



DLC AI Residency 2025 Buenos Aires Hub

Cronograma Tentativo

14:30	Presentación	Preguntas
15:00	Intro ML - DLC	Jornada <i>Hands-on</i>
15:30		
16:00	Coffe Break	Coffe Break
16:30		
17:00	Instalación	
17:30	1eros pasos + Labeling	
18:00	Cómo usar DLC	Más allá del trackeo
18:30	Intro proyectos	Cierre
19:00		

Recursos

Cajal course book

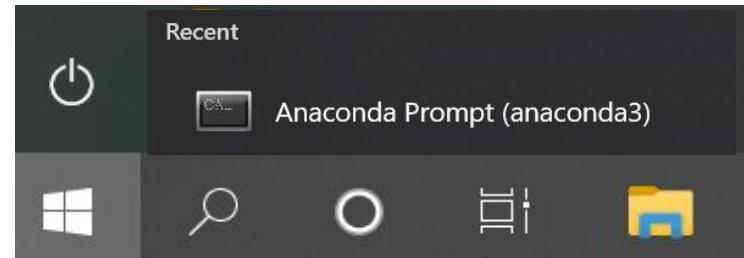
DeepLabCut book

DeepLabCut GitHub (demos, examples)

Papers, en especial DLC Protocol

Rápido: ¿cómo abro DEEPLABCUT? (GUI, Windows)

1) Busco Anaconda Prompt en el menú



2) Escribo **activate DEEPLABCUT** y toco enter

```
(base) C:\Users\Ceci>activate DEEPLABCUT  
(DEEPLABCUT) C:\Users\Ceci>
```

3) Escribo **python -m deeplabcut** y enter

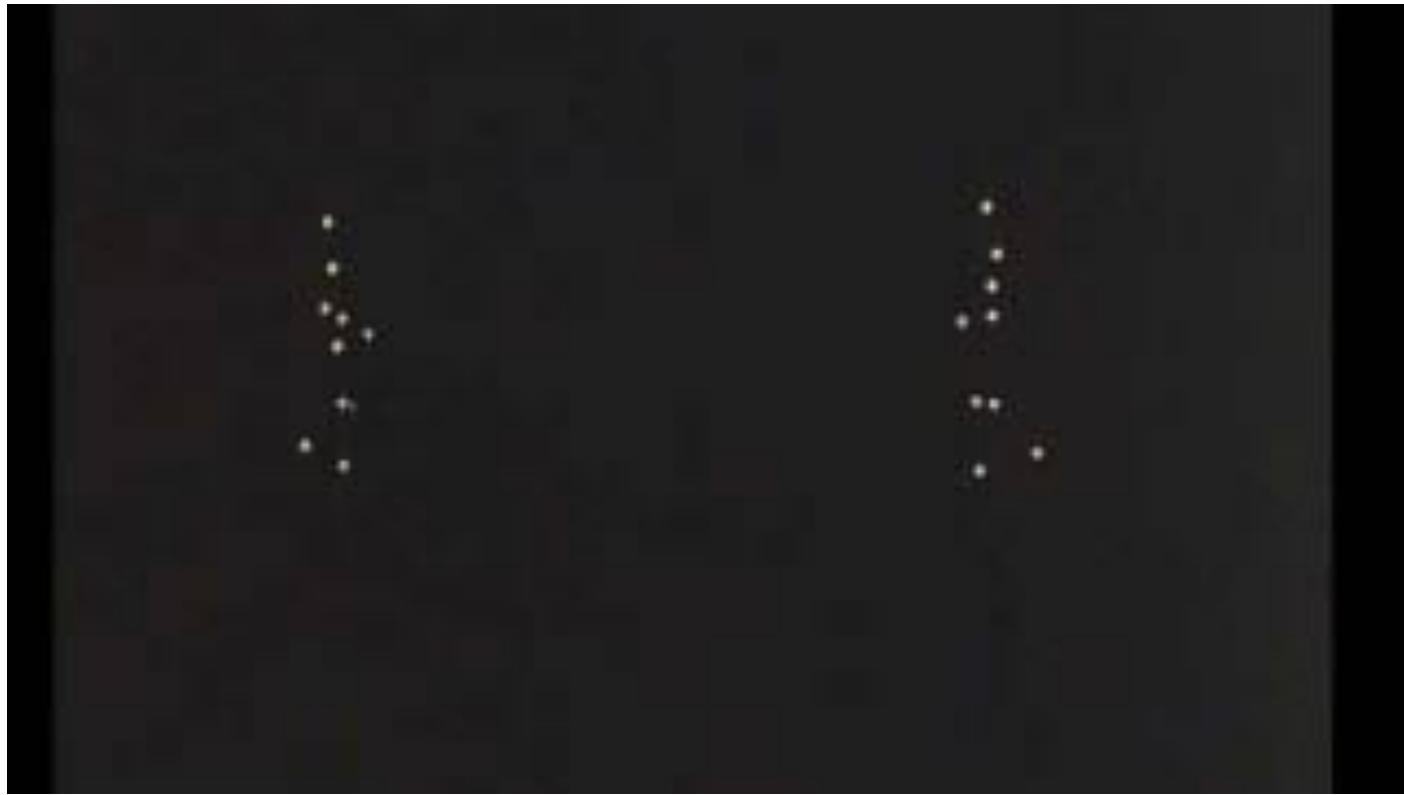
```
■ Anaconda Prompt (anaconda3) - "C:\Users\Labo112\anaconda3\condabin\conda.bat" activate DEEPLABCUT  
(base) C:\Users\Labo112>conda activate DEEPLABCUT  
(DEEPLABCUT) C:\Users\Labo112>python -m deeplabcut
```

4) Se abre la GUI :)

¿Cómo instalar DLC en las compus del O+Inf? (ubuntu) 2023

- 1) source activate base
- 2) conda create --name dlc python=3.8 (sino no anda wxpython)
- 3) conda activate dlc
- 4) pip install "deeplabcut[gui]"
- 5) python -m deeplabcut



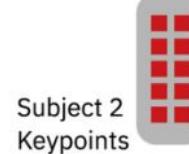
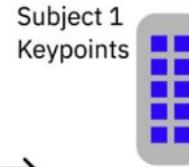


Estimación de la Posición

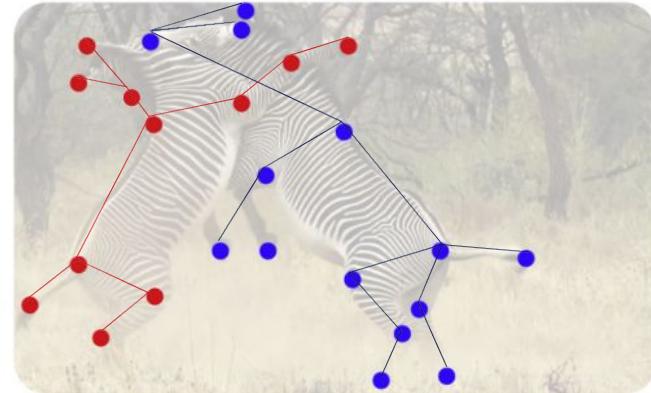
Pixel Representation



Pose Estimation
Algorithm



Keypoint Representation



1887

1978

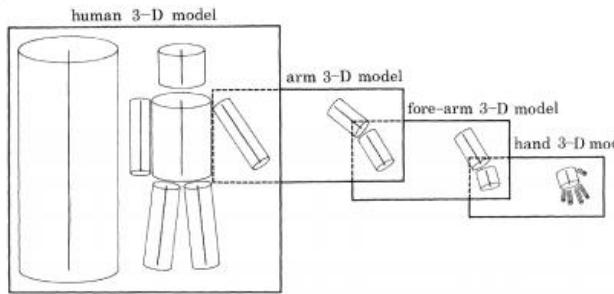
2010

2014

2018

Muybridge

David Marr

Black & Donoghue
(marker based)**Deep learning to
human pose
(Markerless)**Markerless pose
to animals**A****B**

¿Qué herramientas usan?

A Traditional Methods



Lighthouse
Tracking



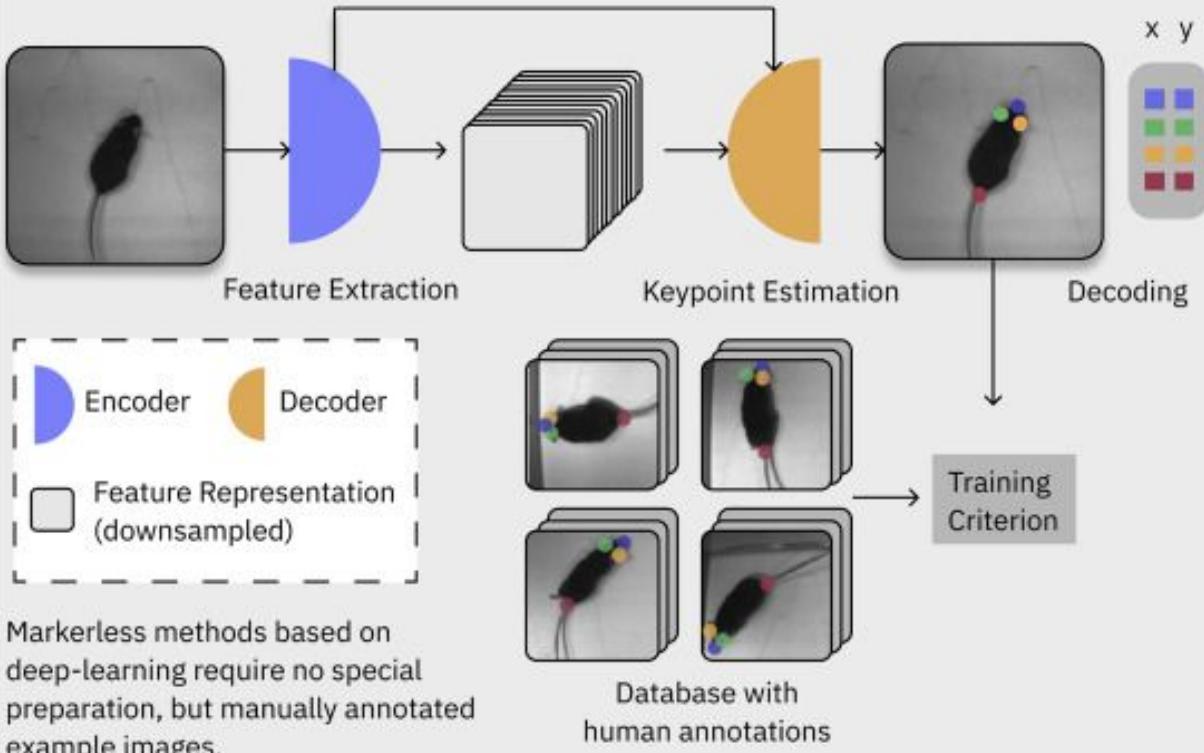
IMU-based
Tracking



Color-based
Tracking

Traditional methods work *ad-hoc*: subjects need to be prepared, but no annotation is needed.

B Markerless Pose Estimation



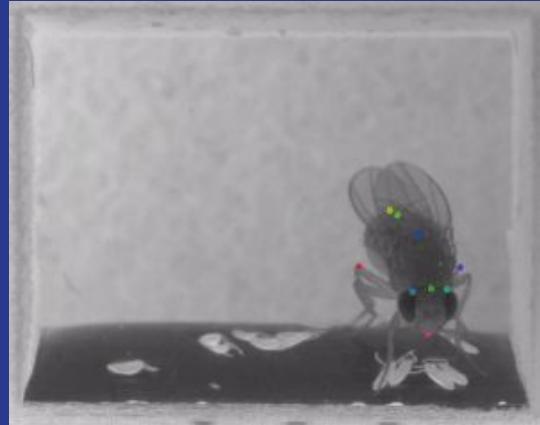
Desafíos en el laboratorio

Los animales tienen cuerpos muy diferentes

No es práctico que una persona etiquete más de 10.000 frames

Análisis de video en tiempo real rápido.

Seguimiento de múltiples animales



Aprovechando el aprendizaje por transferencia

deep neural networks

image → Predictor → pose

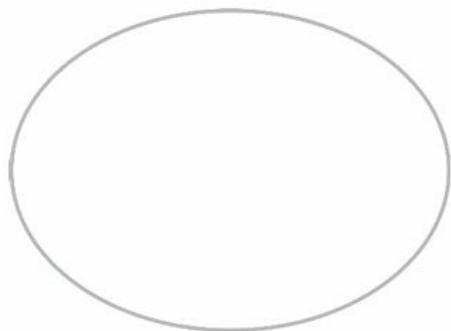


Algunas definiciones de Machine Learning

A field of study that gives computers the ability to learn without being explicitly programmed (Arthur Samuel)

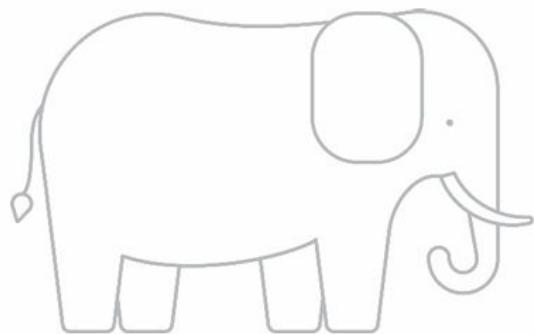
- **train set:** porción de datos que usamos para entrenar el modelo
- **test set:** porción de datos que usamos para evaluar el modelo
- **Supervisado =** datos con etiquetas
- **Overfitting:** el modelo no generaliza adecuadamente

Underfitted



El modelo pasa por alto los patrones subyacentes en el conjunto de entrenamiento.

Generalized



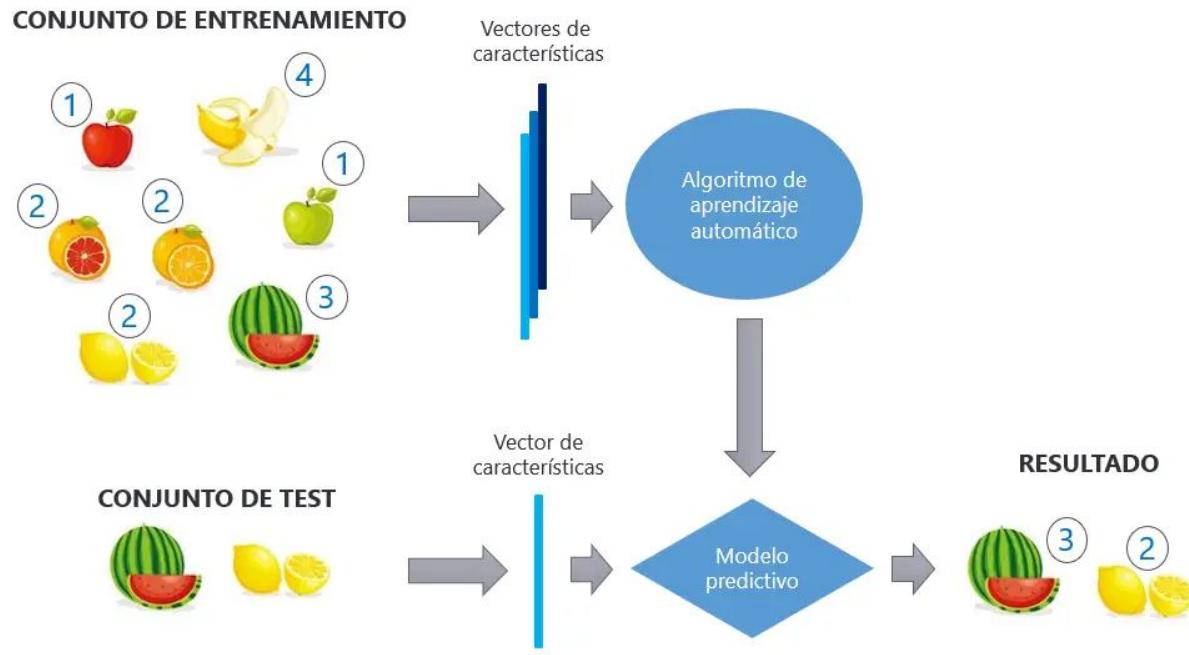
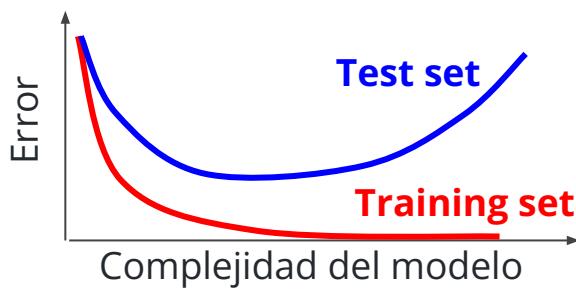
El modelo capta correlaciones en el conjunto de entrenamiento.

Overfitted



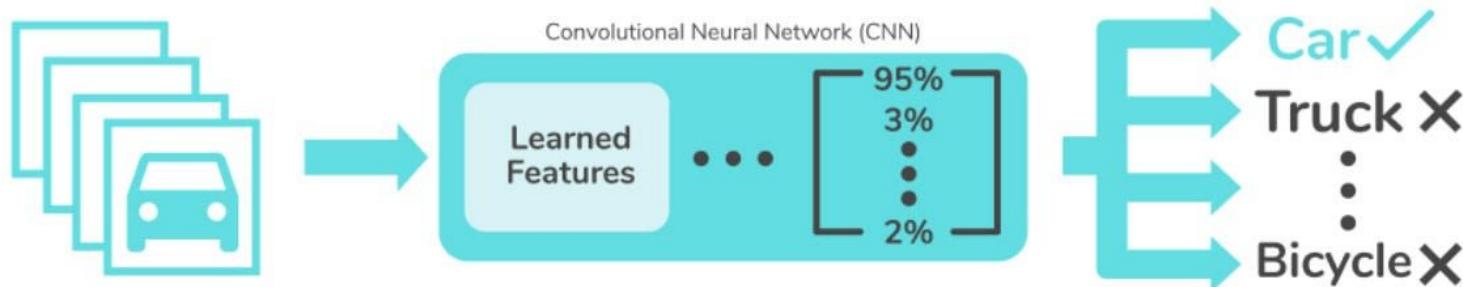
El modelo memoriza el conjunto de entrenamiento y no es bueno para predecir datos no vistos.

Algunas definiciones de Machine Learning



Aprovechando el aprendizaje por transferencia

Training from Scratch



Transfer Learning



Image
Classification



Object
Localization



Dense/Semantic
Segmentation

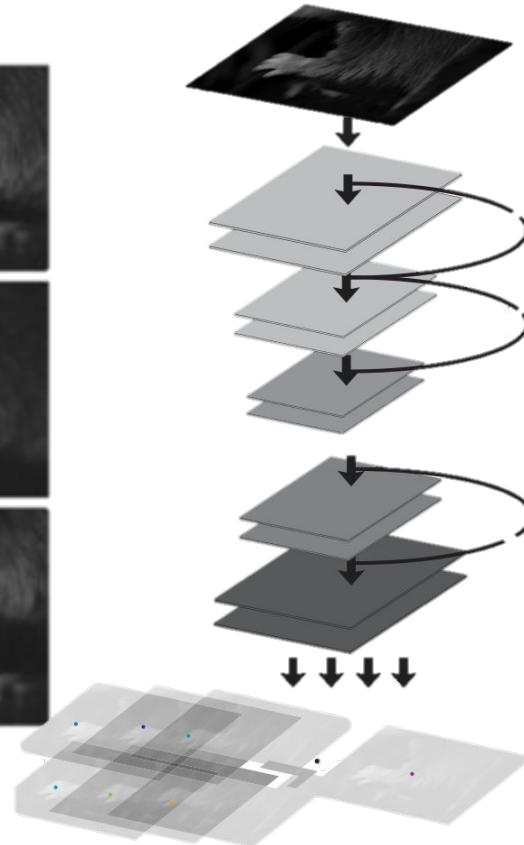
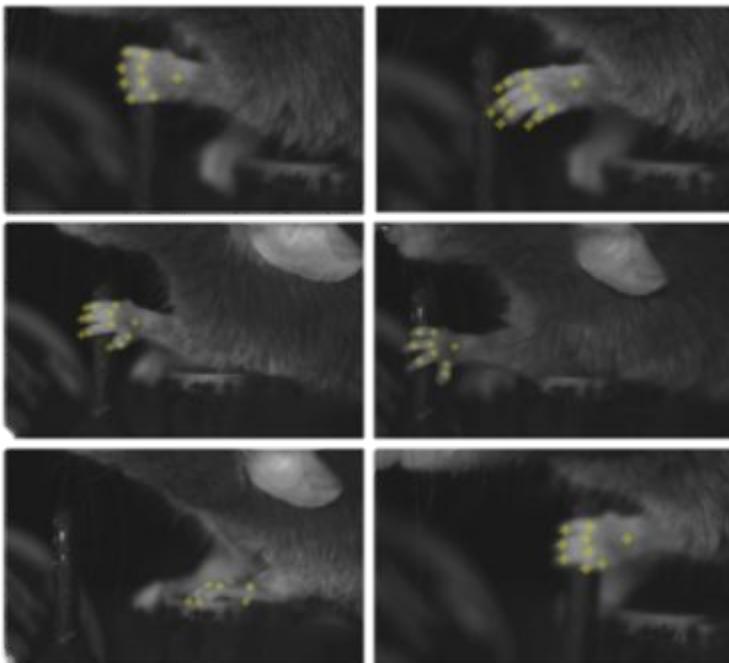


Cat

Cat

Cat

Midiendo el comportamiento con DeepLabCut

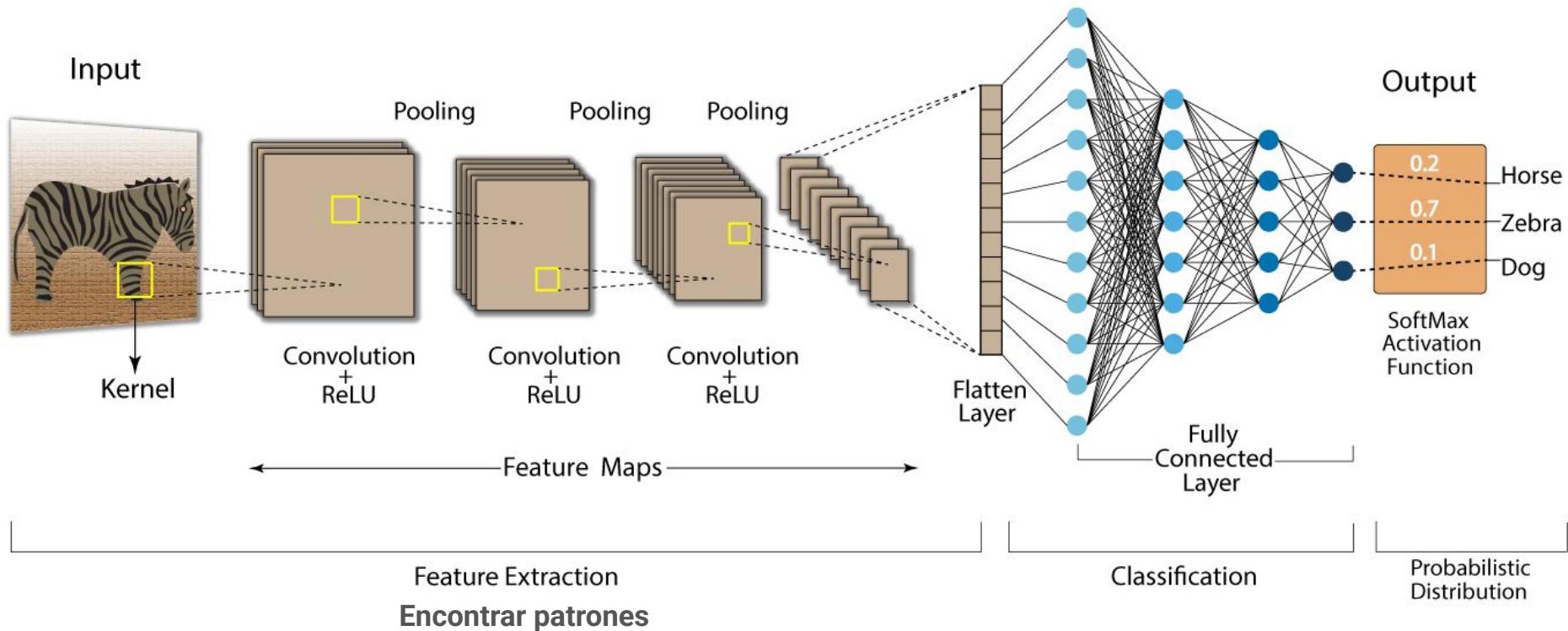


CNN is a neural network: an algorithm used to recognize patterns in data.

<https://poloclub.github.io/cnn-explainer/>



Convolution Neural Network (CNN)

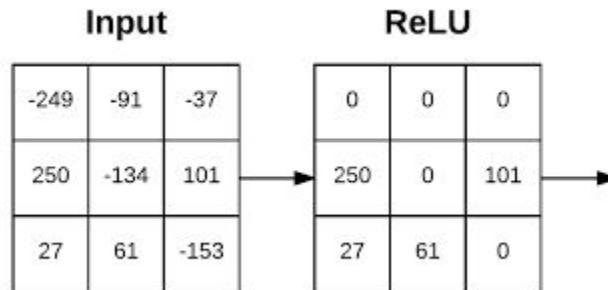


Pooling: gradually decreasing the spatial extent of the network

CNN - Convolución

- como pasar una "ventana" (**kernel**) pequeña por toda la imagen para detectar patrones locales. Ver: https://github.com/vdumoulin/conv_arithmetic
- Cuando el filtro se superpone a una parte de la imagen, se multiplican los valores de los píxeles por los del filtro y se suman, generando un número que indica qué tan presente está ese patrón en esa zona.

Por ejemplo, un filtro puede estar diseñado para detectar bordes verticales: si lo aplicamos sobre toda la imagen, el resultado será una nueva imagen que resalta justamente los bordes verticales.



POOLING

Max pooling

32	19
20	27

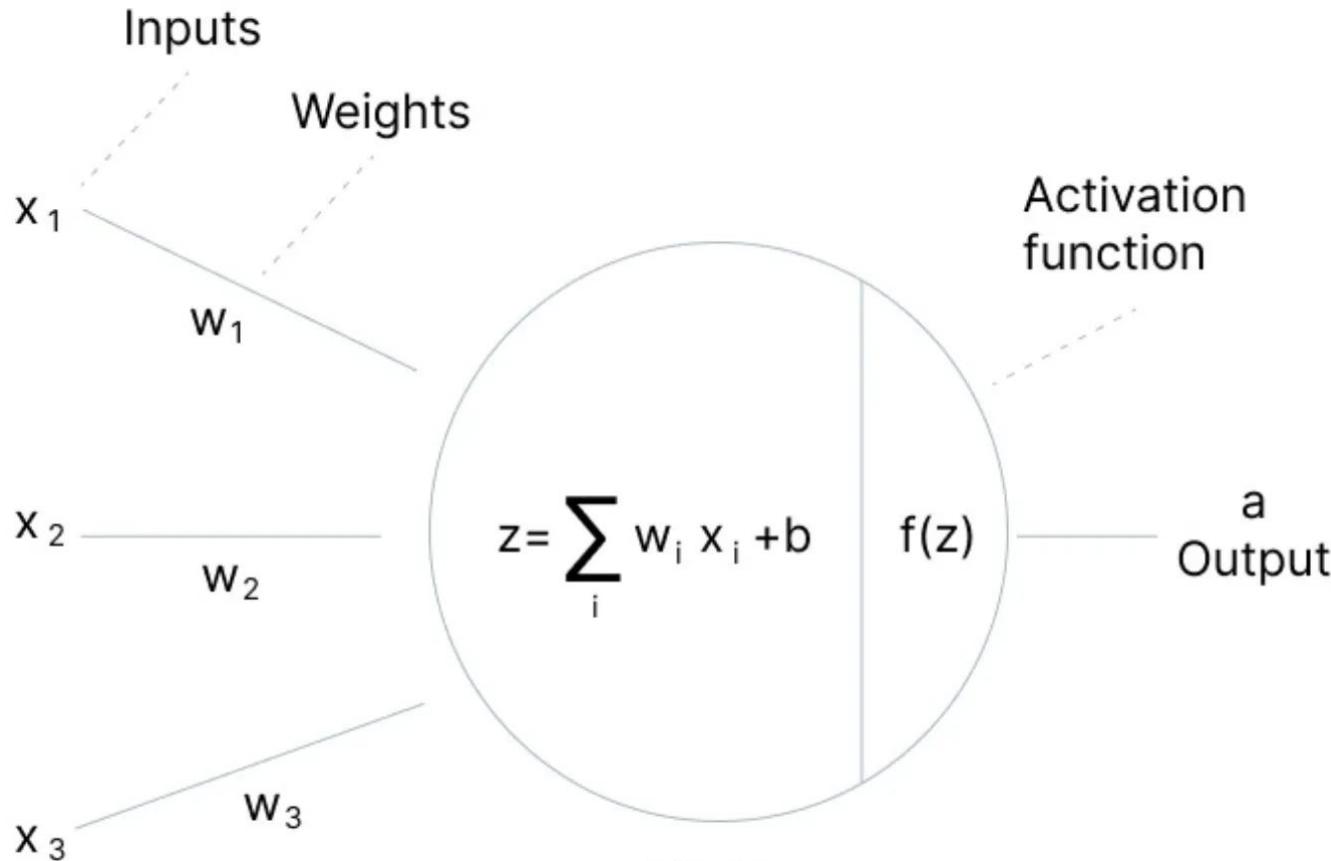


32	10	11	17
4	14	9	19
20	4	16	27
8	12	7	14

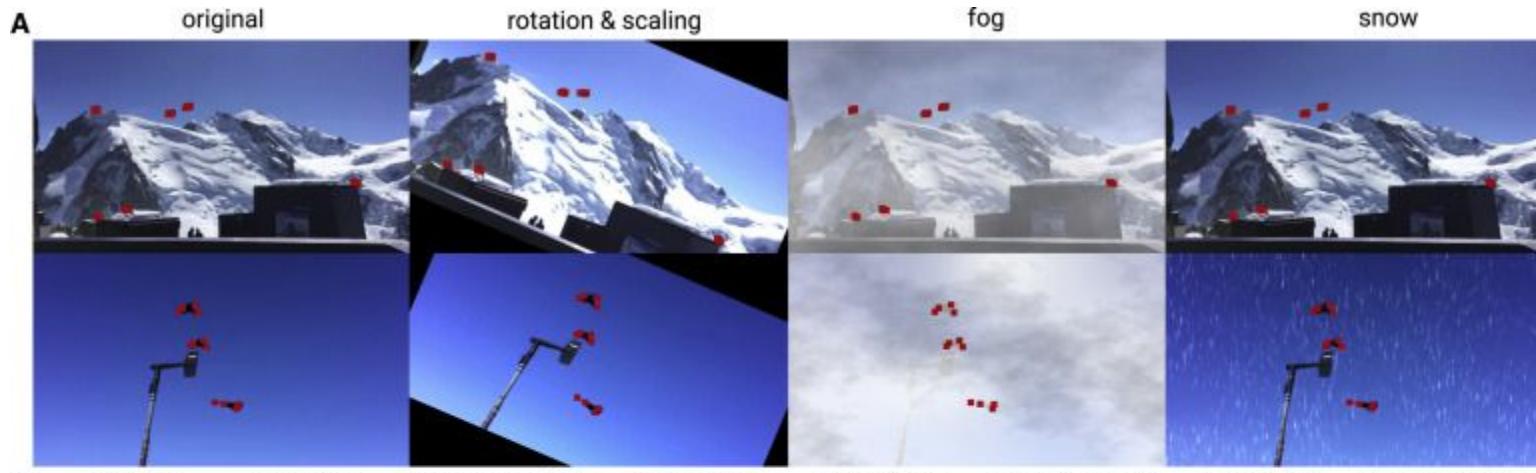
Average pooling

15	14
11	16

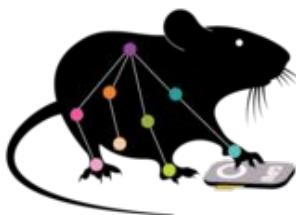
MLP: Multilayer perceptron



Data augmentation para el entrenamiento



Pipeline



DeepLabCut:

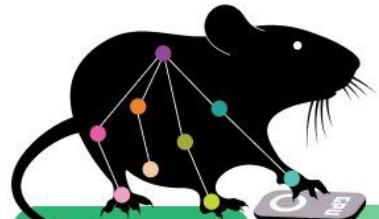
a software package for
animal pose estimation

Create a project,
extract frames, +
GUIs to label your data

Select + Train your
deep neural network

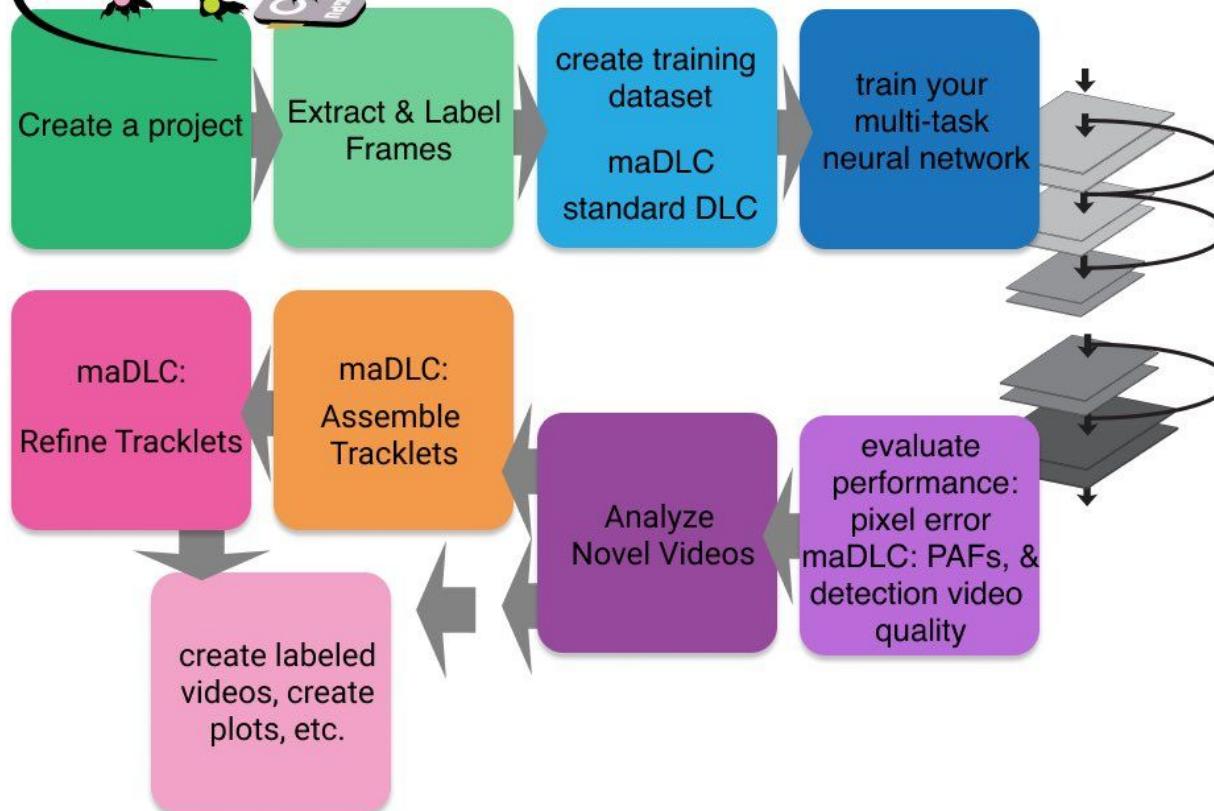
Evaluate network
performance
(active learning + GUIs
if improvement needed)

Run inference on
new videos,
create labeled videos,
+ plot your results!



DeepLabCut

workflow





Frames extracted from video
(any species can be used)

GUI: LABEL

- unique label 1
- unique label 2
- unique label 3

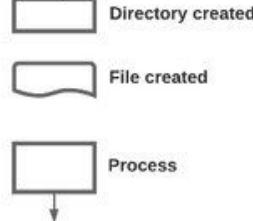
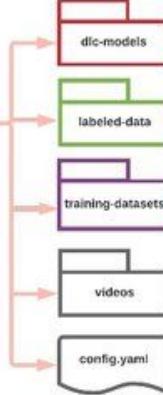


Create project

Extract frames

Label frames
(GUI)

Create training
datasets

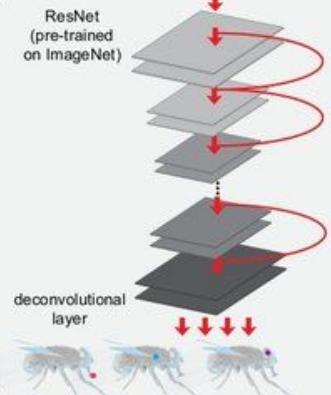


Directory created

File created

Process

Train Network



Train network

Evaluate network

Good Results

Analyze video

Stop

No

Yes

Need more
training data?

Analyze video

Merge datasets

Refine labels
(GUI)

Extract outlier
frames

GUI: REFINE

- unique label 1
- unique label 2
- unique label 3

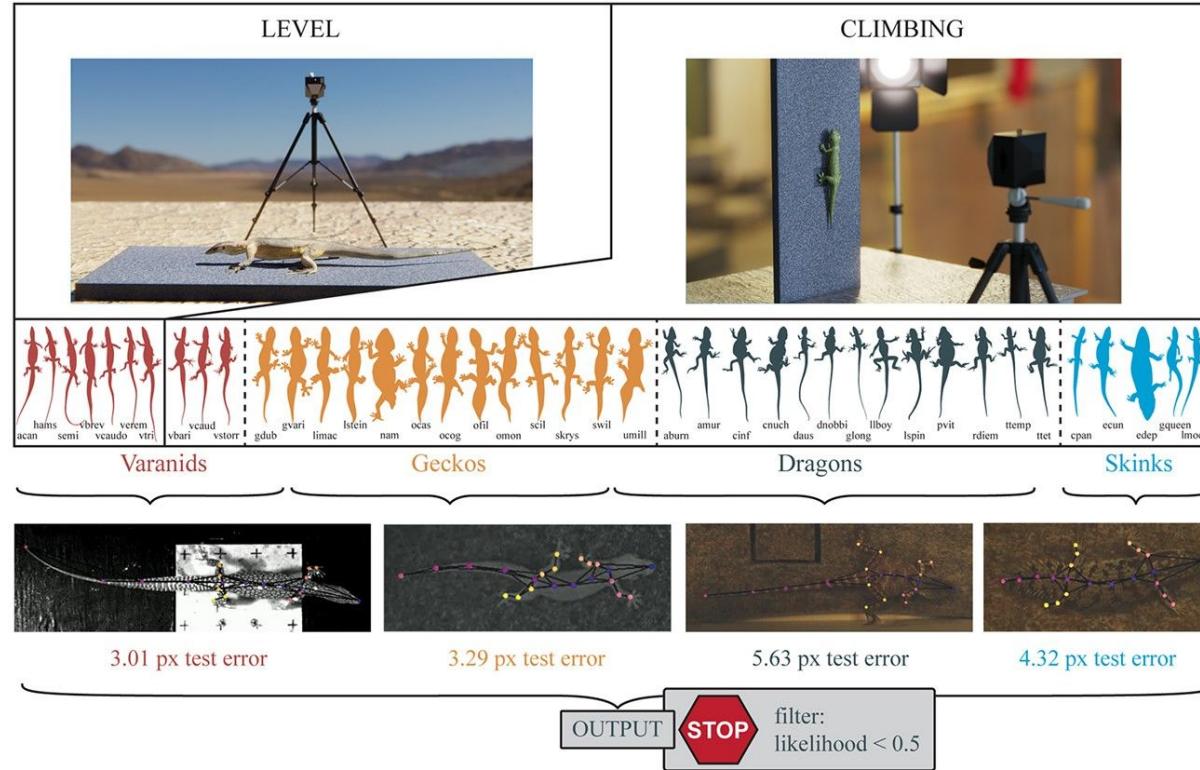


Analyze Novel Videos



Responder la pregunta científica empieza acá

A



B

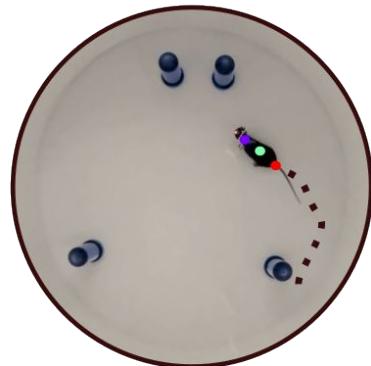
Relación entre el movimiento de la cola y la locomoción:

- El grado de **flexión de la cola** depende principalmente de la **longitud relativa de la cola**.
- cuánto se desvía del eje corporal depende principalmente de la **velocidad relativa y no de la longitud de la cola**.

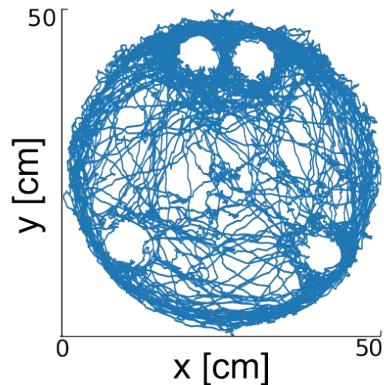
Cómo codifican objetos cercanos y lejanos?



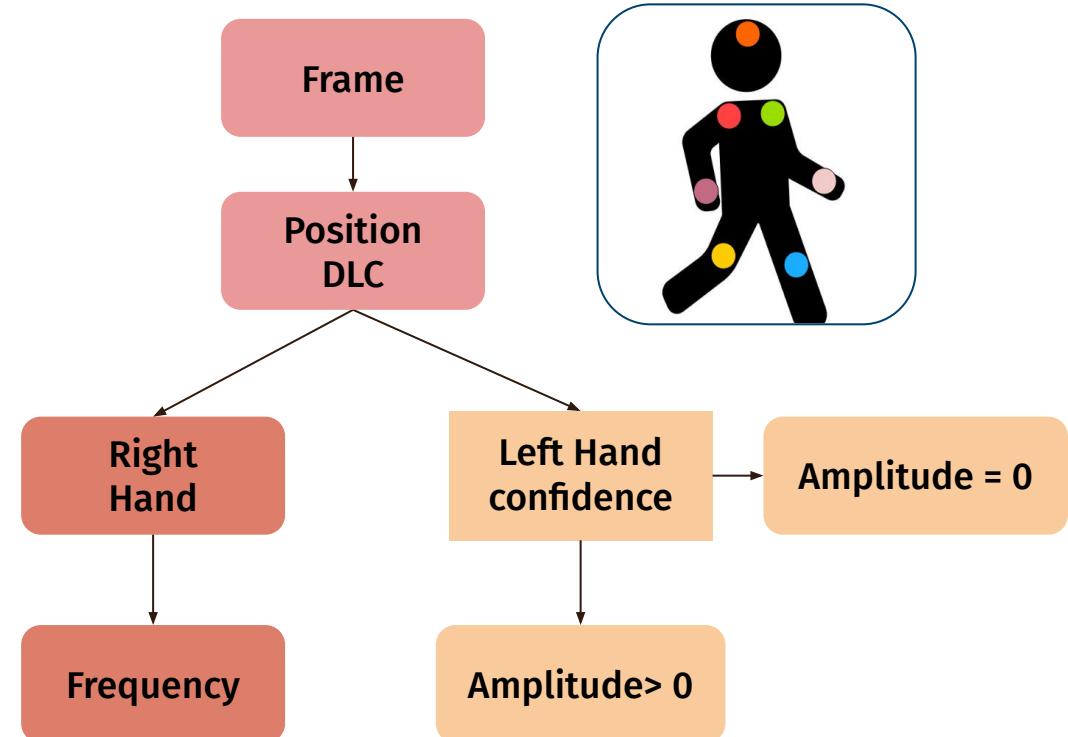
b



C Posición de la cabeza



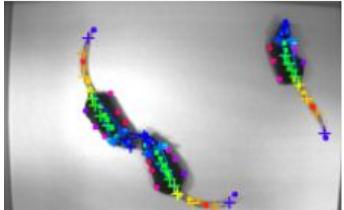
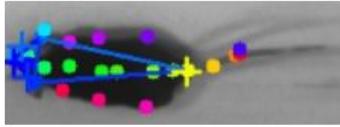
Therem/IAn



Se han entrenado muchísimos modelos sobre **datasets cerrados de animales**.

Pero ¿Podemos combinar todo ese conocimiento para construir **mejores modelos funcionales para estimación de poses animales?**

SuperAnimal- TopView Rodent (5K)



SuperAnimal- Quadruped (80K)

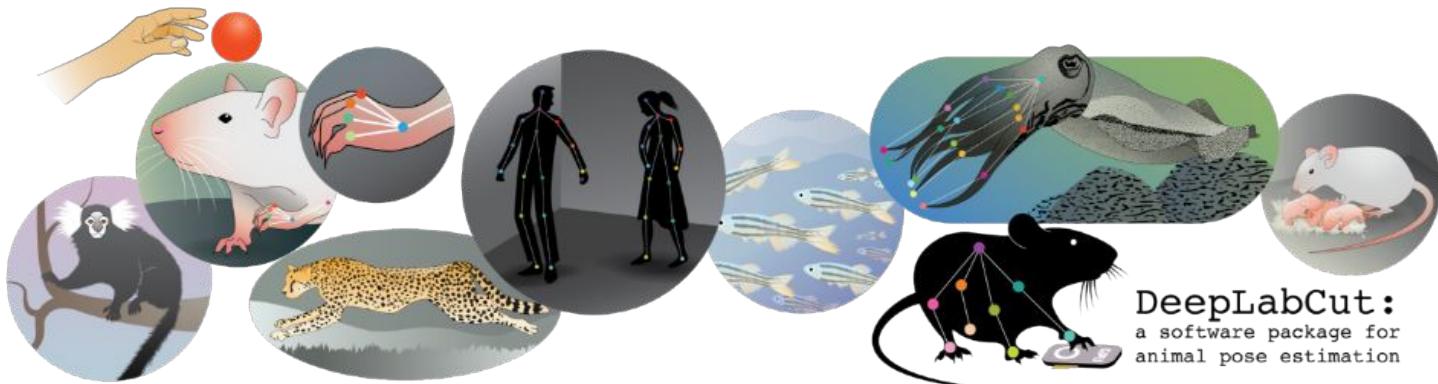


Typical Lab Setting
SuperAnimal TopView

26 keypoints

39 keypoints

ModelZoo



modelzoo.deeplabcut.org

DeepLabCut:
a software package for
animal pose estimation

The DeepLabCut ModelZoo website features a central navigation bar with three main sections: "Contribute models" (pink), "Annotate images" (teal), and "Test our models" (purple). Below the navigation bar is a logo for "DeepLabCut" with the subtitle "a software package for animal pose estimation". A descriptive text box at the bottom left explains the project's goal: "The DeepLabCut ModelZoo Contrib project is an initiative to gather animal pose estimation data in order to create robust models for the community." Another text box at the bottom right provides instructions for contributing: "Help us build these models by contributing your time (labeling) or get in touch if you have data you are willing to share! To read more about this project please see [DeepLabCut.org](#) and [ModelZoo.DeepLabCut.org](#)!"

- Ya no es necesario entrenar desde cero

⚠ Desafíos

- Los roedores y cuadrúpedos en vista superior no representan a todos los animales de la neurociencia.
- Todavía falta cubrir una gran diversidad de especies y comportamientos.

Proyectos

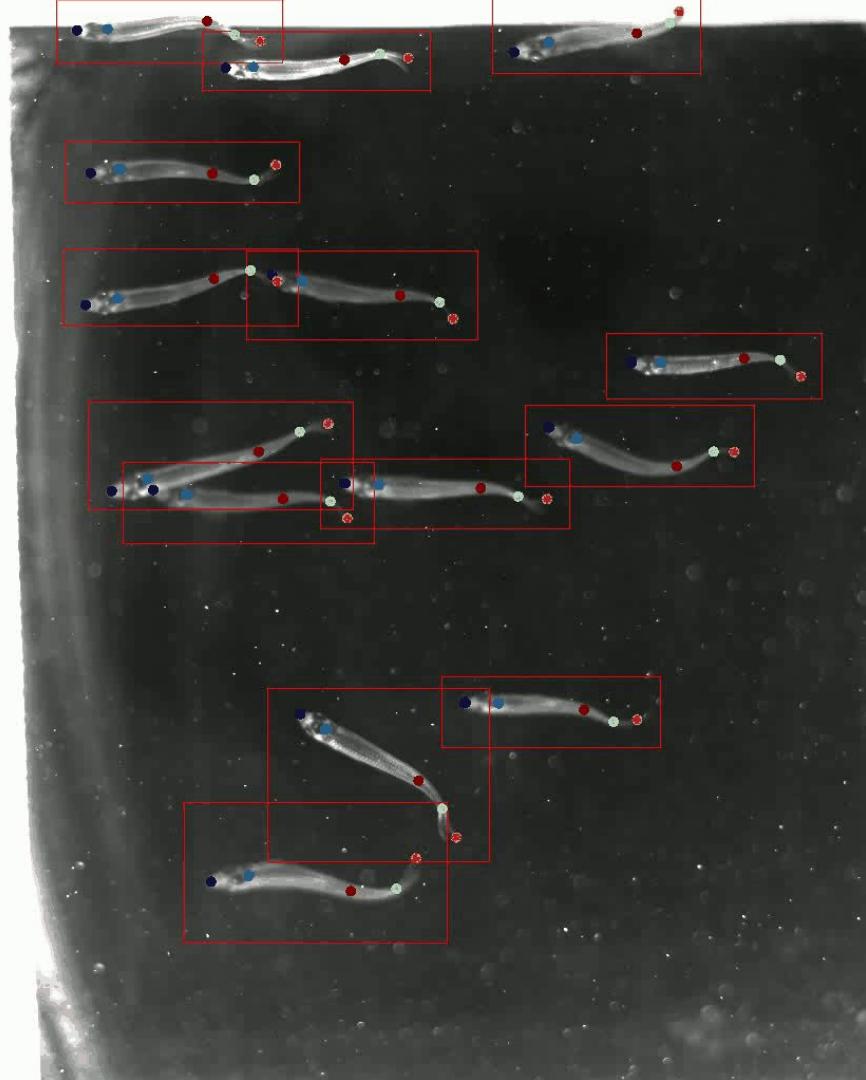
1. **Traé tus propios datos:** Si ya contás con videos o datos de comportamiento, podrás trabajar en analizarlos y optimizar tu pipeline con DeepLabCut.
2. **Director de orquesta:** Analizaremos un video de un director de orquesta. Desafío: *¿Podemos inferir el tempo?*

Posibles videos/links:

<https://www.youtube.com/watch?v=diwV2HGKerE>

<https://www.youtube.com/watch?v=MEeLI9-l63w>

3. **Datos + “CEBRA”:** Usar datos disponibles de un estudio con ratones y usar CEBRA para intentar inferir la posición de un ratón. Desafío: *trabajar con datos nuevos. ¿Podemos encontrar el espacio latente de las grid cells?*



Interactive Project Manager GUI

Acceder a la terminal

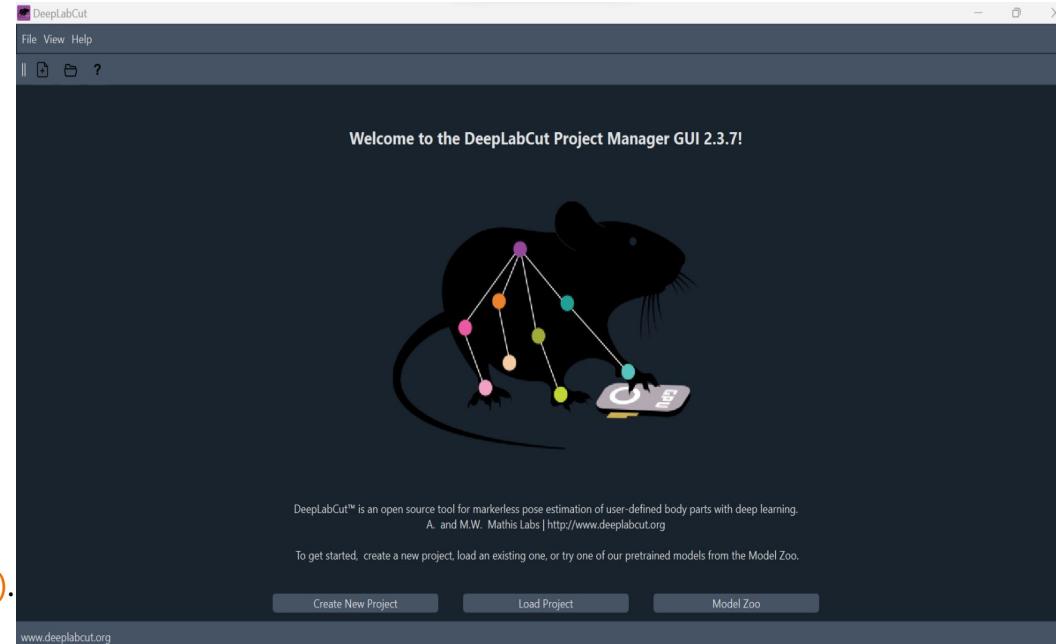
Posicionarse en el video y hacer click derecho: OPEN IN TERMINAL

Se abre la terminal y aparece la dire del video

Tipeo **conda activate DEEPLABCUT** (enter) y ahí accedés a DeepLabCut.

Luego tipeo: **python -m deeplabcut** (enter) y se abre el “GUI” (graphical user interface) que es la interfaz gráfica, que me permite usarlo sin tipear código.

```
Anaconda Prompt (anaconda3) - "C:\Users\Labo112\anaconda3\condabin\conda.bat" activate DEEPLABCUT
(base) C:\Users\Labo112>conda activate DEEPLABCUT
(DEEPLABCUT) C:\Users\Labo112>python -m deeplabcut
```



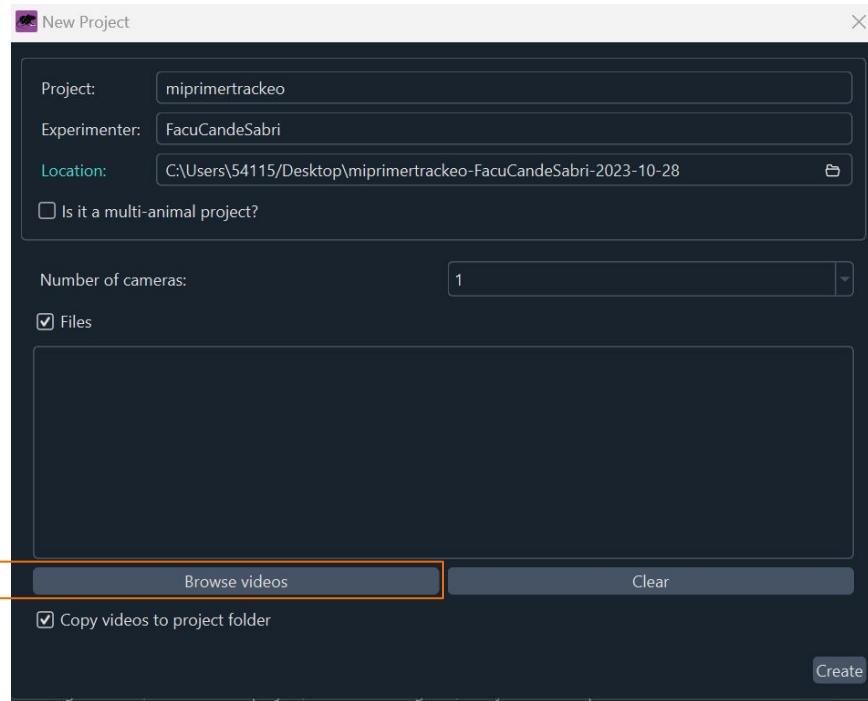
selecciono la opción “Create new project” (1).

Manage Project

Coloco el **nombre del proyecto** y de los experimentadores y luego presiono “**Load videos**” (3) y cargo los videos a analizar.

En las opciones selecciono “**copy the videos**” (4), para que me quede una copia de los videos (los originales y los procesados).

Y le doy “**OK**” (5).



Proyecto creado

Veo en la carpeta desde donde corrí la GUI que se creó una carpeta para el proyecto llamada

ProjectName-Experimenter-Date

Y tiene las siguientes subcarpetas:

-  **dlc-models**
-  **labeled-data**
-  **training-datasets**
-  **videos**
-  **config.yaml**

dlc-models: This directory contains the subdirectories *test* and *train*, each of which holds the meta information with regard to the parameters of the feature detectors in configuration files. The configuration files are YAML files, a common human-readable data serialization language. These files can be opened and edited with standard text editors. The subdirectory *train* will store checkpoints (called snapshots in TensorFlow) during training of the model. These snapshots allow the user to reload the trained model without re-training it, or to pick-up training from a particular saved checkpoint, in case the training was interrupted.

labeled-data: This directory will store the frames used to create the training dataset. Frames from different videos are stored in separate subdirectories. Each frame has a filename related to the temporal index within the corresponding video, which allows the user to trace every frame back to its origin.

training-datasets: This directory will contain the training dataset used to train the network and metadata, which contains information about how the training dataset was created.

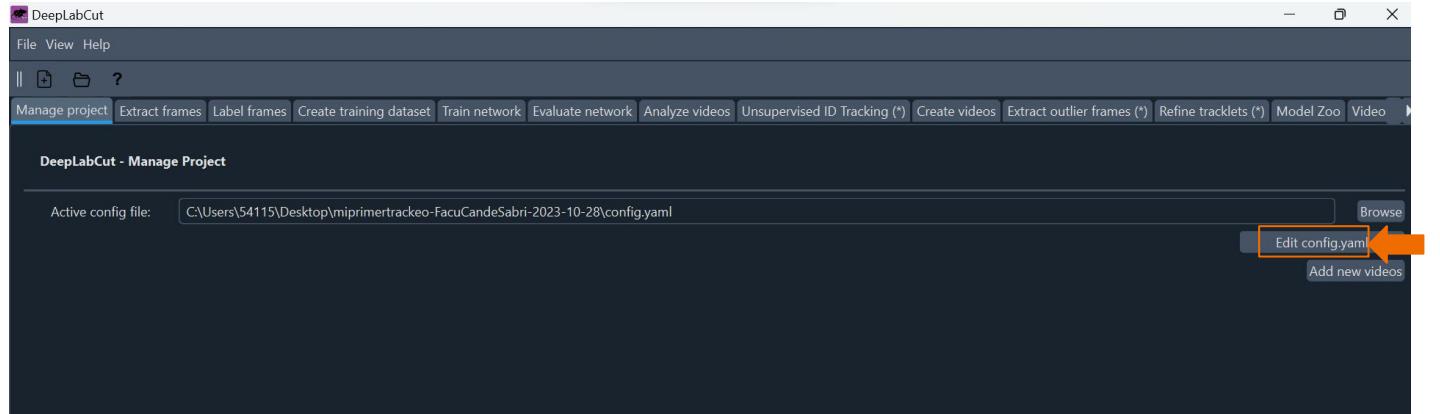
videos: Directory of video links or videos. When **copy_videos** is set to **False**, this directory contains symbolic links to the videos. If it is set to **True** then the videos will be copied to this directory. The default

Edit config file

Una vez que se crea el proyecto se habilitan un montón de solapas nuevas y en esta solapa en particular, se habilita la opción “Edit config file” (1).

Cuando hago clic, se abre un archivo de texto que yo voy a editar:

En donde dice “body parts” (2) puedo nombrar a los puntos del cuerpo del ratón que quiero usar (sean 3, 4, 5, etc.). En este caso puedo poner 4 y nombrarlos como **nose**, **rightear**, **leftear**, **tailbase** por ej.



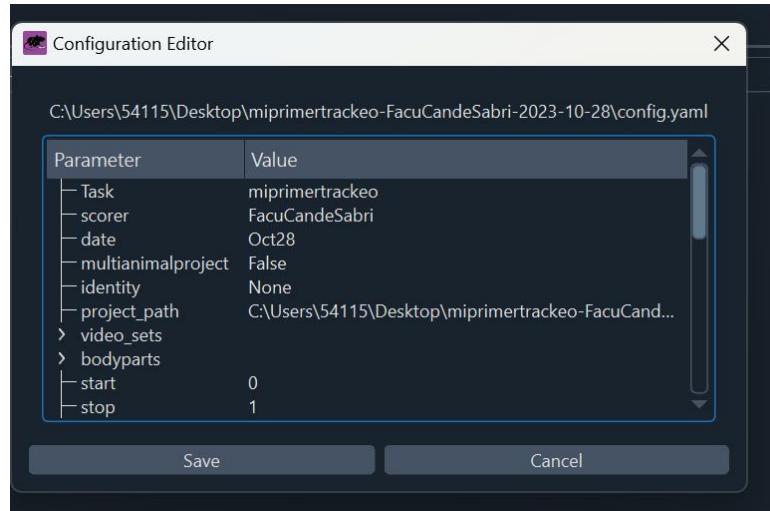
Edit config file

Una vez que se crea el proyecto se habilitan un montón de solapas nuevas y en esta solapa en particular, se habilita la opción “Edit config file” (1).

Cuando hago clic, se abre un archivo de texto que yo voy a editar:

En donde dice “body parts” (2) puedo nombrar a los puntos del cuerpo del ratón que quiero usar (sean 3, 4, 5, etc.).

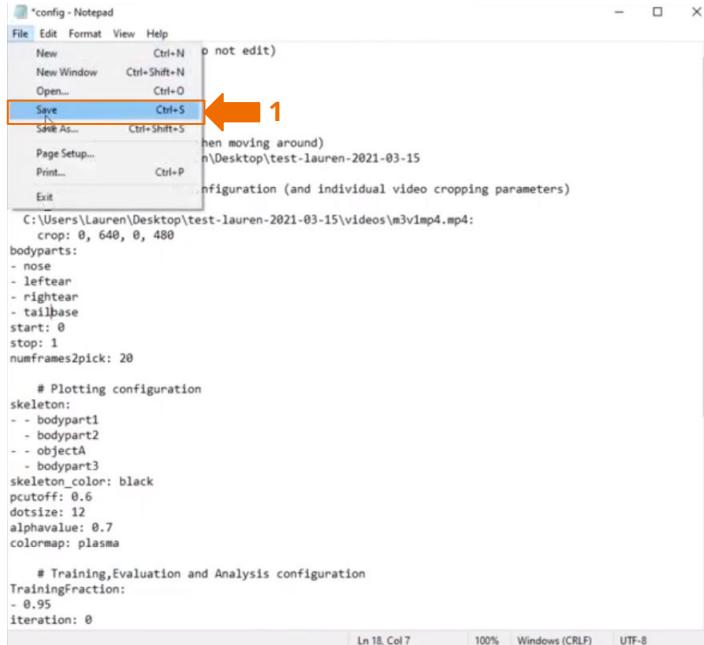
En este caso puedo poner 4 y nombrarlos como **nose, rightear, leftear, tailbase** por ej.



Una vez que cambie los nombres de mis puntos, voy a “File” y hago click en “Save” (1) y luego cierro el archivo de texto.

Edit config file

Una vez que cambie los nombres de mis puntos, voy a “File” y hago click en “Save” (1) y luego cierro el archivo de texto.



Edit config file

nature
protocols

PROTOCOL

<https://doi.org/10.1038/s41596-019-0176-0>

Using DeepLabCut for 3D markerless pose estimation across species and behaviors

Tanmay Nath^{1,5}, Alexander Mathis^{1,2,5}, An Chi Chen³, Amir Patel³, Matthias Bethge⁴ and Mackenzie Weygandt Mathis^{1,*}

[Link al artículo](#)

Box 1 | Glossary of parameters in the project configuration file (*config.yaml*)

The *config.yaml* file sets the various parameters for generation of the training set file and evaluation of results. The meaning of these parameters is defined here, as well as referenced in the relevant steps.

Parameters set during the project creation

- **task:** Name of the project (e.g., mouse-reaching). (Set in Step 1; do not edit.)
- **scorer:** Name of the experimenter (set in Step 1; do not edit).
- **date:** Date of creation of the project. (Set in Step 1; do not edit).
- **project_path:** Full path of the project, which is set in Step 1; edit this if you need to move the project to a cluster/server/another computer or a different directory on your computer.
- **video_sets:** A dictionary with the keys as the full path of the video file and the values, `crop` as the cropping parameters used during frame extraction. (Step 1; use the function `add_new_videos` to add more videos to the project; if necessary, the paths can be edited manually, and the `crop` can be edited manually).

Important parameters to edit after project creation

- **bodyparts:** List containing names of the points to be tracked. The default is set to `bodypart1`, `bodypart2`, `bodypart3`, `objectA`. Do not change after labeling frames (and saving labels). You can add additional labels later, if needed.
- **numframes2pick:** This is an integer that specifies the number of frames to be extracted from a video or a segment of video. The default is set to 20.
- **colormap:** This specifies the colormap used for plotting the labels in images or videos in many steps. Matplotlib colormaps are possible (https://matplotlib.org/examples/color/colormaps_reference.html).
- **dotsize:** Specifies the marker size when plotting the labels in images or videos. The default is set to 12.
- **alphavalue:** Specifies the transparency of the plotted labels. The default is set to 0.5.
- **iteration:** This keeps the count of the number of refinement iterations used to create the training dataset. The first iteration starts with 0 and thus the default value is 0. This will auto-increment once you merge a dataset (after the optional refinement stage).

If you are extracting frames from long videos

- **start:** Start point of interval to sample frames from when extracting frames. Value in relative terms of video length, i.e., [`start=0, stop=1`] represents the full video. The default is 0.
- **stop:** Same as `start`, but specifies the end of the interval. Default is 1.

Related to the Neural Network Training

- **TrainingFraction:** This is a two-digit floating-point number in the range [0-1] used to split the dataset into training and testing datasets. The default is 0.95.
- **resnet:** This specifies which pre-trained model to use. The default is 50 (user can choose 50 or 101; see also Mathis et al.¹²).

Used during video analysis (Step 13)

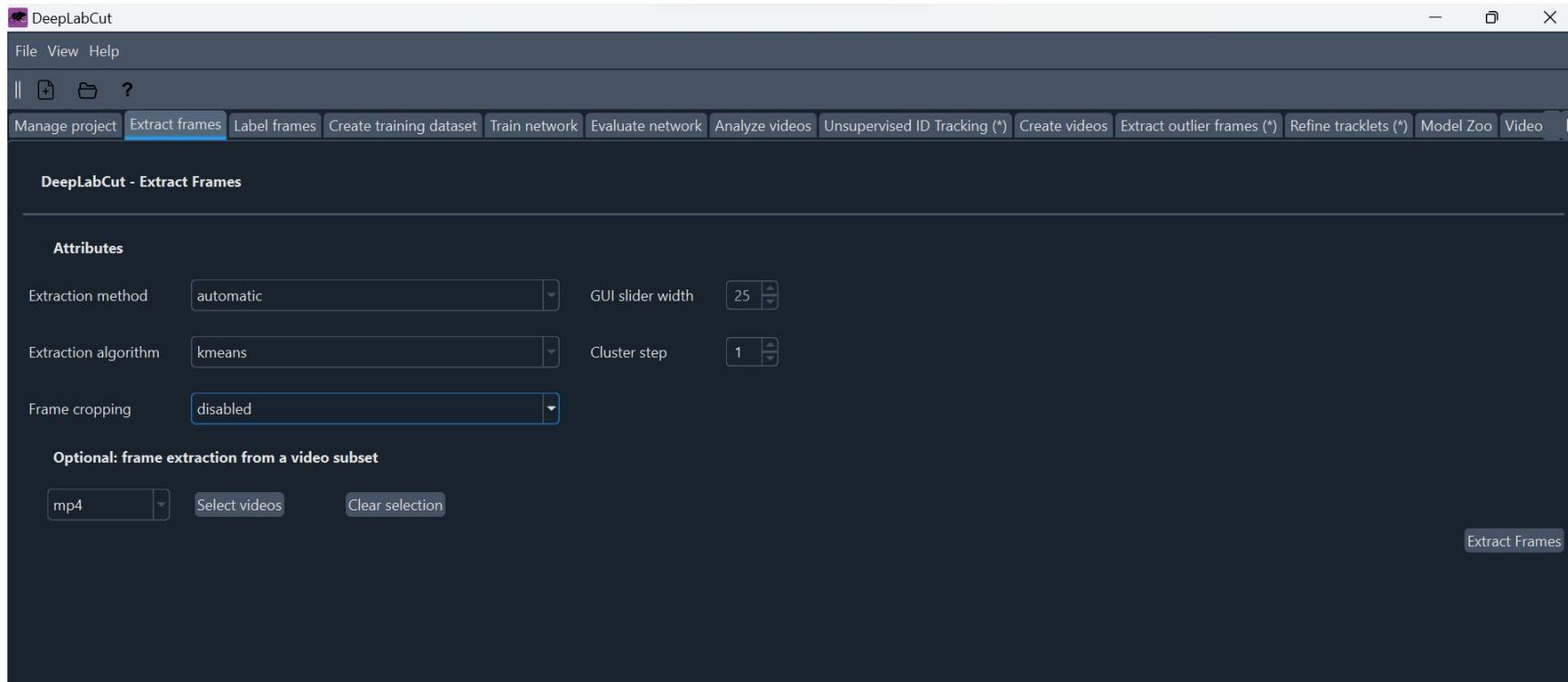
- **batch_size:** This specifies how many frames to process at once during inference (for tuning of this parameter, see Mathis and Warren²⁷).
- **snapshotindex:** This specifies which checkpoint to use to evaluate the network. The default is -1. Use `all` to evaluate all the checkpoints. Snapshots refer to the stored TensorFlow configuration, which holds the weights of the feature detectors.
- **pcutoff:** This specifies the threshold of the likelihood and helps to distinguish likely body parts from uncertain ones. The default is 0.1.
- **cropping:** This specifies whether the analysis video needs to be cropped (in Step 13). The default is `False`.
- **x1, x2, y1, y2:** These are the cropping parameters used for cropping novel video(s). The default is set to the frame size of the video.

Used during refinement steps

- **move2corner:** In some (rare) cases, the predictions from DeepLabCut will be outside of the image (because of the location refinement shifts). This binary parameter ensures that those points are mapped to a user-defined point within the image so that the label can be manually moved to the correct location. The default is `True`.
- **corner2move2:** This is the target location, if `move2corner` is `True`. The default is set to (50, 50).

Extract frames

Como acabo de configurar todo, me abre todo con esa configuración, y solo pongo “OK” (1). Ir a la terminal a ver si está cargando



Extract frames

La función de extracción de fotogramas puede hacerlo de tres formas:

- a) **uniform**: selección aleatoria uniforme (recomendado si las posturas varían a lo largo del video),
- b) **k-means**: agrupamiento basado en la apariencia visual de los fotogramas (útil para comportamientos raros o rápidos, como el "reach and pull"),
- c) **manual**: selección directa por parte del usuario.

Extract frames

El resultado es:

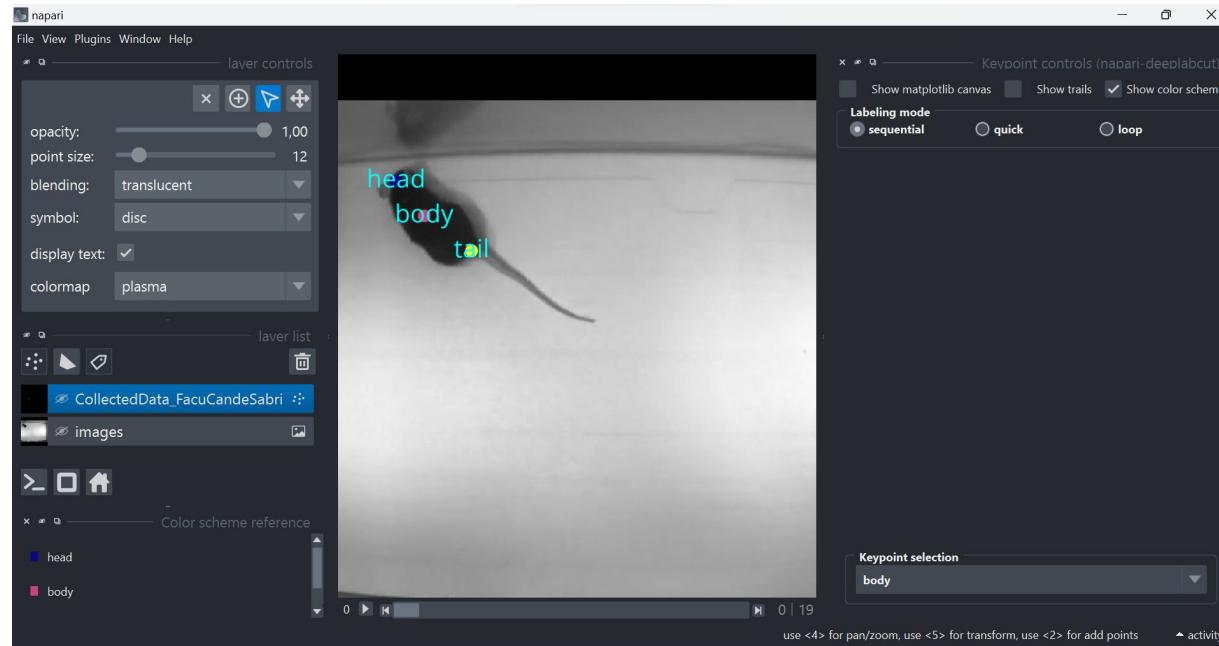
```
Frames were successfully extracted, for the videos listed in the config.yaml file.
```

```
You can now label the frames using the function 'label_frames' (Note, you should label frames extracted from diverse videos (and many videos; we do not recommend training on single videos!)).
```

Label data

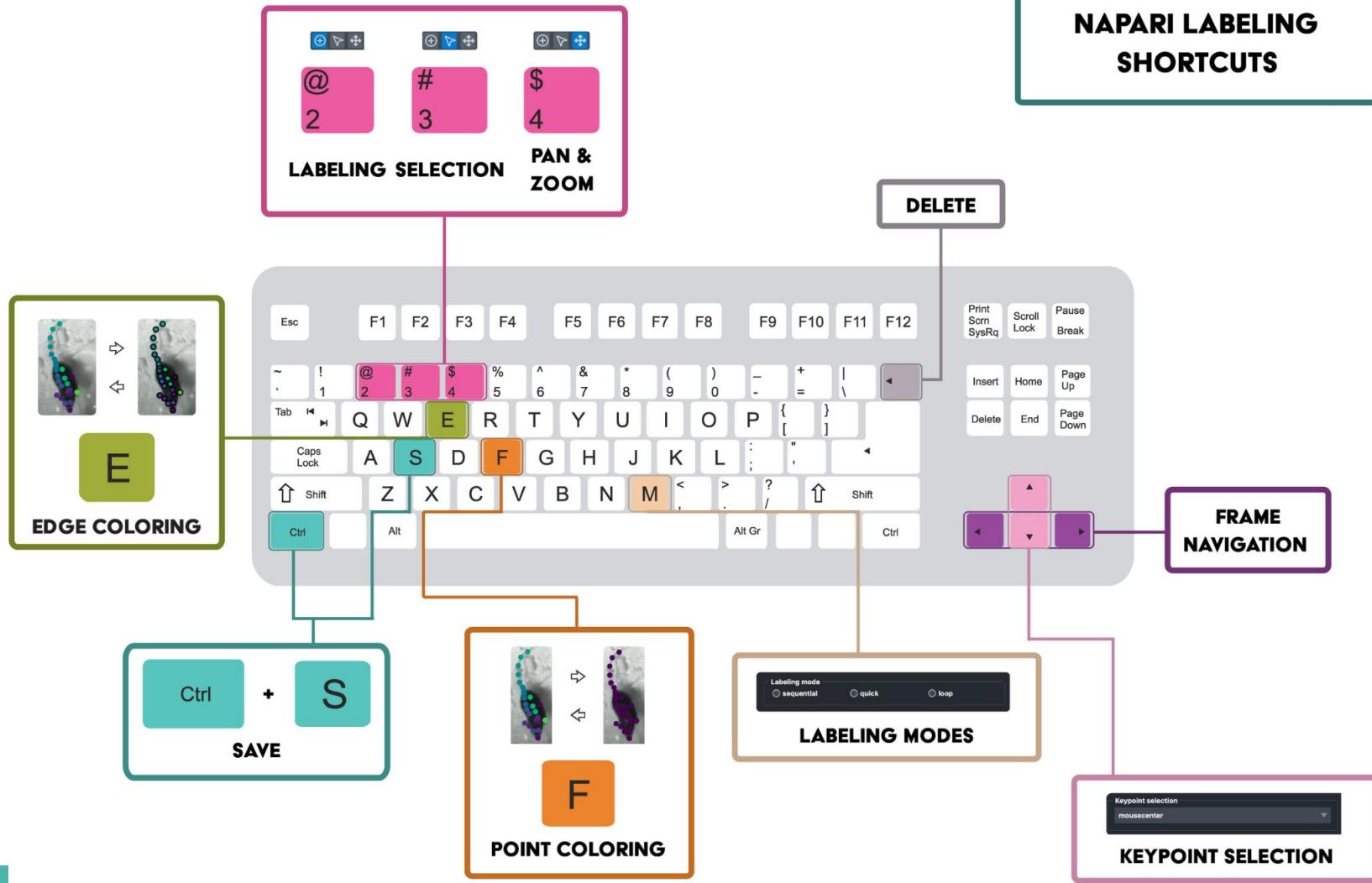
Va a aparecer una nueva ventana. Tenemos que cargar el conjunto de frames que se seleccionaron de cada video para etiquetarlos.

Tocamos en **Load Data (1)** y elegimos la carpeta dentro de **\labeled-data** con el nombre del video



En general, los puntos invisibles u ocultos no deben ser etiquetados por el usuario. Simplemente pueden omitirse sin colocar ninguna etiqueta en el fotograma.

NAPARI LABELING SHORTCUTS



Crear Videos con labels

(Gracias Mexico Hub!)

14.1. El más sencillo:

```
deeplabcut.create_labeled_video(config_path, ["ruta/a/video1.avi","ruta/a/video2.avi"], save_frames=False)
```

14.2. Con datos filtrados

```
deeplabcut.create_labeled_video(config_path, ["ruta/a/carpeta_de_videos"], videotype=".mp4", filtered=True)
```

14.3. Puntos arrastrados (trailpoints):

```
deeplabcut.create_labeled_video(config_path, ["ruta/a/carpeta"], videotype=".mp4", trailpoints=10)
```

14.4. Dibujar esqueleto:

```
deeplabcut.create_labeled_video(config_path, ["ruta/a/carpeta"], videotype=".mp4", draw_skeleton=True)
```

14.5. Solo con los puntos:

```
deeplabcut.create_labeled_video(config_path, ["ruta/a/carpeta"], videotype=".mp4", keypoints_only=True)
```

TIP:

Mejor calidad: Añade fastmode=False siempre que uses trailpoints o draw_skeleton.

Refinamos el entrenamiento

extraer cuadros problemáticos (**outliers**), corregirlos manualmente y volver a entrenar.

```
deeplabcut.extract_outlier_frames(config_path, ["videofile_path"])
```

Métodos disponibles para detectar outliers (outlieralgorithm):

"uncertain" → frames donde la probabilidad (likelihood) de una parte del cuerpo es baja.

"jump" → frames donde una parte "salta" más de cierto umbral (epsilon) entre cuadros consecutivos.

"fitting" → frames donde la predicción se desvía de un modelo temporal (ARIMA).

"manual" → selección manual por inspección visual.

Al abrir la GUI (`deeplabcut.refine_labels(config_path)`), se corrigen las etiquetas de los cuadros extraídos:

Una vez corregidas, se **fusionan los datasets** con:

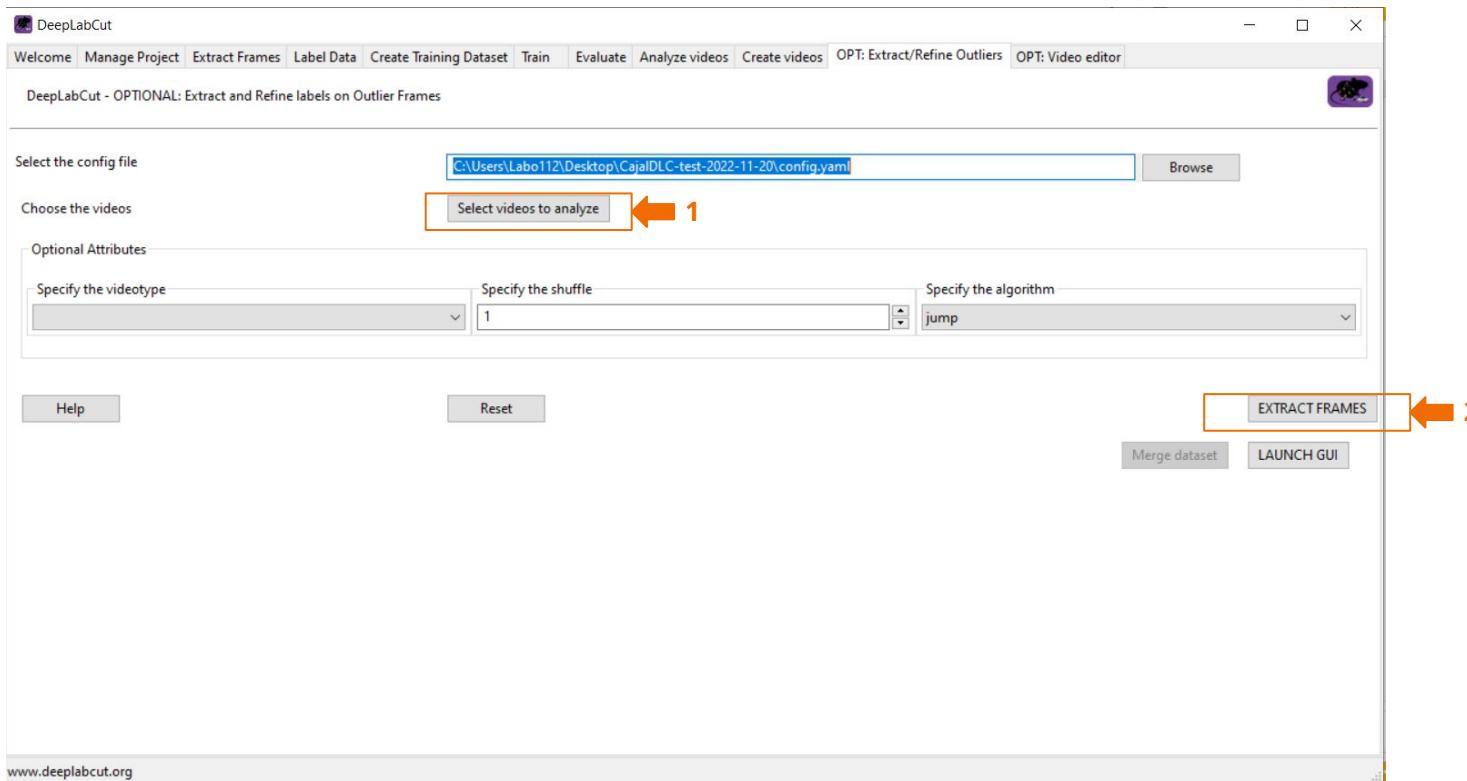
```
deeplabcut.merge_datasets(config_path)
```

Esto crea un **nuevo dataset** (nueva iteración) que sirve para generar un nuevo set de entrenamiento y volver a entrenar la red.

Si ajustaste las etiquetas originales, lo mejor es entrenar desde pesos frescos; si no, podés reutilizar pesos previos.

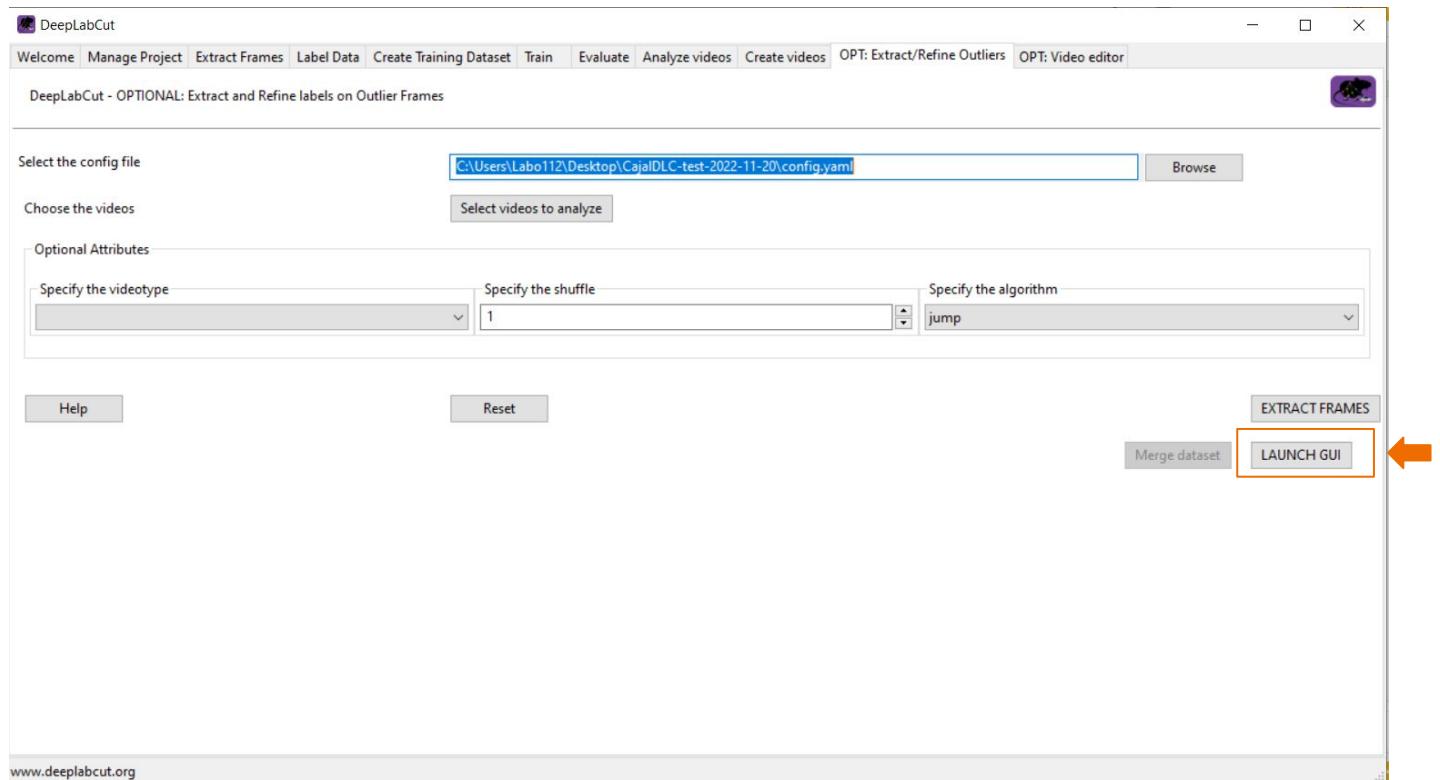
Extract Outlier Frames

Si vemos que tenemos muchos errores en el video podemos corregir algunas predicciones, aumentar el training set y volver a entrenar. Para eso volvemos a cargar los videos (1) y extraemos los frames (2)



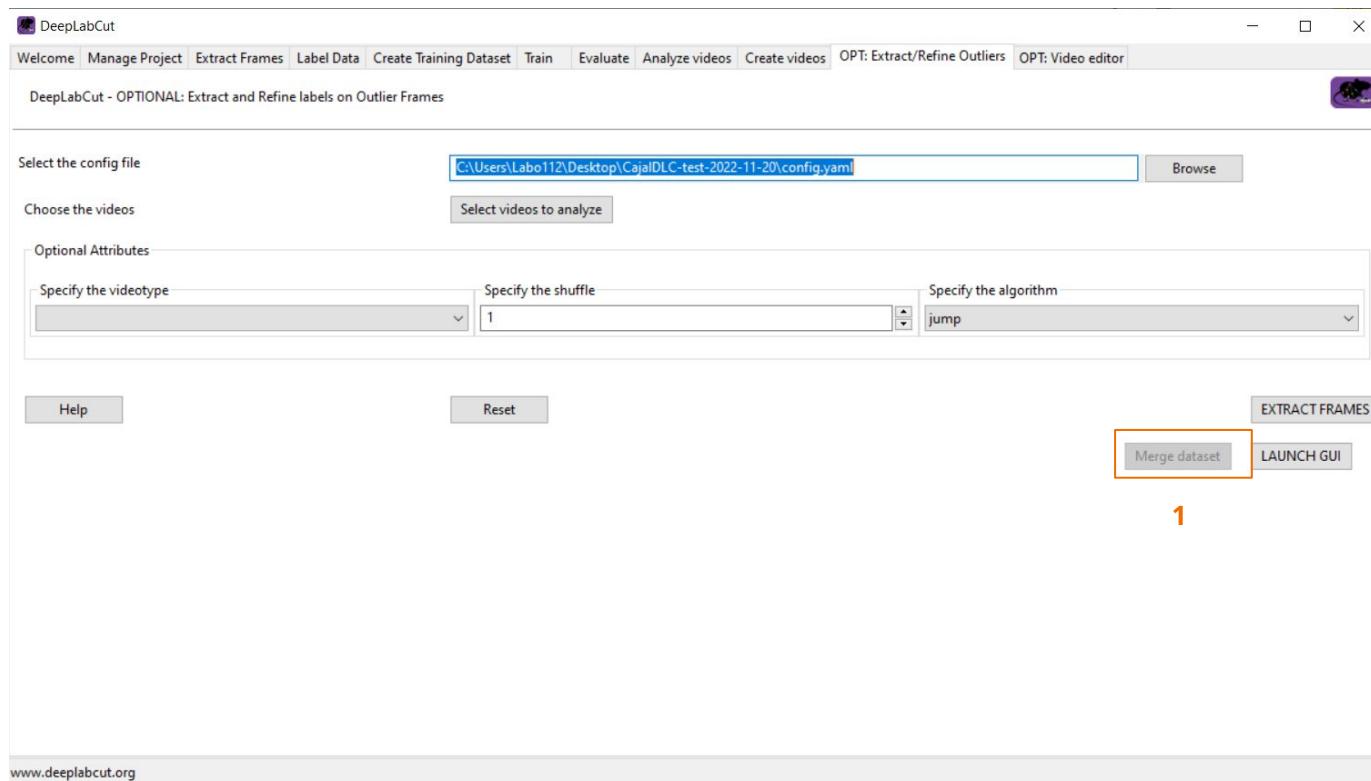
Refine Labels

Para corregirlos, abrimos la GUI (1) y se repiten los pasos de **label data**



MERGE!

No olvidar hacer el merge de los nuevos frames corregidos con los viejos (1). Ahora si podemos volver a crear un nuevo training_dataset y entrenar la red y volver a crear los videos con el pose estimation.



Resumen:

Detectar outliers → `deeplabcut.extract_outlier_frames()` con distintos algoritmos (incertidumbre, saltos, fitting con ARIMA, manual).

Corregir labels → `deeplabcut.refine_labels()` con la **GUI** (mover, borrar, validar).

Unir datasets → `deeplabcut.merge_datasets()` → genera nueva iteración de entrenamiento.

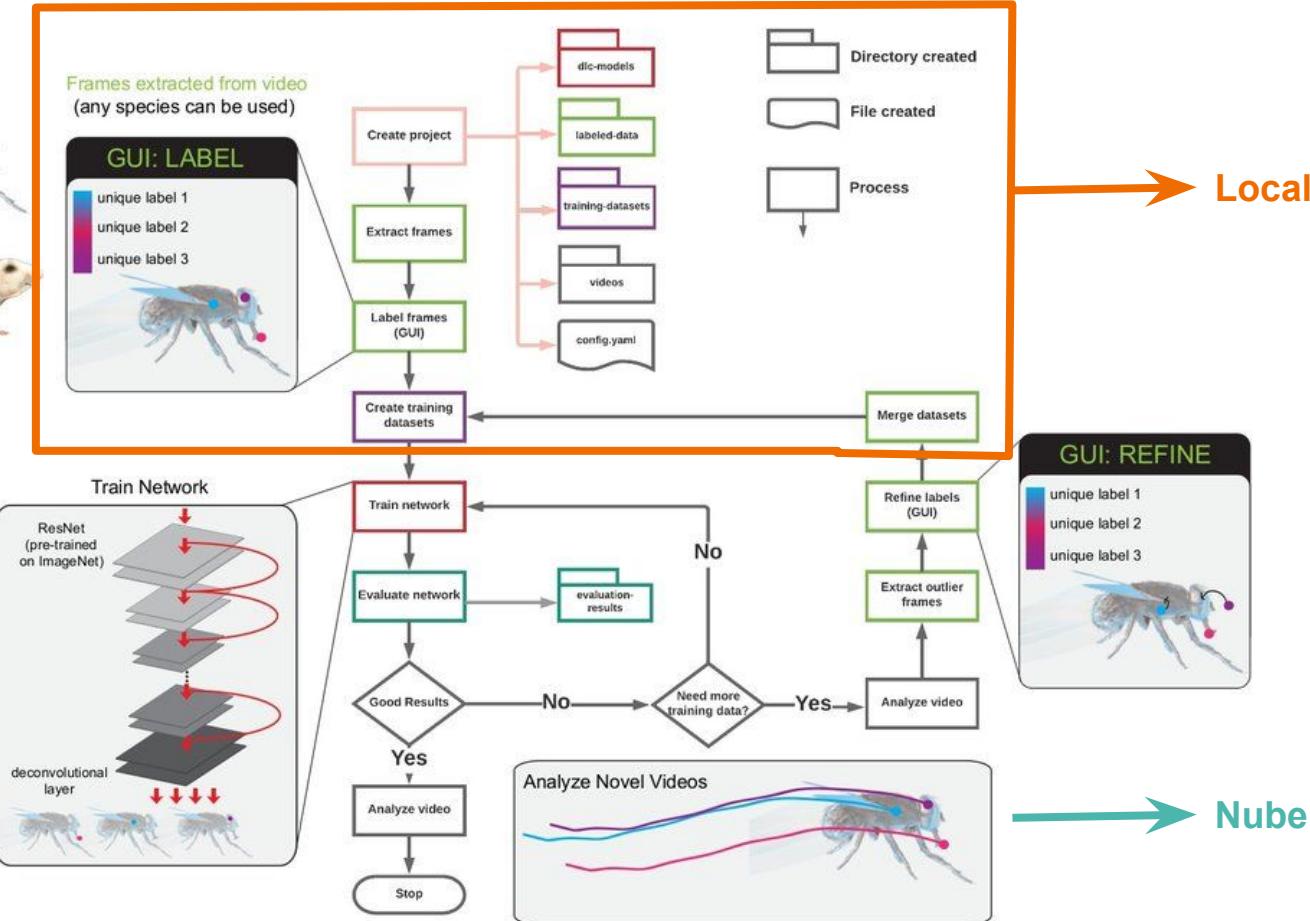
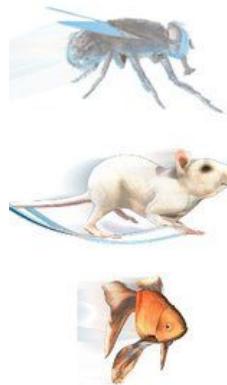
Reentrenar con el dataset expandido → PONER NUEVO **TRAININGSETINDEX**

```
deeplabcut.train_network(config_path, shuffle=1, trainingsetindex=<nueva_iteración>)
```

Repite el refinamiento hasta alcanzar la precisión deseada antes de analizar nuevos videos.

Google Colab

Para Colab



¿Cómo entreno en Colab un proyecto que etiqueté en la GUI?

- 1) Copiar el [código modelo](#) a un Drive propio para poder editarlo
- 2) Subir la carpeta del proyecto que creé en mi compu local a Drive.

- 3) Descargar el archivo **config.yaml** y abrirlo con un editor de texto

- 4) Modificar el parámetro "project_path" para que apunte a la nueva ubicación del proyecto en el Drive

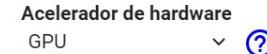
project_path: /content/drive/My Drive/

- 5) Guardar el archivo **config.yaml** y volverlo a subir para reemplazarlo en el Drive

---- *para los pasos 3-5 se puede [usar un atajo](#)*



- 6) En el menú Entorno de ejecución > Cambiar tipo de entorno de ejecución > GPU (!!)



Para copiar rutas de archivos en el Drive fácilmente, una vez que el Drive está montado se puede usar la ventana de la izq que muestra un árbol de archivos, navegar por las carpetas y hacer clic derecho sobre el archivo del cual queremos la ruta y usar 'copy as path'.

¿Cómo entreno en Colab un proyecto que etiqueté en la GUI?

continúa de la [slide anterior](#)

- 5) En el menú Entorno de ejecución > Cambiar tipo de entorno de ejecución > GPU
- 6) En el código, cambiar la variable path_config_file para que apunte a la ubicación en el Drive del archivo config.yaml

```
# Creo una variable con el path de la config.yaml  
path_config_file = '/content/drive/MyDrive/...../config.yaml'
```

- 7) Crear el train dataset (si no lo hice localmente con la GUI)

```
[ ] deeplabcut.create_training_dataset(path_config_file) #creo el training dataset (es decir que imagenes van al train y cuales al test)
```

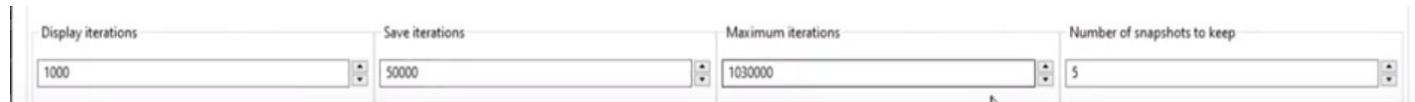
- 8) Entrenar la red con deeplabcut.train_network:

```
[ ] # Comenzamos a entrenar ! ==> segun el protocolo podemos tendiramos que entrenar en el orden de las 100K iteraciones o mas  
deeplabcut.train_network(path_config_file, shuffle=1, displayiters=5000, saveiters=15000)
```

Por defecto, maxiters es hasta que corta Colab y snapshots cada 5 snapshots, pero se pueden especificar los valores así:

deeplabcut.train_network(path_config_file, shuffle=1, displayiters=5000, saveiters=15000, max_iterators=100000, max_snapshots_to_keep=5)

(Tal cual vimos que podíamos hacer en la GUI)



El entrenamiento lo podemos cortar cuando queramos (o cuando se corta Colab, [ver slide siguiente](#)).

Nos conviene guardar varios snapshots (se guardan en la carpeta **/dlc-models/iteration-n/NOMBRE/train**, igual pesan, tampoco guardaríamos uno por cada iteración), para poder [seguir entrenando desde el último snapshot, con los pesos ya entrenados.](#)

¿Cómo entreno en Colab un proyecto que etiqueté en la GUI?

continúa de la [slide anterior](#)

Por defecto, maxiters es hasta que corta Colab y snapshots cada 5 snapshots, pero se pueden especificar los valores así:

```
deeplabcut.train_network(path_config_file, shuffle=1, displayiters=5000, saveiters=15000, maxiters=100000,  
max_snapshots_to_keep=5)
```

El entrenamiento lo podemos cortar cuando queramos (o cuando se corta Colab, [ver slide siguiente](#)).

Nos conviene guardar varios snapshots (se guardan en la carpeta **/dlc-models/iteration-n/NOMBRE/train**, igual pesan, tampoco guardaríamos uno por cada iteración), para poder [seguir entrenando desde el último snapshot, con los pesos ya entrenados.](#)

¿Qué limitaciones tiene Colab?

Nos restringen el uso gratuito de sus GPU a un cierto límite de tiempo. Aprox 90 mins sin actividad, y hasta un máx de 12 horas de GPU/usuario.

Muchas veces no alcanza para entrenar la red con el número de iteraciones recomendado.

¿Cómo las supero?

- A. Entreno guardando snapshots y cuando se me agota el tiempo de Colab vuelvo a abrirlo y sigo entrenando usando el último snapshot del entrenamiento anterior.
[VER SLIDE SIGUIENTE](#) Ojo que los snapshot pesan, tampoco vamos a guardar uno por cada iteración.
- B. Link con truquitos para que Colab nos deje correr al máximo de tiempo
<https://stackoverflow.com/questions/55050988/can-i-run-a-google-colab-free-edition-script-and-then-shut-down-my-computer>
- C. Fuera de Colab, usando una máquina con GPU o corriendo con CPU con una red liviana como mobilenet por *muchas* horas.
- D. Sin entrenar!! Uso un modelo entrenado que me pasa alguien, ejemplo [ModelZoo](#)

¿Cómo usar pesos de una red ya entrenada en Colab?

(Ya hice el paso del training y tengo guardado uno o varios snapshots de la red entrenada)

1) En Drive, dentro del proyecto, abrir la carpeta **/dlc-models/iteration-n/NOMBRE/train**

Aquí deben estar guardados los snapshots del training anterior, por ejemplo usamos el snapshot-30000.

2) Descargar el archivo **pose_cfg.yaml** y abrirlo con un editor de texto

3) Modificar el parámetro "init_weights" para que apunte a la ruta del snapshot de la última iteración en vez de a la red preentrenada

init_weights: /content/drive/My Drive/your project/dlc-models/iteration-n/NOMBRE/train/snapshot-30000

en vez de lo que tienen que es algo como:

```
'init_weights': '/usr/local/lib/python3.7/dist-packages/deeplabcut/pose_estimation_tensorflow/models/pretrained/resnet_v1_50.ckpt'
```

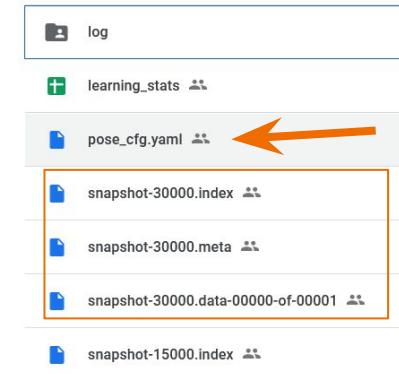
4) Guardar el archivo **pose_cfg.yaml** y volverlo a subir para reemplazarlo en el Drive

5) En Colab, correr de nuevo el entrenamiento con deeplabcut.train_network (ver [esta slide](#))

Cada snapshot se guarda como tres archivos: *.index*, *.meta* y *.data*

En la ruta que le indicamos a *init_weights*, no debemos poner ninguna terminación,

solo snapshot-30000, sin terminación, porque así usa los tres archivos



Atajo para editar el config_file sin tener que bajarlo

Uso una función auxiliar llamada edit_config

Por ejemplo, para:

- Los pasos 3-5 donde se modifica el project_path de [esta slide](#):

```
from deeplabcut.utils.auxiliaryfunctions import edit_config  
edit_config(path_config_file,{'project_path': '/content/drive/My Drive/....'})
```

- Los pasos 2-4 donde se modifica el init_weights de [esta slide](#):

```
from deeplabcut.utils.auxiliaryfunctions import edit_config  
edit_config(path_config_file,  
'{init_weights': '/content/drive/My Drive/your project/dlc-models/iteration-n/NOMBRE/train/snapshot-30000'})
```

la ruta de path_config_file debe apuntar al config.yaml

De todas formas, se sugiere forzar el remount del Drive para que los cambios se realicen.

Arriba de todo en la función de montar, agregar el parámetro force_remount=True

Colab con un proyecto ya etiquetado (Demos de DLC)

[\COLAB YOURDATA TrainNetwork VideoAnalysis.ipynb](#)

[\COLAB DEMO mouse openfield.ipynb](#)

Los demos siguen los mismos pasos que hicimos. Tenemos que tener un proyecto ya etiquetado y subirlo a nuestro Drive (o en el demo se clona el repositorio de data demo) . Habilitar el uso de GPU. Instalar DLC, vincular el Colab con el Drive y apuntar a la ubicación de nuestro proyecto en el config.

- ▼ First, go to "Runtime" ->"change runtime type"->select "Python3", and then select "GPU"
Entorno de ejecución > cambiar entorno de ejecución

```
[ ] 1 #(this will take a few minutes to install all the dependences!)
2 !pip install deeplabcut
```

(Be sure to click "RESTART RUNTIME" if it is displayed above before moving on !)



- ▼ Link your Google Drive (with your labeled data, or the demo data):

First, place your project folder into you google drive! "i.e. move the folder named "Project-YourName-TheDate" into google drive.

```
[ ] 1 #Now, let's link to your GoogleDrive. Run this cell and follow the authorization instructions:
2 #(We recommend putting a copy of the github repo in your google drive if you are using the demo "examples")
3
4 from google.colab import drive
5 drive.mount('/content/drive')
```

YOU WILL NEED TO EDIT THE PROJECT PATH **in the config.yaml file** TO BE SET TO YOUR GOOGLE DRIVE LINK!

Typically, this will be: [/content/drive/My Drive/yourProjectFolderName](#)

Día 2

- Refinamos labels
- CEBRA + Proyecto 3
- Otras Herramientas
- MultiAnimal

Algunas Aclaraciones...

- DLC divide los datos en train y test → en el config es el .95
- La Importancia de la cámara fija
- desafíos Open Source → instalación y versiones
 - ver:
<https://forum.image.sc/t/deeplabcut-colab-installation-issues/107559/20>
-

CEBRA

[link al colab](#)

[Docs](#)

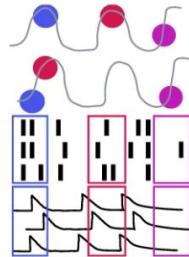
CEBRA

a

Behaviour labels

Time labels

Neural data (N)

Nonlinear encoder
(neural network (f))Contrastive learning
(loss function)

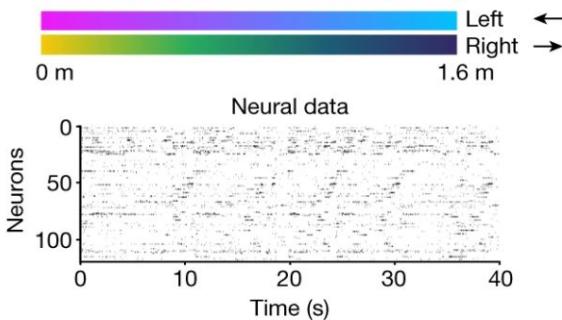
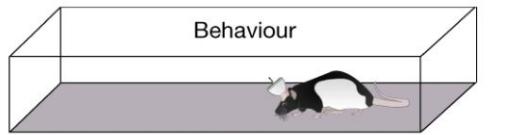
Low-dimensional embedding



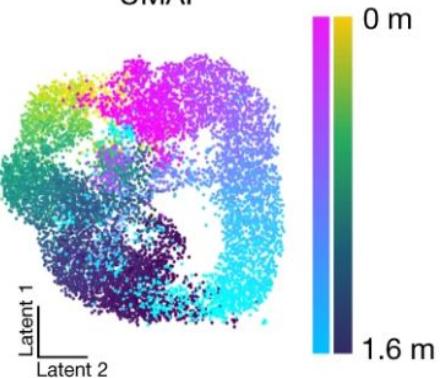
Final layer (L) output



+ Decoder

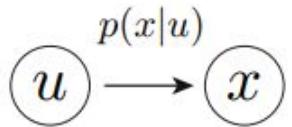
c**d**CEBRA
Behaviour

UMAP



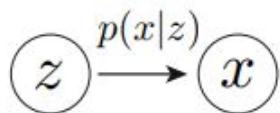
a

Regression-based
e.g. tuning curve



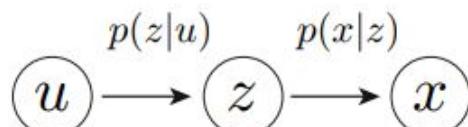
label observation

Latent-based
e.g. vanilla VAE



latent observation

Hybrid approach
e.g., our model

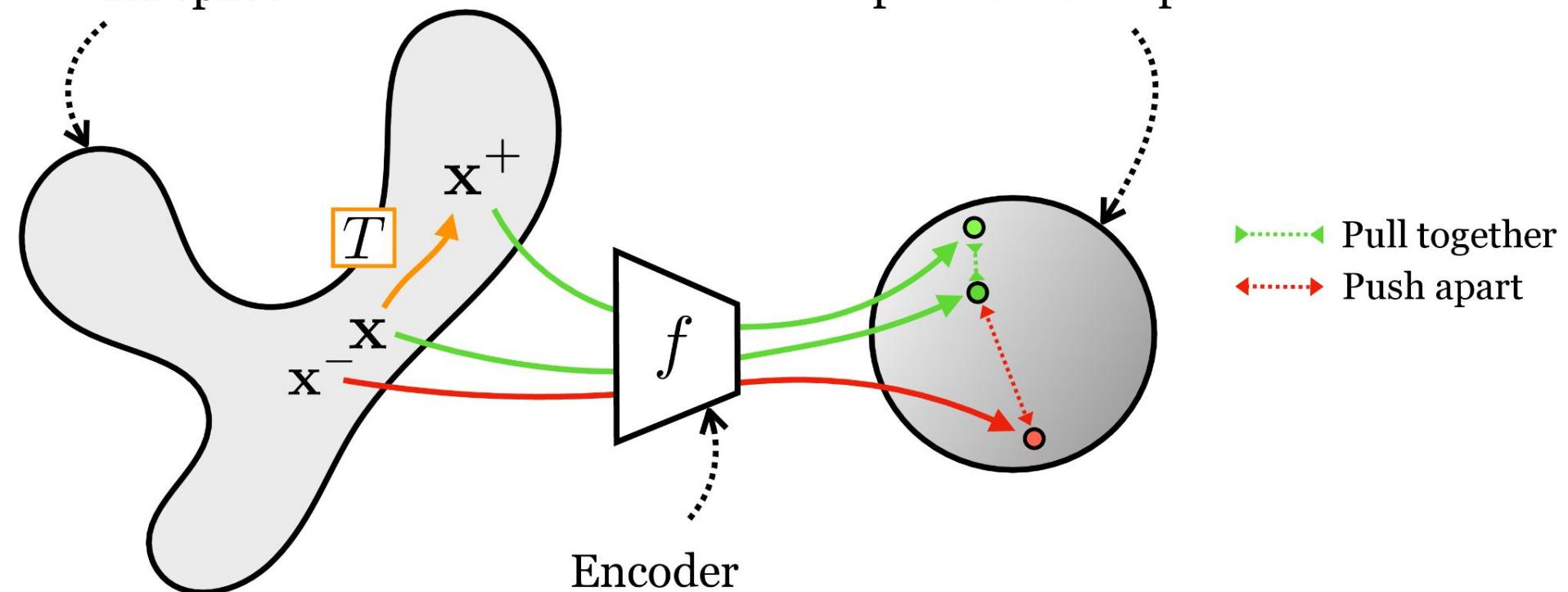


label latent observation

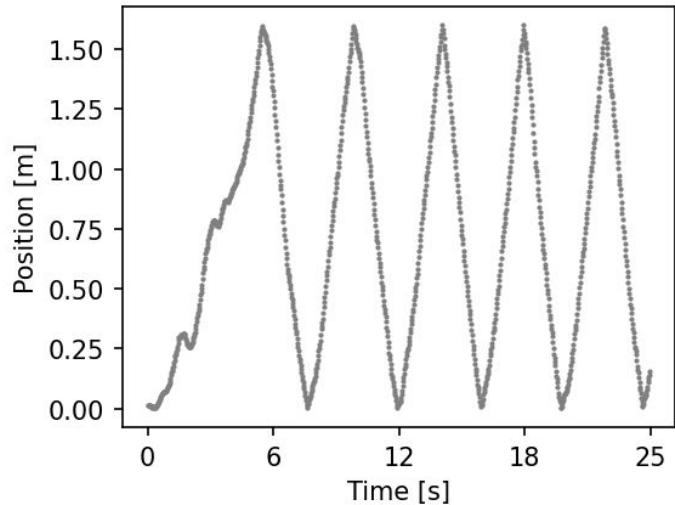
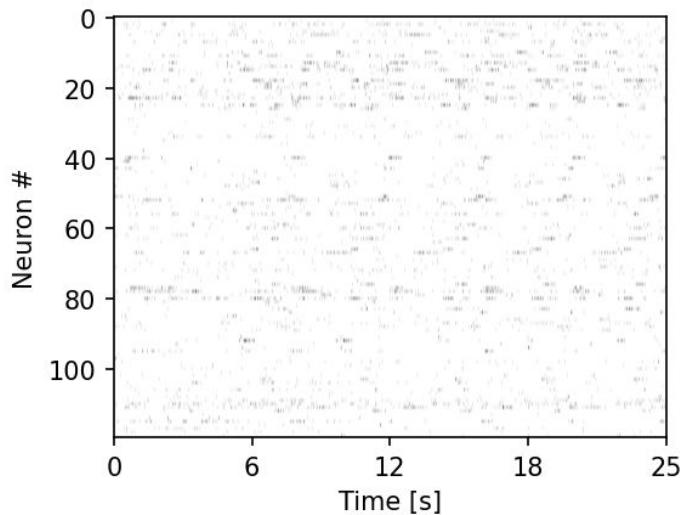
Contrastive Learning

Data space

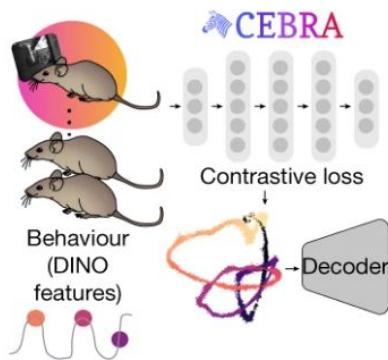
Representation space



Ejemplos:



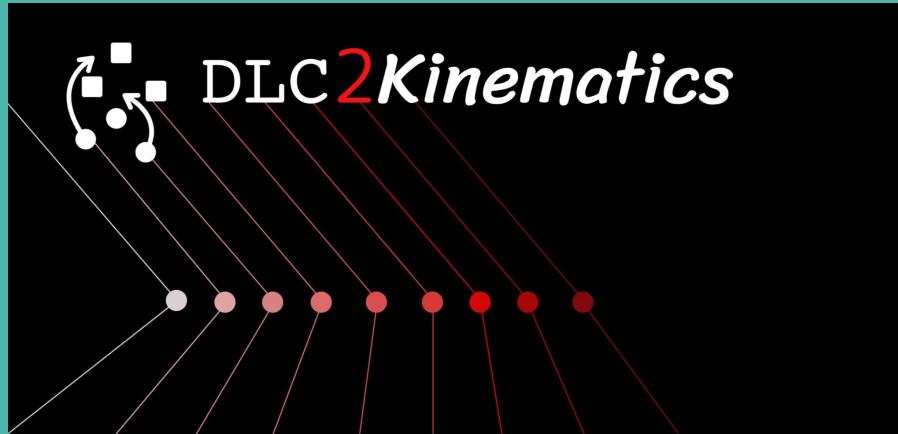
a Pseudo-mouse decoding



b



DLC2Kinematics: Analizamos el output de DLC



Output: archivo csv o h5

Sesion_1_2022_10_27DLC_resnet50_2022_10_27 y 2022_10_28Oct30shuffle1_400000.csv - LibreOffice Calc

A	B	C	D	E
1	scorer	DLC_resnet50_2022_10_27 y 2022_10_28Oct30shuffle1_400000	DLC_resnet50_2022_10_27 y 2022_10_28Oct30shuffle1_400000	DLC_resnet50_2022_10_27 y 2022_10_28Oct30shuffle1_400000
2	bodyparts	head	head	head
3	coords	x	y	likelihood
4	0 303.37835693359375	101.26012420654297	0.9999997615814209	284.9397888183594
5	1 321.29974365234375	98.5357894897461	0.9999995231628418	299.61297607421875
6	2 340.0681457519531	97.29985046386719	0.9999998807907104	320.7022399902344
7	3 363.398193359375	97.19338989257812	0.9999991655349731	344.16912841796875
8	4 378.8782958984375	94.6460952758789	0.9999947547912598	360.1474304199219
9	5 390.7650451660156	95.06999206542969	0.999998927116394	370.74932861328125
10	6 402.6038171386719	93.1337661743164	0.9999935626983643	380.8533935546875
11	7 417.10675048828125	93.8199462890625	0.9999974966049194	393.19512939453125
12	8 429.7066955566406	91.93440246582031	0.9999867677688599	405.603912353156
13	9 436.19049072265625	92.53881072998047	0.9999914169311523	418.4158630371094
14	10 447.6786193847656	97.06023406982422	0.9999988079071045	429.996826171875

Instalación de librería para abrir archivos h5

En el env de DLC: pip install dlc2kinematics
(para abrir el archivo h5)

Herramientas para Segmentar Comportamiento

SimBA: <https://github.com/sgoldenlab/simba>

DLC2Action: <https://github.com/AlexEMG/DLC2action#installation>

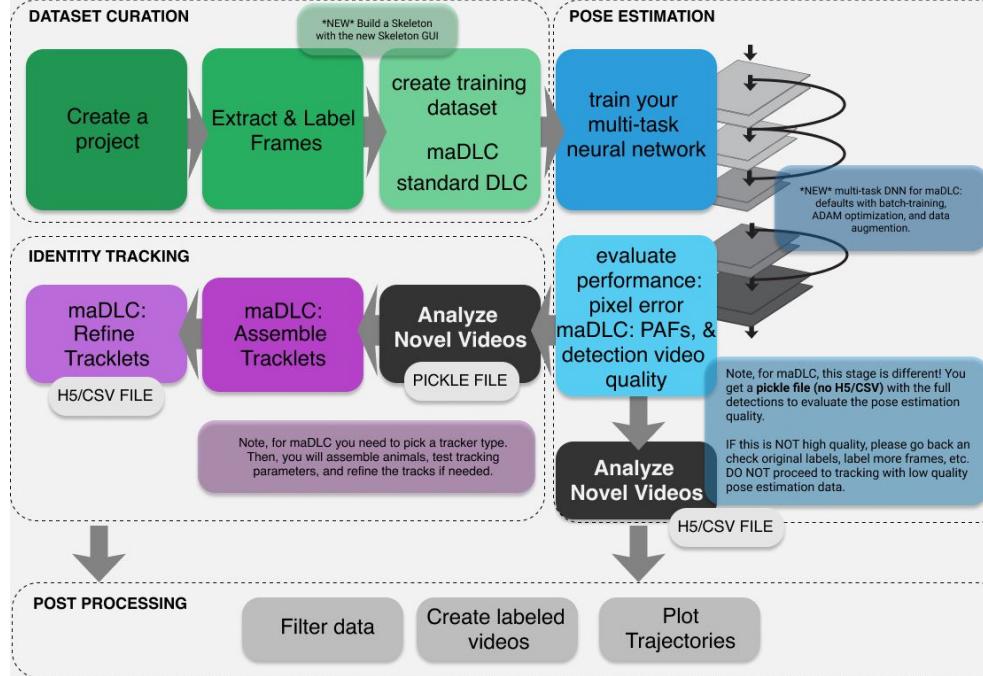
Moseq (Keypoint-SLDS)

DeepOF

Multi-Animal



DeepLabCut 2.2 workflow



[link1](#)
[link2](#)

<https://www.youtube.com/watch?v=Kp-stcTm77g>



Tips and tricks:

- . En el config file podemos decirle identity true o false si queremos preservar la identidad del animal o no. Por ejemplo: si tengo 3 ratones blancos no me interesa preservar el ID. Pero si los puedo distinguir sí. Aca es muy muy importante ser consistente en el etiquetado.
- . En el config file podemos decirle objetos o bodyparts que solo aparecen una vez por frame. Por ejemplo, un objeto en el ambiente. Eso lo ponemos en uniquebodyparts.
- . Es muy recomendable etiquetar frames donde los animales estén interactuando.
- . Si importa la identidad → ver assembly and tracking.

config file:

```
individuals:  
- m1  
- m2  
- m3  
  
uniquebodyparts:  
- topleftcornerofBox  
- toprightcornerofBox  
  
multianimalbodyparts:  
- snout  
- leftear  
- rightear  
- tailbase  
  
identity: True/False
```

Inference config file → para mantener el ID

```
# Tracking:  
#p/m pixels in width and height for increasing bounding boxes.  
boundingboxslack : 0  
  
# Intersection over Union (IoU) threshold for linking two bounding boxes  
iou_threshold: .2  
  
# maximum duration of a lost tracklet before it's considered a "new animal" (in frames)  
max_age: 100  
  
# minimum number of consecutive frames before a detection is tracked  
min_hits: 3
```

Refine → arreglar intercambios en el ID

The screenshot shows the DeepLabCut software interface with the 'Refine tracklets' tab selected. Several annotations provide instructions for using the tool:

- Select the config file:** "select the file that end with trackerName.pickle, such as "...el.pickle"
- Select the tracklet pickle file:** "smartly" completed based on your training, but you can update if you have more or less animals in the video!
- Number of animals:** "min swap length to highlight" is the length of a potential animal ID swap that the GUI can highlight for you, as a way to notify if there may be time points in the video needing checked!
- Optional: if you want to edit the animal tracking, pick the corresponding video!** Select the video
- Specify the max gap size of missing data to fill:** "max gap to fill" is if you want to interpolate and fill any gaps (due to occlusions)
- Optional: Refine tracks** Launch GUI!
- Optional: Filter Tracks (then you also get a CSV file!)** Once edited, if you want to smooth the data! (Typically recommended)
- Optional: Merge refined data**

Refine → arreglar intercambios en el ID

<https://www.youtube.com/watch?v=bEuBKB7eqmk>

