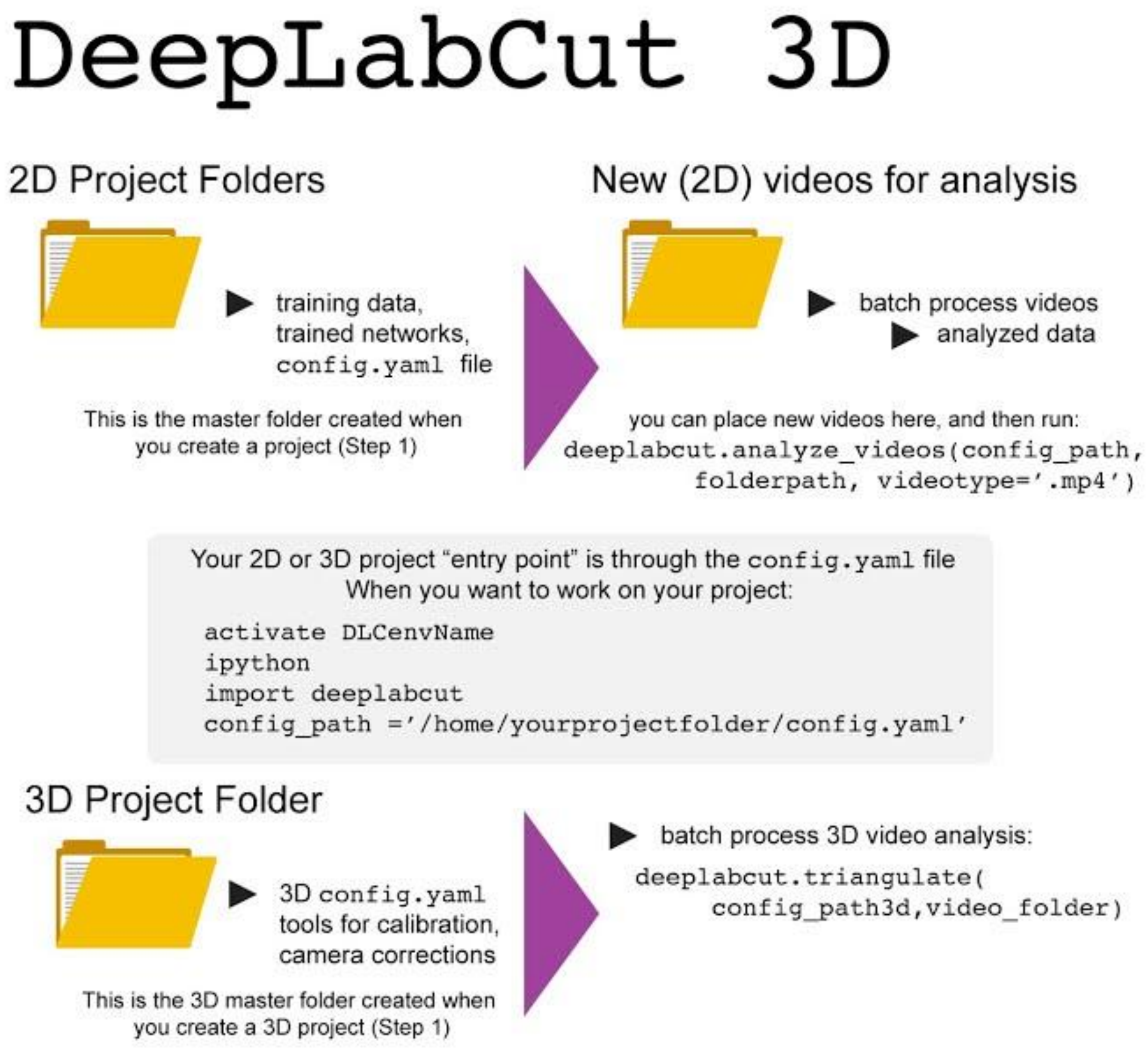


Bohic, et al. (2023). Neuron.



Martin-Fernandez, et al. (2023). Nature Neuroscience.

DeepLabCut 3D



DeepLabCut 3D

2D Project Folders

▶ training data, trained networks, `config.yaml` file

This is the master folder created when you create a project (Step 1)

New (2D) videos for analysis

▶ batch process videos ▶ analyzed data

you can place new videos here, and then run:

```
deeplabcut.analyze_videos(config_path,
                 folderpath, videotype='.mp4')
```

Your 2D or 3D project "entry point" is through the `config.yaml` file
When you want to work on your project:

```
activate DLCenvName
ipython
import deeplabcut
config_path ='/home/yourprojectfolder/config.yaml'
```

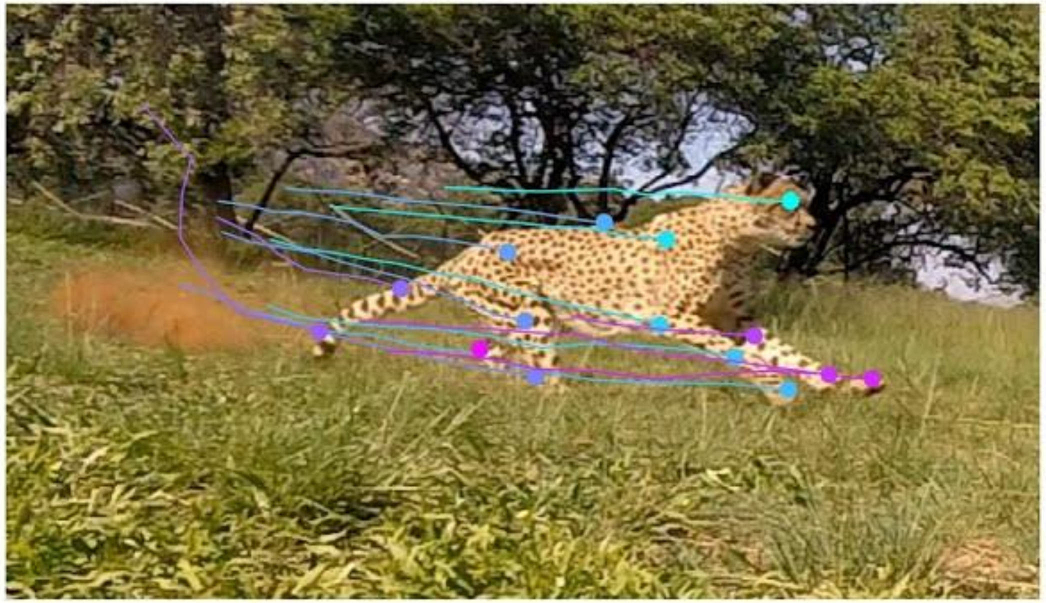
3D Project Folder

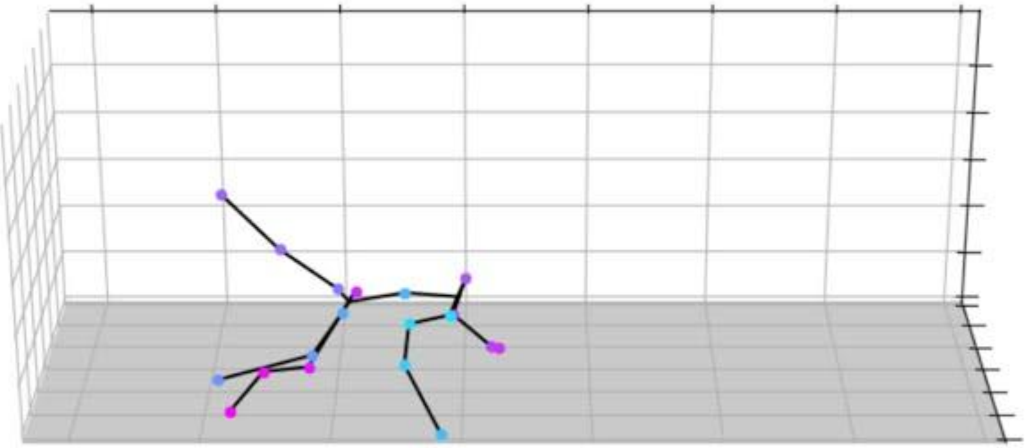
▶ 3D `config.yaml` tools for calibration, camera corrections

This is the 3D master folder created when you create a 3D project (Step 1)

▶ batch process 3D video analysis:

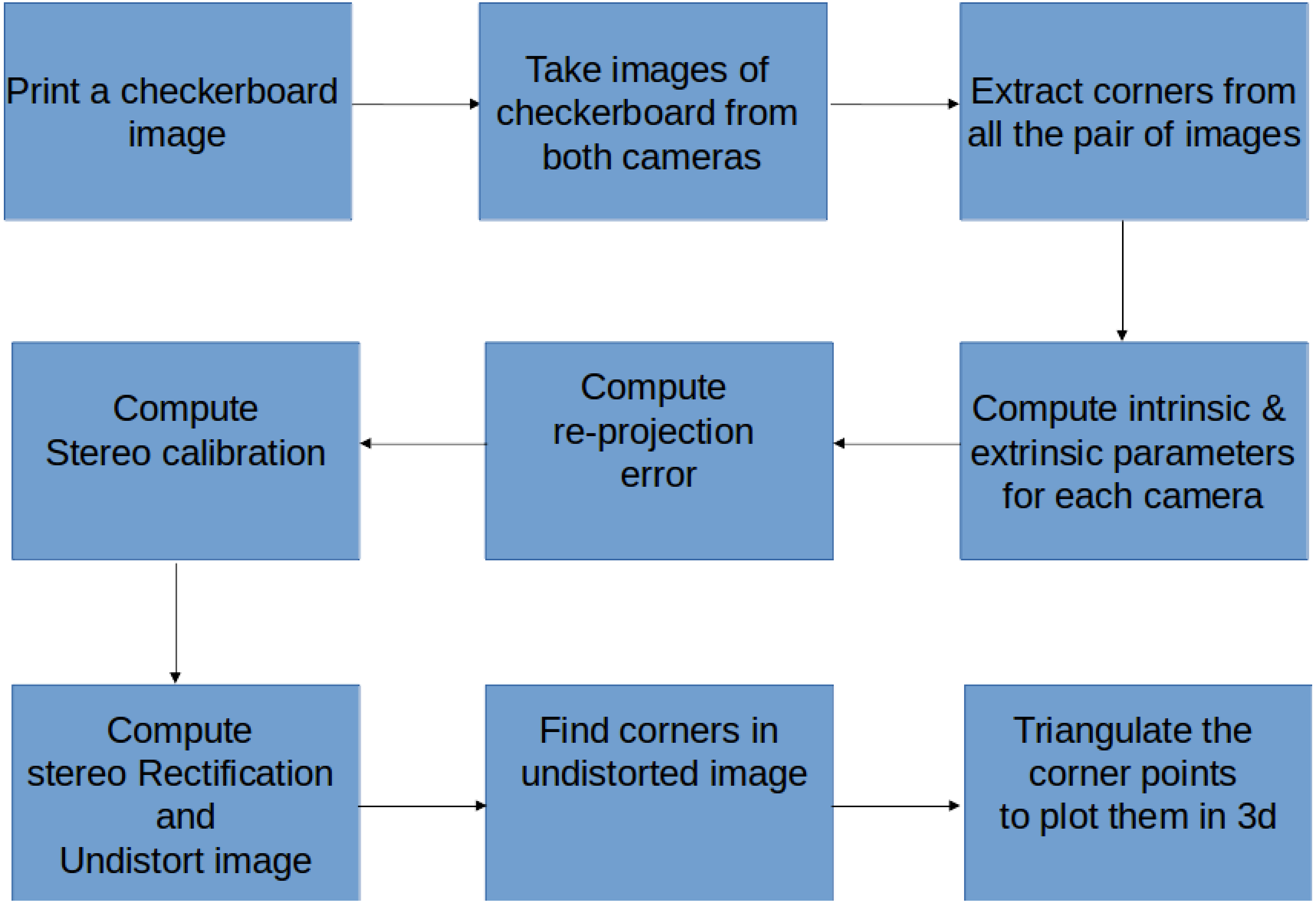
```
deeplabcut.triangulate(
        config_path3d,video_folder)
```





Nath, et al. (2019). Nature Protocols.
Lauer, et al. (2022). Nature Methods.

DeepLabCut 3D workflow



Print a checkerboard image

Take images of checkerboard from both cameras

Extract corners from all the pair of images

Compute intrinsic & extrinsic parameters for each camera

Compute re-projection error

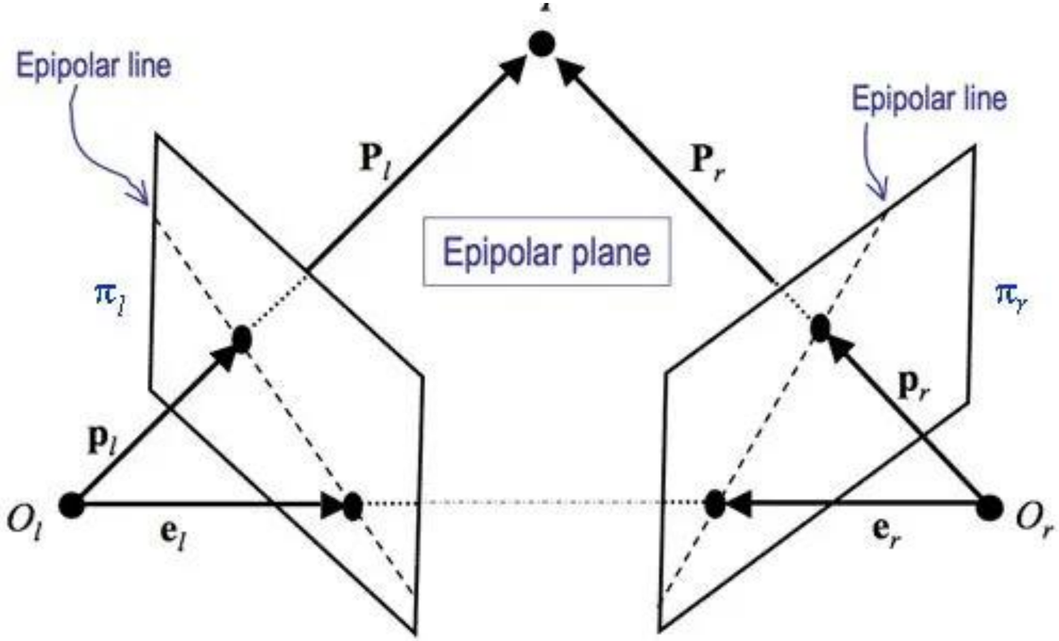
Compute Stereo calibration

Compute stereo Rectification and Undistort image

Find corners in undistorted image

Triangulate the corner points to plot them in 3d

3D Reconstruction



Stereo calibration estimates the **intrinsic** and **extrinsic** parameters of two cameras used together to reconstruct 3D information from 2D images.

- How the 2D cameras' lenses distort the image?
- How far apart they are, and which way they're pointing?

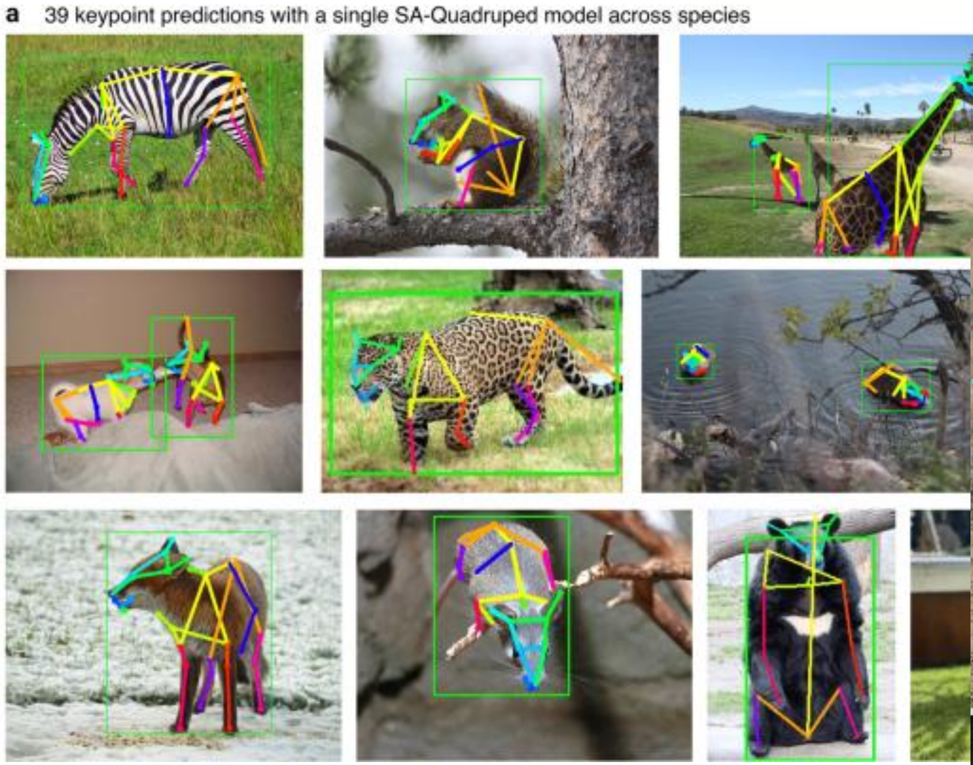
DeepLabCut 3D commands

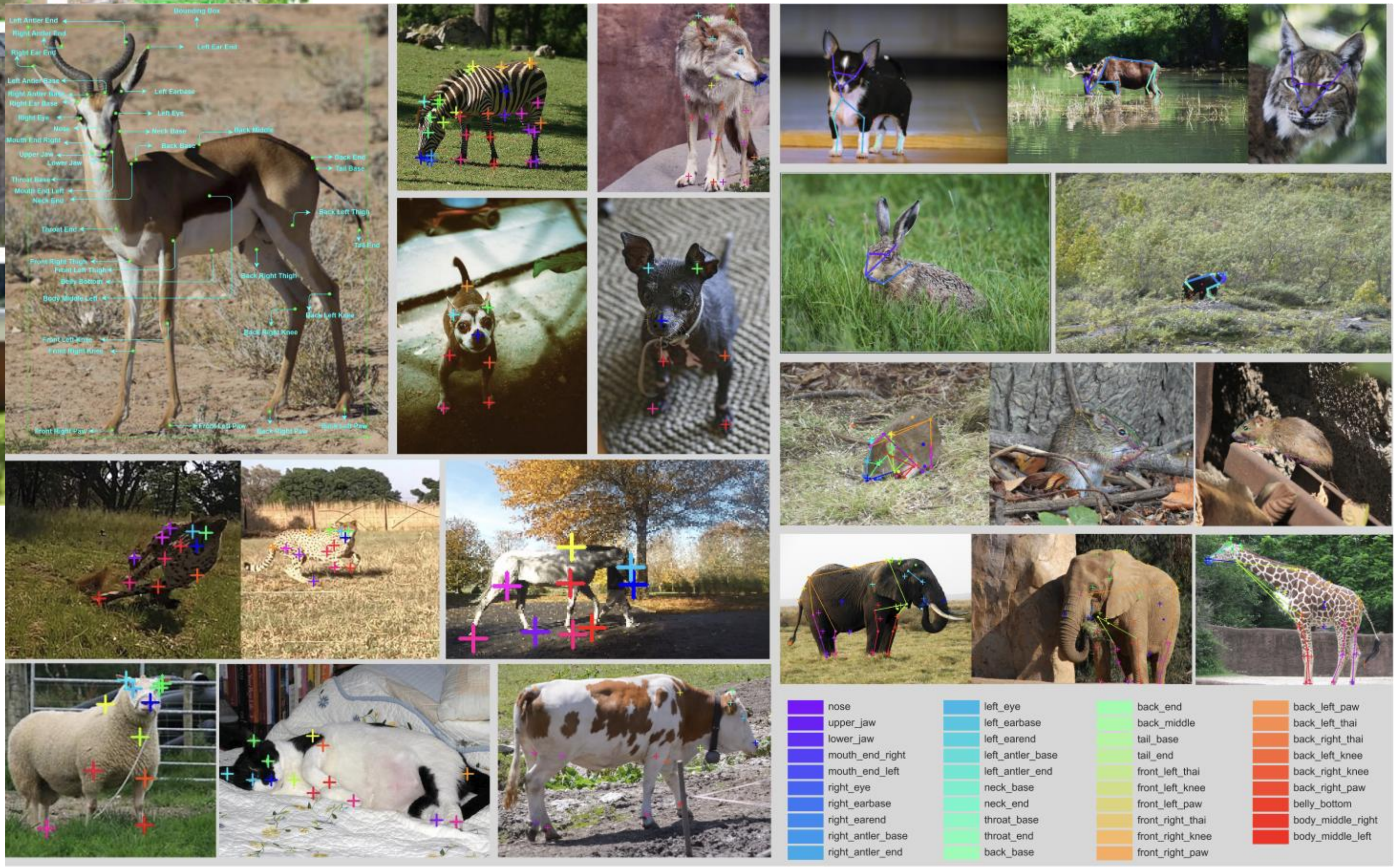
1. Creating a 3D project config file, deeplabcut.create_new_project_3d(‘3D_prog’,’scorer',num_cameras = 2)
2. Editing the config file skeleton & bodyparts based on 2D projects,
3. Extract calibration images (cameraName-##),

   - 30 - 50 pairs of images (1 per camera → pairs)
4. Calibrating cameras, - deeplabcut.calibrate_cameras(config3d, cbrow=8, cbcol=6, calibrate=True/False, alpha=0.9)
5. Undistorting, deeplabcut.check_undistortion(config_path3d, cbrow=8, cbcol=6)
6. Triangulating videos, deeplabcut.triangulate(config_path3d, video_path, videotype='.mp4', filterpredictions=True, save_as_csv=True)
7. Creating labeled video, deeplabcut.create_labeled_video_3d(config_path3d, [video_path], videotype='.mp4', view=[120, 270], xlim=[-250, 250], ylim=[-250, 250], zlim=[-250, 250])

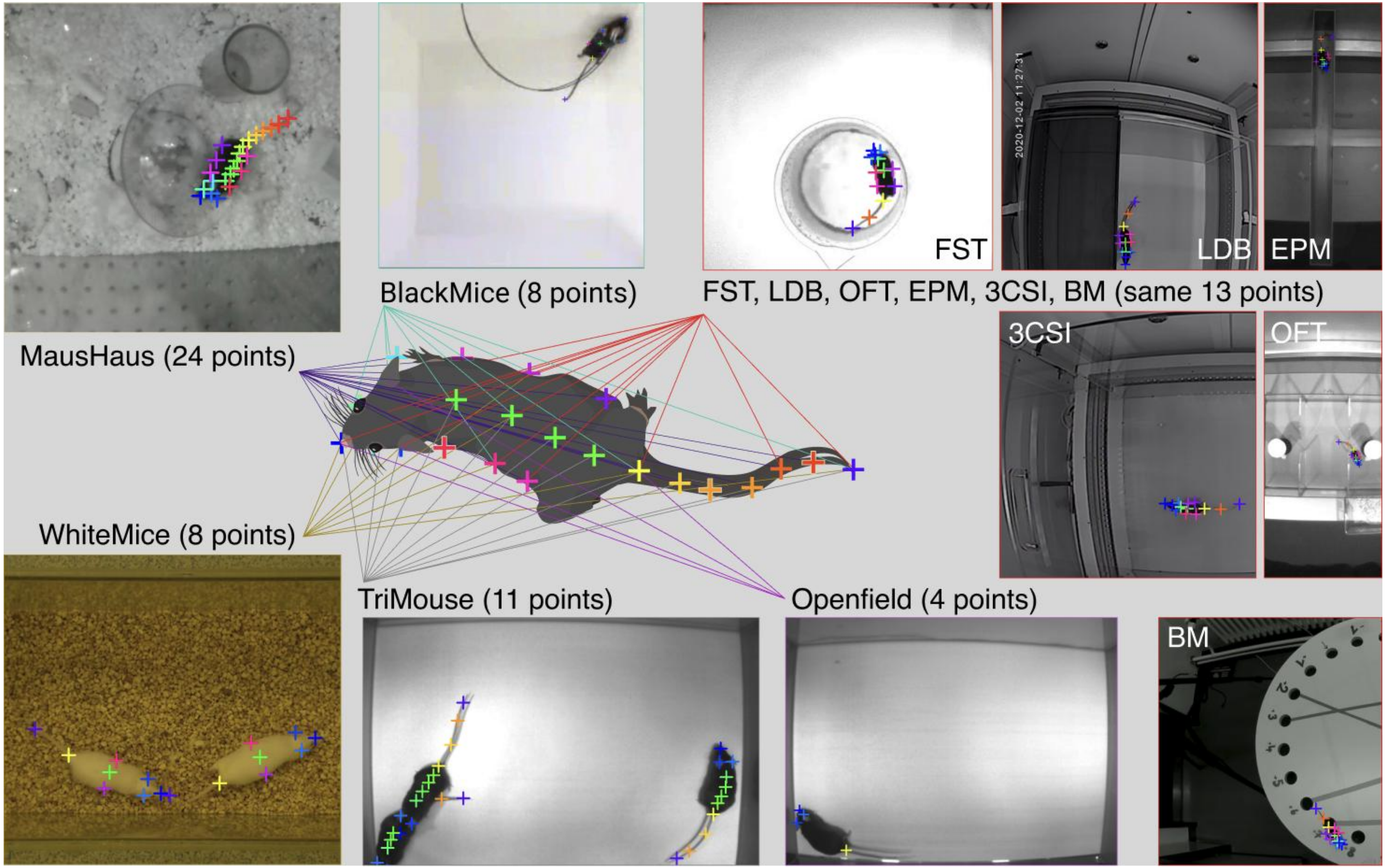
Model Zoo

Ye et al. (2024). Nature Communications.
https://deeplabcut.github.io/DeepLabCut/docs/ModelZoo.html


a
39 keypoint predictions with a single SA-Quadruped model across species

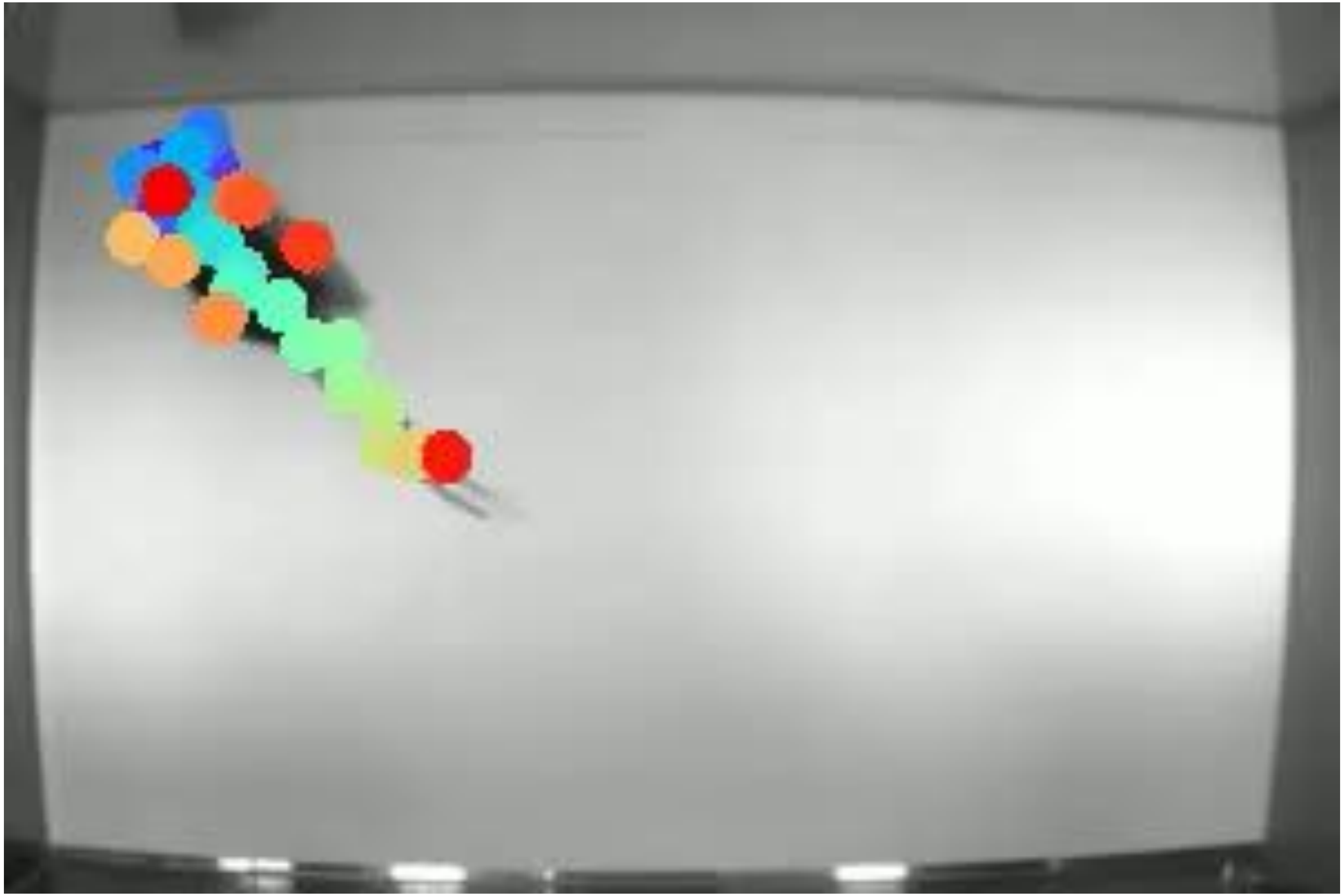


Model Zoo

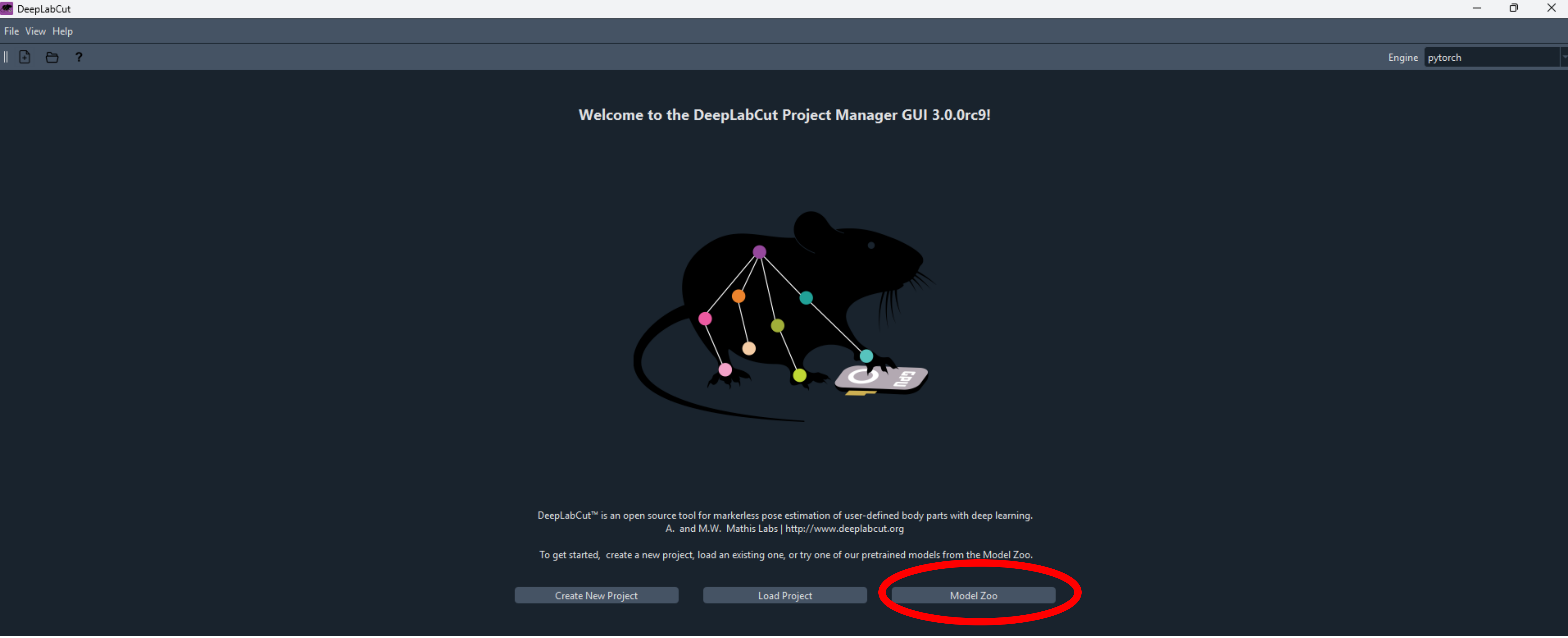


Model Zoo

https://github.com/DeepLabCut/DeepLabCut/blob/main/examples/COLAB/COLAB_DEMO_SuperAnimal.ipynb



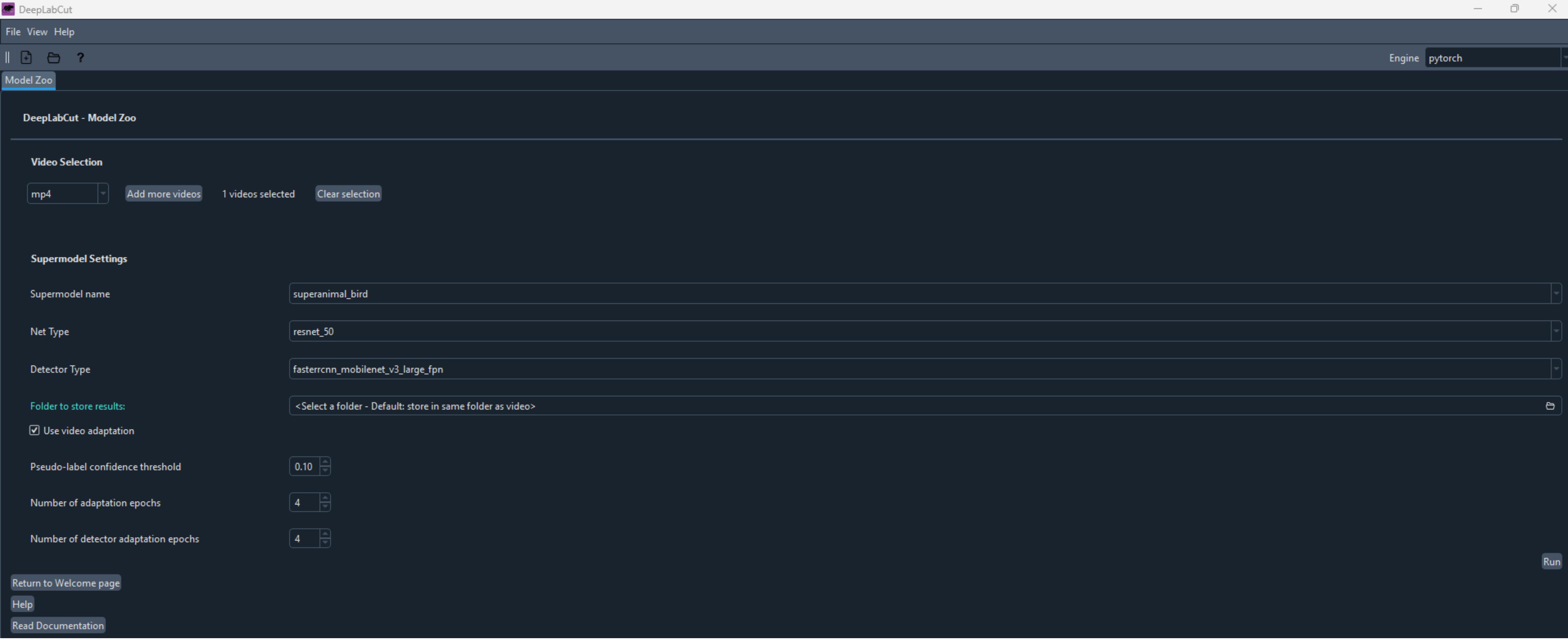
Model Zoo GUI



Welcome to the DeepLabCut Project Manager GUI 3.0.0rc9!

DeepLabCut™ is an open source tool for markerless pose estimation of user-defined body parts with deep learning.
A. and M.W. Mathis Labs | http://www.deeplabcut.org

To get started, create a new project, load an existing one, or try one of our pretrained models from the Model Zoo.



DeepLabCut - Model Zoo

Video Selection

mp4 | Add more videos | 1 videos selected | Clear selection

Supermodel Settings

Supermodel name: superanimal_bird

Net Type: resnet_50

Detector Type: fasterrcnn_mobilenet_v3_large_fpn

Folder to store results: <Select a folder - Default: store in same folder as video>

☑ Use video adaptation

Pseudo-label confidence threshold: 0.10

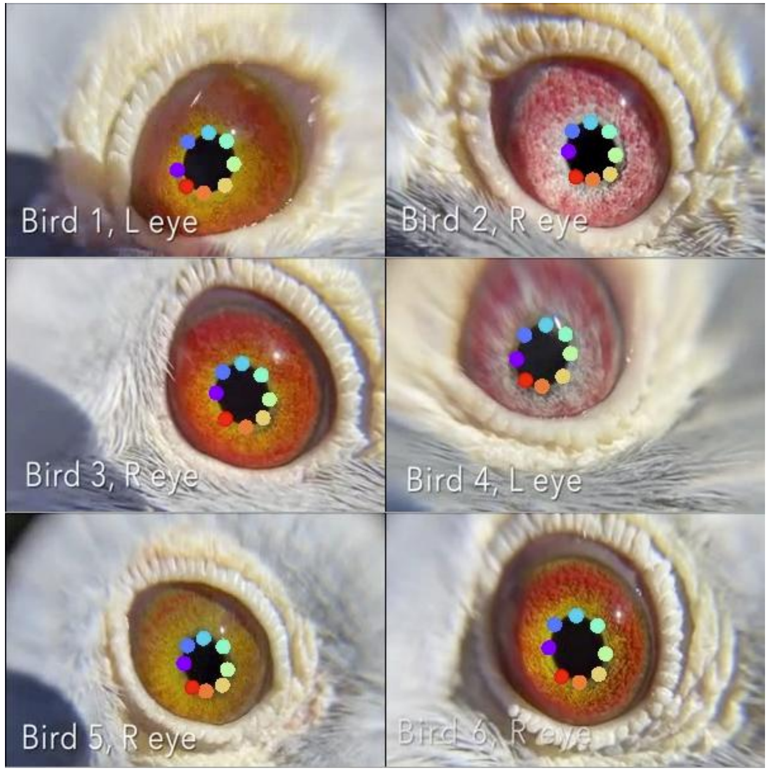
Number of adaptation epochs: 4

Number of detector adaptation epochs: 4

Try DLC model zoo!

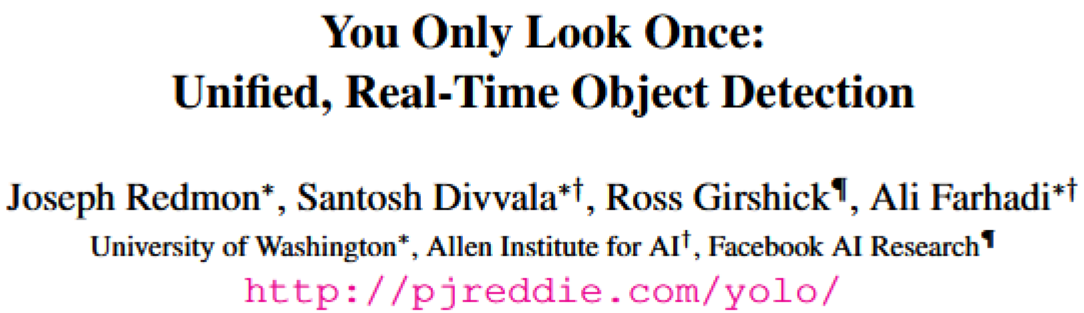
On their website: https://contrib.deeplabcut.org/

HuggyFace app: https://huggingface.co/DeepLabCut/DeepLabCutModelZoo-SuperAnimal-Bird/tree/main



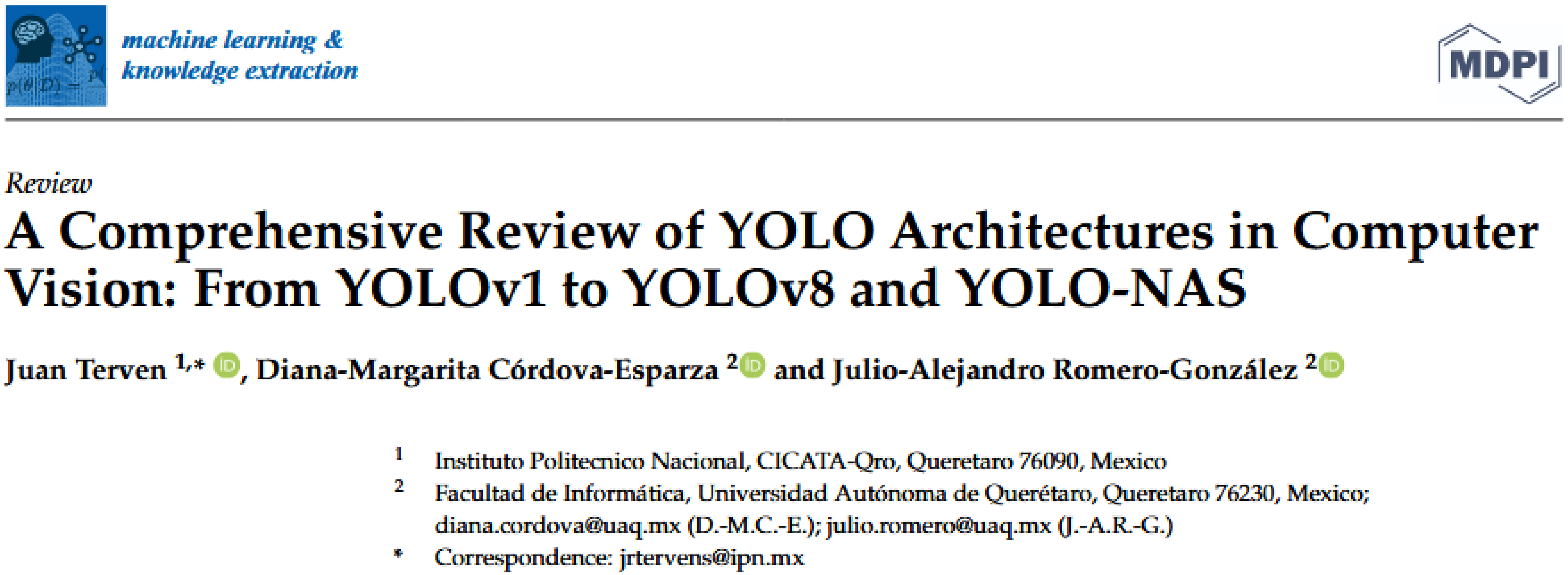
https://deeplabcut.github.io/DeepLabCut/docs/recipes/UsingModelZooPupil.html

You Only Look Once (YOLO)



You Only Look Once: Unified, Real-Time Object Detection

Joseph Redmon[*], Santosh Divvala[*†], Ross Girshick[¶], Ali Farhadi[*†]
University of Washington[*], Allen Institute for AI[†], Facebook AI Research[¶]
http://pjreddie.com/yolo/



machine learning & knowledge extraction

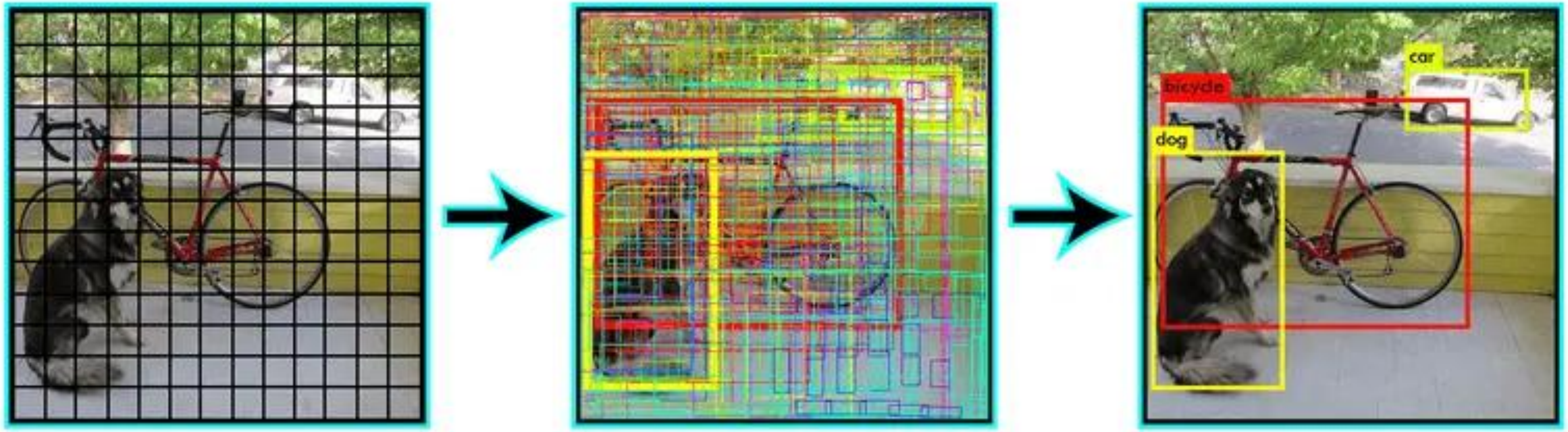
MDPI

Review

A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS

Juan Terven [1,*], Diana-Margarita Córdova-Esparza [2] and Julio-Alejandro Romero-González [2]

[1] Instituto Politecnico Nacional, CICATA-Qro, Queretaro 76090, Mexico
[2] Facultad de Informática, Universidad Autónoma de Querétaro, Queretaro 76230, Mexico; diana.cordova@uaq.mx (D.-M.C.-E.); julio.romero@uaq.mx (J.-A.R.-G.)
[*] Correspondence: jrtervens@ipn.mx



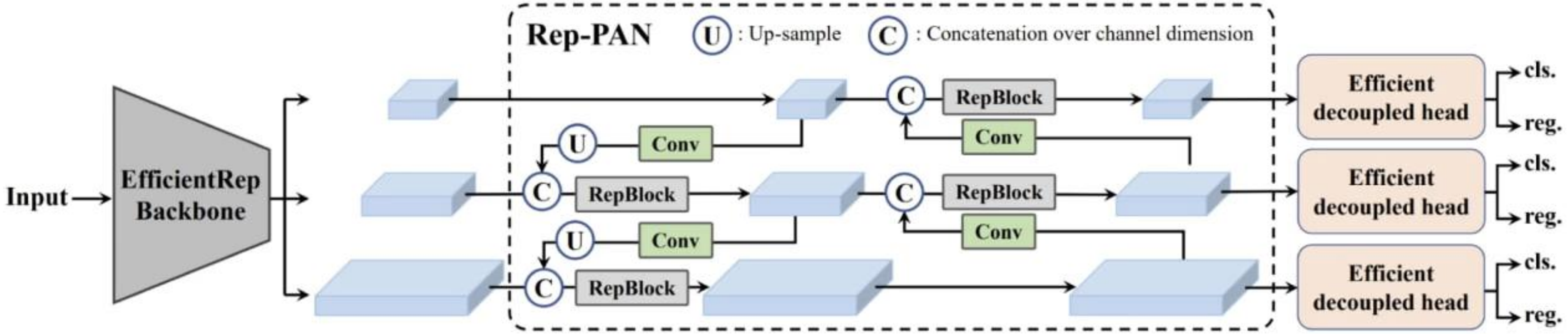


Figure 2: The YOLOv6 framework (N and S are shown). Note for M/L, RepBlocks is replaced with CSPStackRep.

Li, et al. (2022). ArXiv.

YOLOv8 commands

1. Creating a YOLO environment,
2. Install ultralytics, pip intall ultralytics
3. Install pip dependency, including torch, etc.
4. Install pre-trained YOLOv8 model weights, https://github.com/ultralytics/ultralytics
5. Creating a .yaml file to define labels, locations of train & test sets,
6. Training the detection/recognition model, yolo task=detect mode=train epochs=100 data=.yaml model=yolov8m.pt
7. Running the prediction based on detection/recognition model, yolo task=detect mode=predict model=best.pt show=True source=clip.mp4/image.jpg save_txt=True