Creating a Training Dataset for Scratcher Using Roboflow

A complete step-by-step guide for absolute beginners

Introduction

Scratcher is a machine learning GUI-based tool that allows users to train models to automatically identify and quantify scratching behavior in mice.
In order to train a model using Scratcher, a properly annotated training dataset must first be created.

This document explains, in a very detailed and explicit manner, how to create such a training dataset using Roboflow.
No prior knowledge of machine learning, annotation, Roboflow, or computer vision is assumed.

If you follow this guide step by step, you will be able to generate a dataset that is fully compatible with Scratcher.

What you need before starting

Before beginning, make sure you have the following:

1. A computer running Windows, macOS, or Linux
2. A stable internet connection
3. A modern web browser such as Google Chrome, Microsoft Edge, or Firefox
4. Video recordings of mice that include scratching behavior using our custom made videotaping box and dimensions
5. The videos should be in MP4 (strongly preferred) or AVI format

---

Important technical requirement for Scratcher

Scratcher operates internally at exactly 30 frames per second.

This means that:

- All videos must be converted into image frames at exactly 30 frames per second
- Any other frame rate will lead to incorrect temporal alignment

- Incorrect frame rates will reduce model performance and may break downstream analysis

When extracting frames in Roboflow, you must ensure that the extraction rate is set to 30 frames per second.

This requirement applies to all datasets without exception.

Step 1: Opening Roboflow in your internet browser

1. Turn on your computer
2. Open your internet browser
3. Click once on the address bar at the top of the browser window
4. Type the following web address exactly as written: https://roboflow.com
5. Press the Enter key on your keyboard

You should now see the Roboflow homepage.

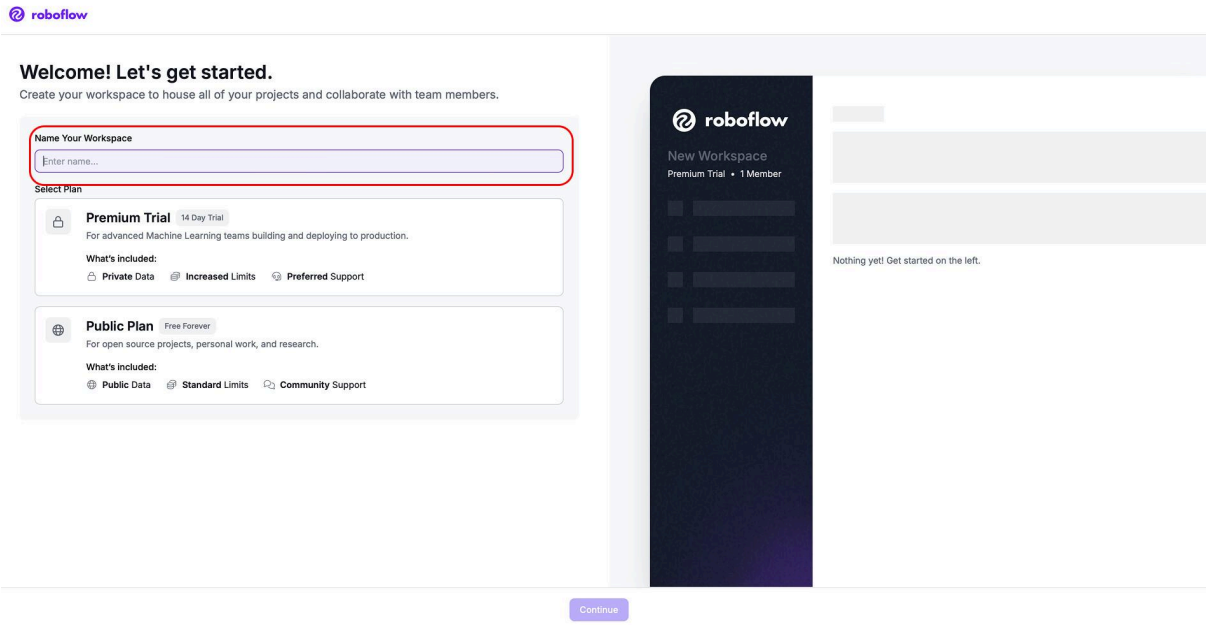
Step 2: Creating a Roboflow account or logging in

1. On the Roboflow homepage, locate the button labeled Sign Up
2. Click on Sign Up
3. Choose one of the available sign-up methods:
    - Google account
    - GitHub account
    - Email address and password
4. Follow the on-screen instructions to complete account creation
5. Once completed, you will be logged in automatically

If you already have a Roboflow account:

1. Click on Log In instead
2. Enter your login credentials
3. Proceed to the dashboard

Step 3: Creating a workspace



1. After logging in, you will see the Roboflow dashboard
2. Look for an option labeled Create Workspace
3. Click on Create Workspace
4. Enter a workspace name, for example: Scratcher_Mouse_Behavior
5. Set the workspace visibility to Private
6. Click the button to create the workspace

You will now be inside your newly created workspace.

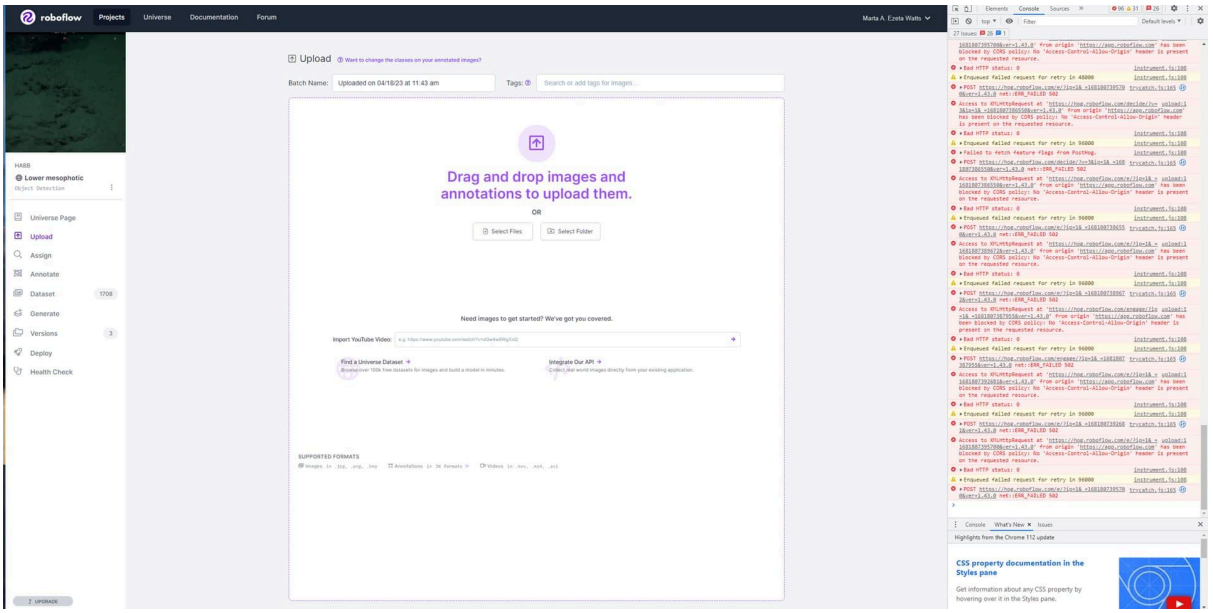
Step 4: Creating a new project

1. Inside your workspace, locate the button labeled Create Project
2. Click on Create Project
3. Enter a project name, for example: Mouse_Scratching_Detection
4. For project type, select Object Detection

5. For annotation type, select Bounding Boxes
6. Confirm and create the project

This project will hold all images, annotations, and dataset versions.

Step 5: Uploading videos to Roboflow



1. After creating the project, you will be taken to the Upload Data page
2. Click on the Upload button
3. Browse your computer and select the video files you want to use
4. Confirm the selection
5. Wait for the upload to complete

Upload time depends on video size and internet speed.

Step 6: Extracting frames from videos at 30 frames per second

test_mov-13.jpg test_mov-12.jpg test_mov-11.jpg test_mov-9.jpg test_mov-8.jpg test_mov-7.jpg test_mov-6.jpg

test_mov-5.jpg test_mov-3.jpg test_mov-2.jpg test_mov-0.jpg

This step is critical and must be done correctly.

1. After video upload, Roboflow will prompt you to extract frames
2. Choose the option for frame extraction
3. Select extraction by frames per second
4. Enter the value 30
5. Confirm frame extraction
6. Wait until Roboflow finishes extracting frames from all videos

Do not proceed unless frames have been extracted at exactly 30 frames per second.

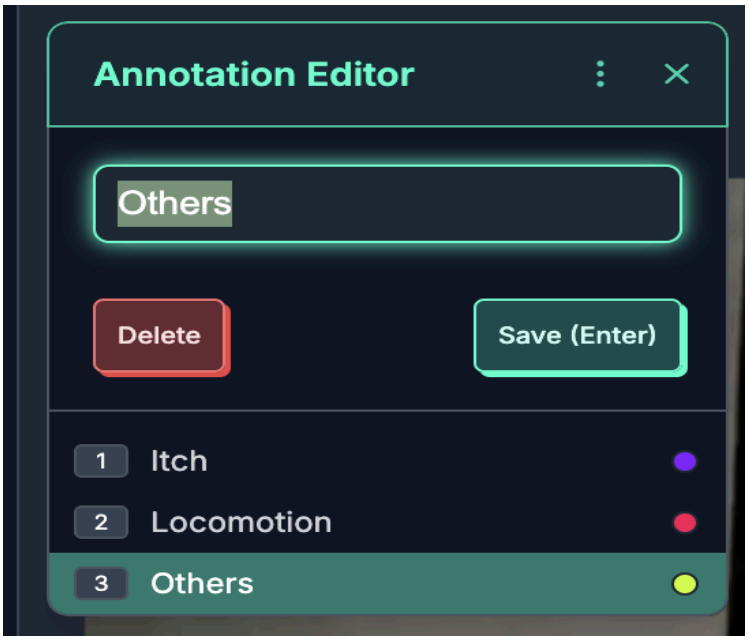
Step 7: Defining annotation classes

Scratcher requires exactly three annotation classes.
These class names must be used exactly as written.

The three classes are:

1. Itch
2. Locomotion
3. Others

To create these classes:



1. Go to the Annotate section of the project
2. Locate the class list panel
3. Add a new class named Itch
4. Add a new class named Locomotion
5. Add a new class named Others
6. Save the class list

Do not rename these classes and do not add additional classes.

Step 8: Understanding how to label each class correctly

Correct annotation is essential for accurate model training.

Class: Itch

Label a frame as Itch only when the scratching paw is actively approaching the nape of the neck or is in contact with the nape.

This includes:

- The upward movement of the hind paw toward the nape
- Active scratching contact on the nape

This does not include:

- The paw moving downward after completing a scratch
- Licking the paw after scratching
- Any grooming behavior
- Any rearing behavior

If the paw is not approaching or contacting the nape, the frame must not be labeled as Itch.

Class: Others

Label a frame as Others in the following situations:

- The paw is moving downward after completing a nape scratch
- The mouse is licking its paw after scratching
- The mouse is grooming itself
- The mouse is rearing in the arena
- Any stationary behavior that is not scratching

When uncertain between Itch and Others, choose Others unless the paw is clearly approaching or contacting the nape.

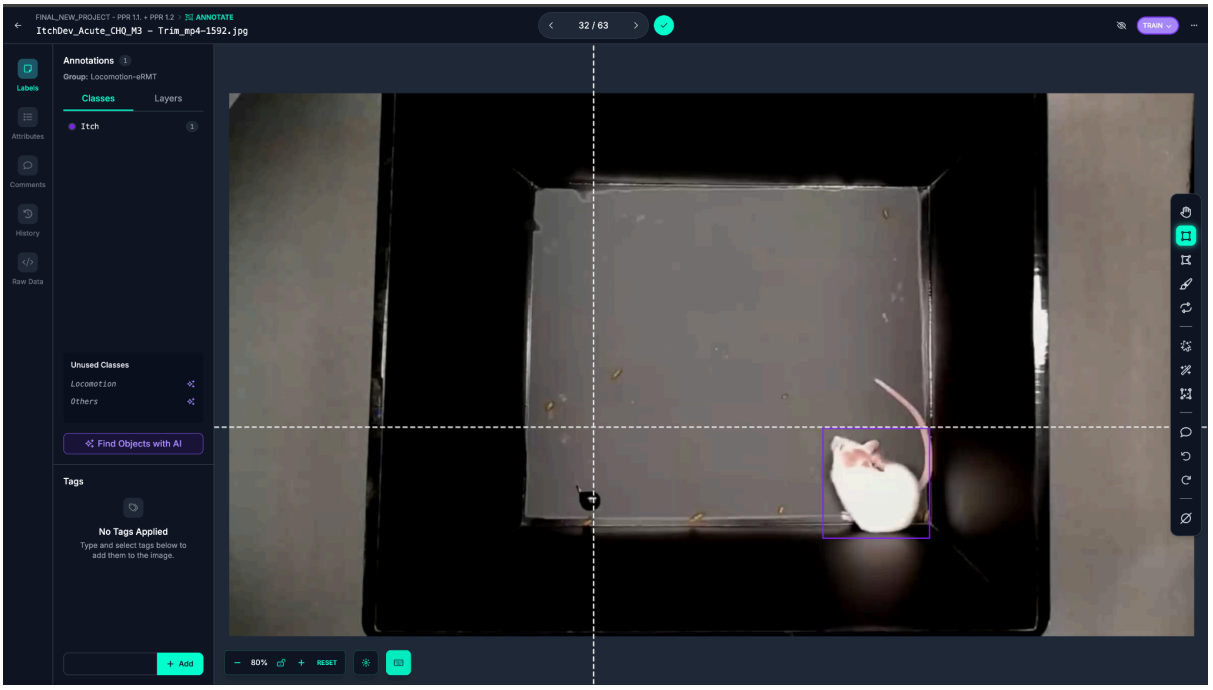
Class: Locomotion

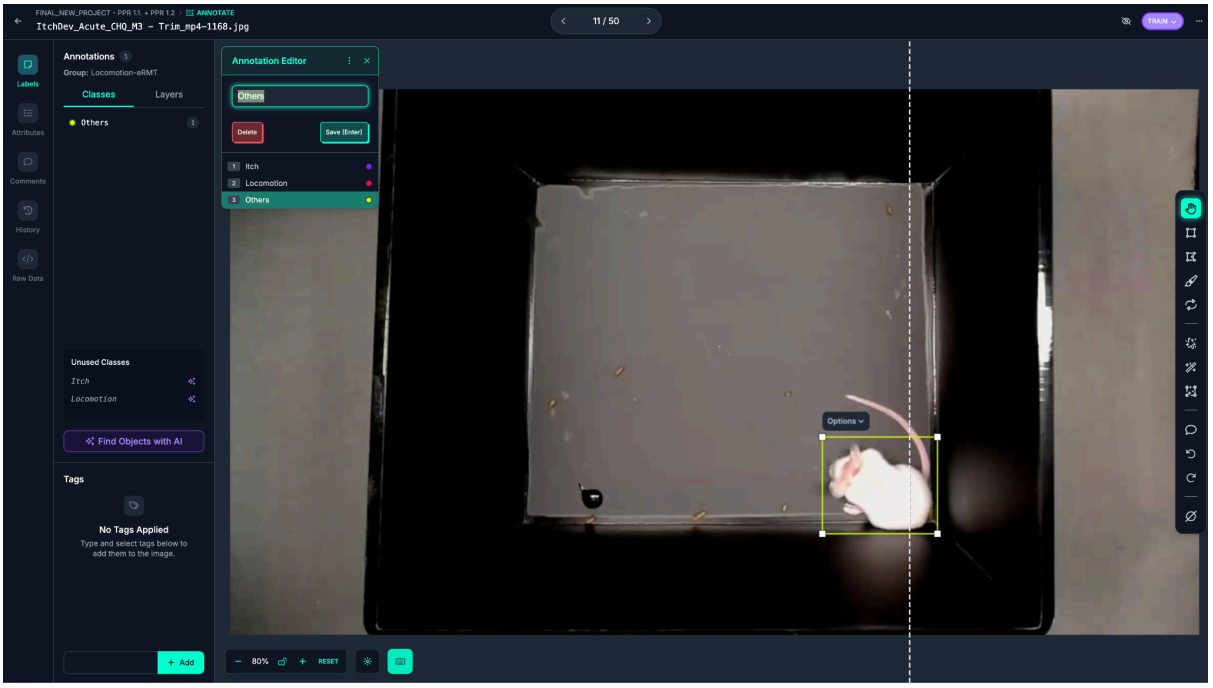
Label a frame as Locomotion when:

- The mouse is walking
- The mouse is running
- The mouse is freely moving around the arena without scratching or grooming

This class should only represent general movement.

Step 9: Annotating frames one by one





For each extracted image:

1. Click on the image to open it
2. Select the bounding box tool
3. Draw a bounding box around the mouse body
4. Assign exactly one class based on the rules above
5. Save the annotation
6. Move to the next image

Repeat this process for all images.

Consistency across frames is more important than speed.

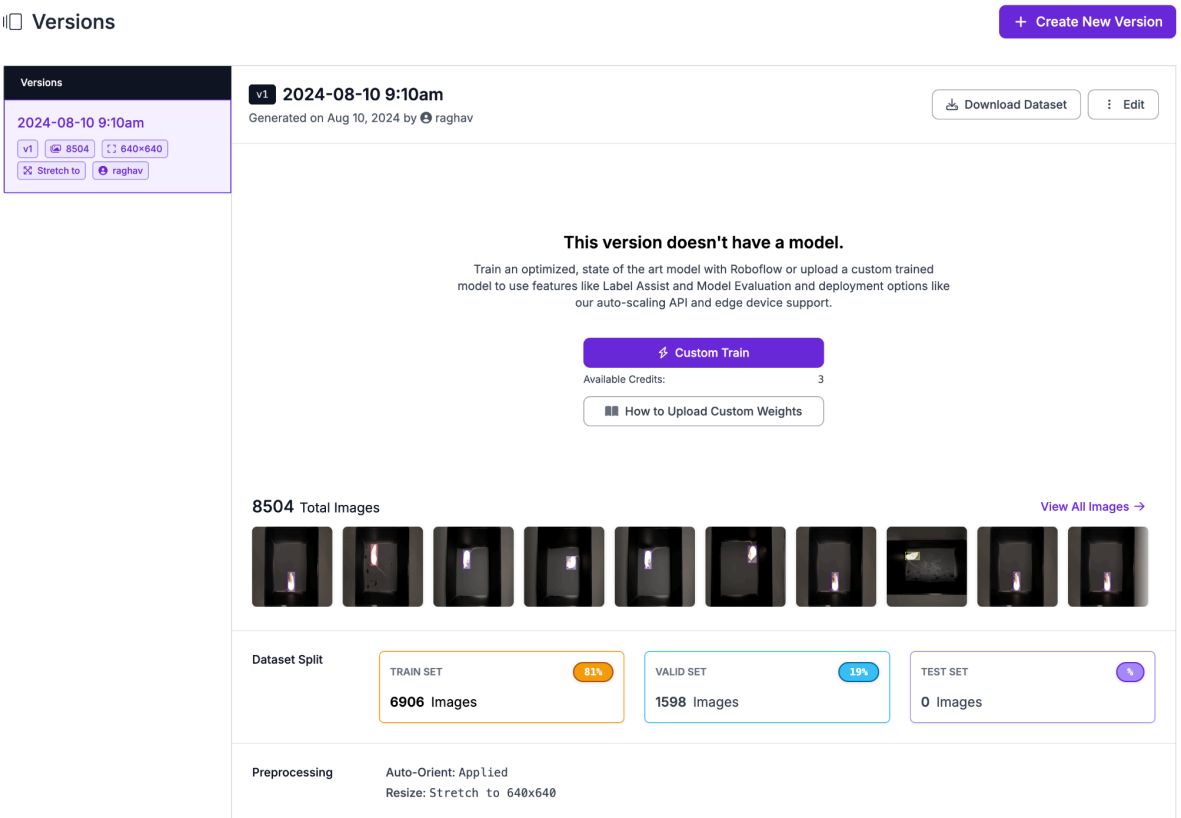
Step 10: Reviewing annotations

After annotating:

1. Scroll through annotated images
2. Check for incorrect labels
3. Check for missing annotations
4. Correct any mistakes before proceeding

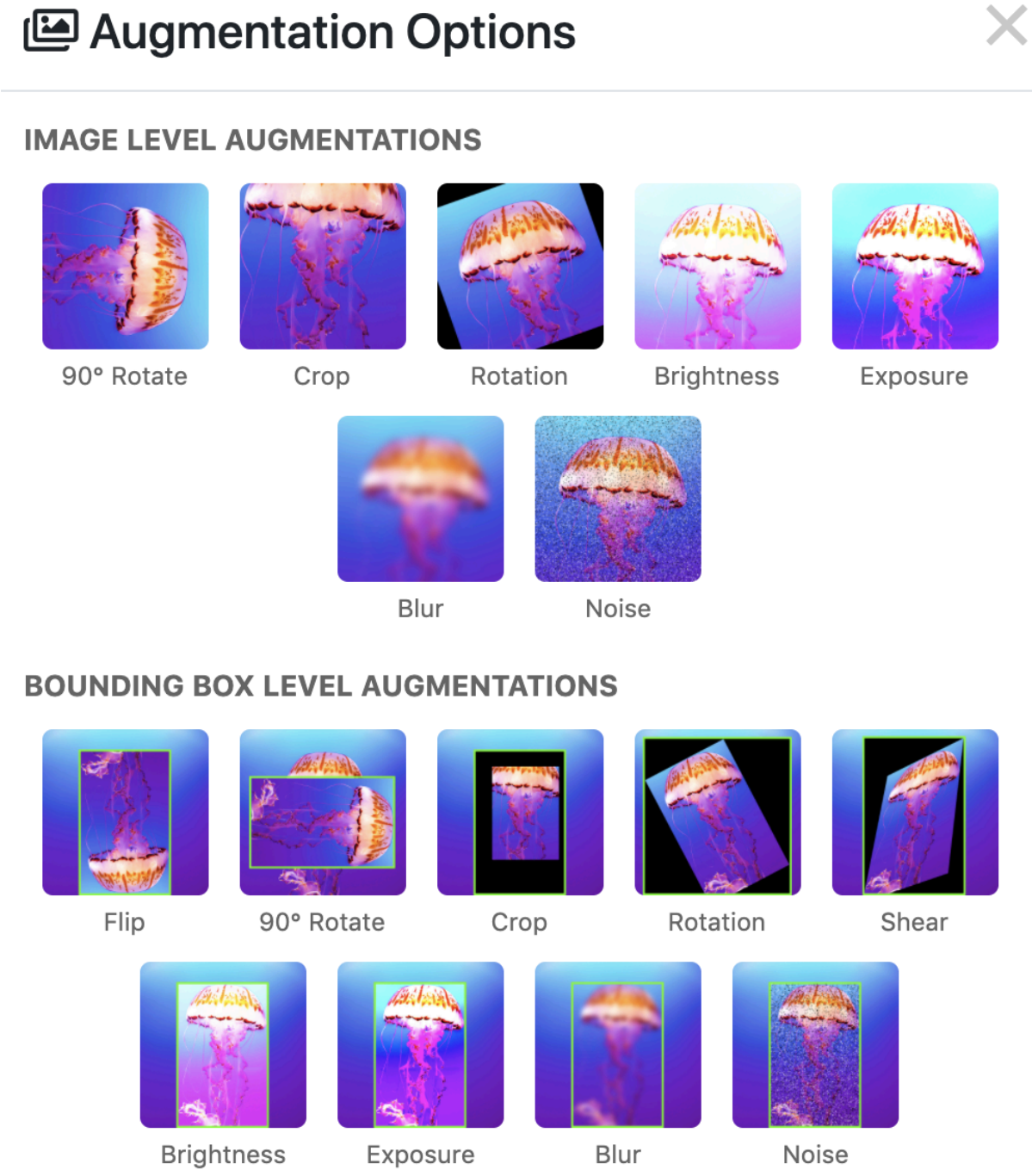
Poor annotation quality will directly reduce model performance.

Step 11: Creating a dataset version



1. Click on Generate New Version
2. Set the dataset split as follows:
    - Training: 80 percent
    - Validation: 20 percent
    - Test: 0 percent
3. Ensure the split is randomized
4. Proceed to version generation

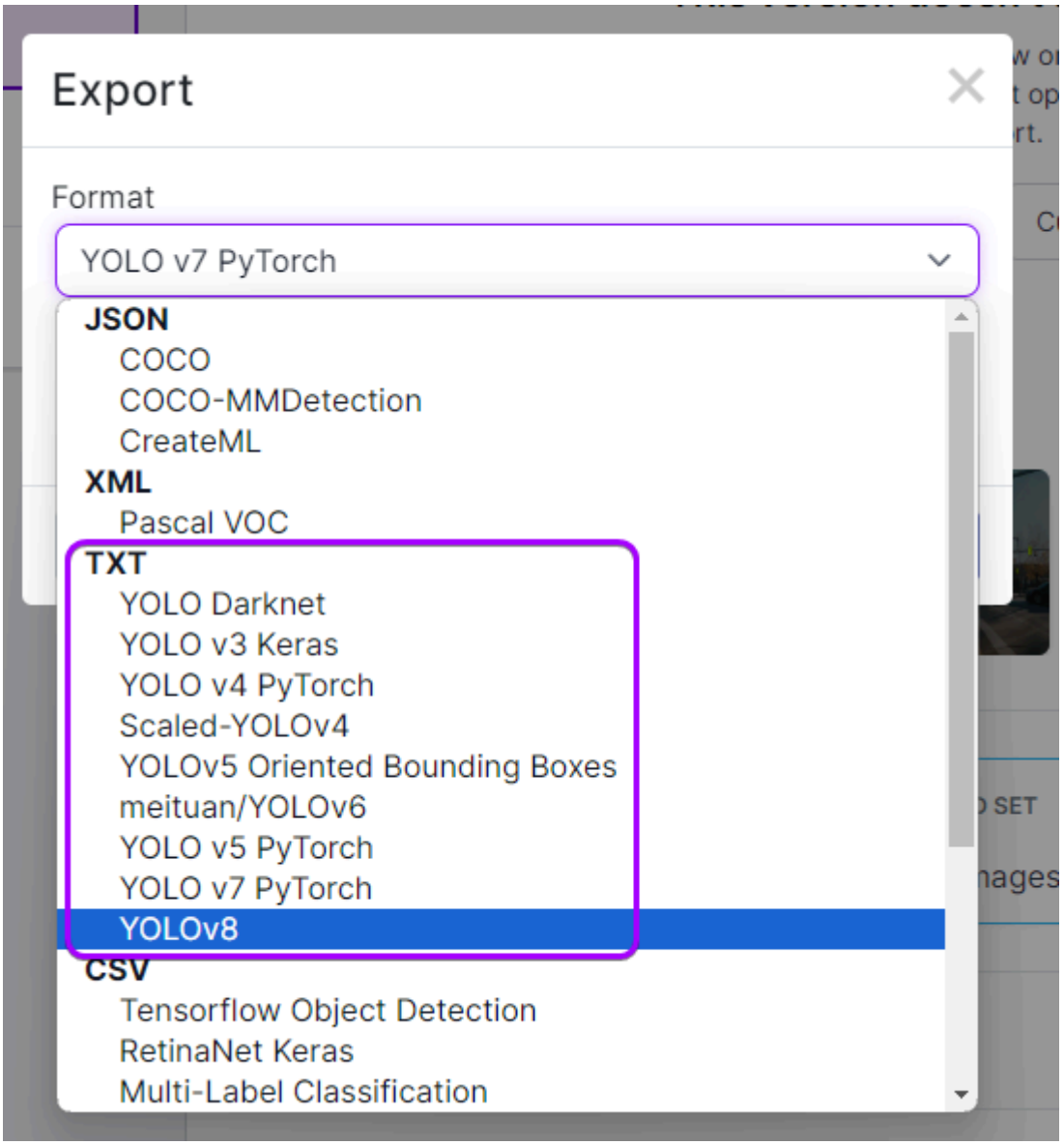
Step 12: Choosing data augmentation settings

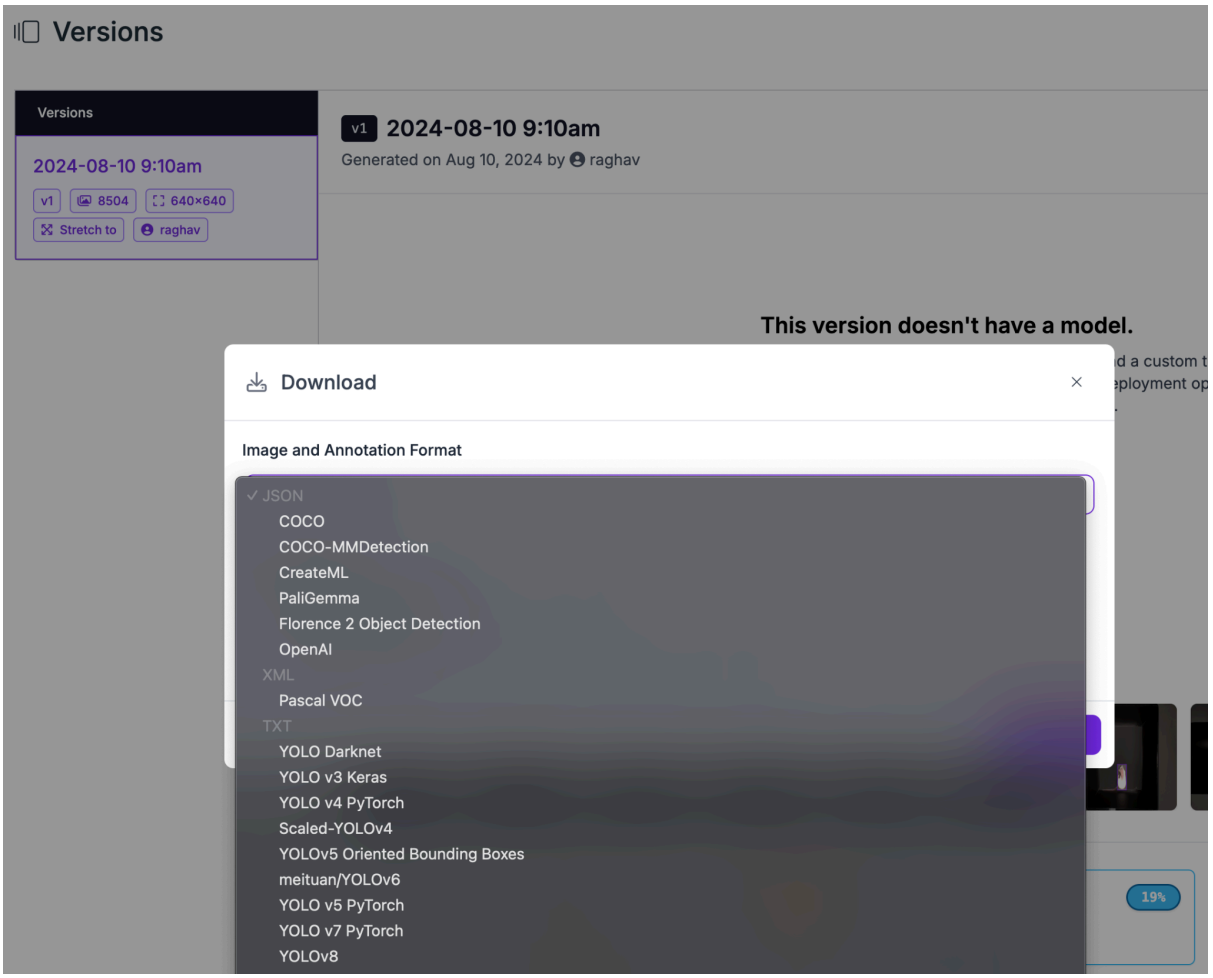


The augmentations we strongly suggest to be used are shown in the image above. Avoid excessive blur or aggressive cropping.

Generate the dataset version after selecting augmentations.

Step 13: Exporting the dataset in COCO JSON format





1. Open the generated dataset version
2. Click Download Dataset
3. Select COCO JSON as the export format
4. Download and extract the dataset archive

This dataset is now ready to be used with Scratcher.

Step 14: Understanding YOLO training outputs in Scratcher

After training a model in Scratcher, the YOLO framework produces plots showing training loss and validation loss.

These plots help determine whether the model is learning correctly.

Interpreting training loss

Training loss represents how well the model fits the training data.

- Training loss should decrease steadily over epochs
- A rapid decrease early in training is normal
- Very low training loss combined with high validation loss may indicate overfitting

---

Interpreting validation loss

Validation loss represents how well the model generalizes to unseen data.

- Validation loss should decrease or stabilize
- Minor fluctuations are expected
- Consistent increase in validation loss indicates poor generalization

---

How to judge model quality

A good model typically shows:

- Decreasing training loss
- Decreasing or stable validation loss
- No large divergence between training and validation curves

A problematic model may show:

- Training loss decreasing while validation loss increases
- Both losses remaining high
- Highly unstable validation loss

In such cases, improve annotation consistency, add more data, or rebalance classes.

Troubleshooting and common issues

This section describes common problems users encounter while creating datasets in Roboflow or training models in Scratcher, along with clear explanations of why they occur and how to fix them.

Issue 1: The model does not learn and losses stay high

Description:

- Training loss remains high across epochs

- Validation loss remains high
- No clear downward trend in either plot

Likely causes:

- Insufficient amount of annotated data
- Incorrect or inconsistent annotations
- Poor class balance

How to fix:

- Annotate more frames, especially frames containing Itch behavior
- Ensure that the same behavior is labeled consistently across videos
- Check that all three classes are being used correctly
- Avoid labeling ambiguous frames as Itch

Issue 2: Training loss decreases but validation loss increases

Description:

- Training loss steadily decreases
- Validation loss increases or fluctuates upward
- Model performs well on training data but poorly on new data

Likely cause:

- Overfitting

Why this happens:

- The model memorizes the training data instead of learning general patterns
- This often happens when the dataset is too small or too repetitive

How to fix:

- Increase the number of annotated frames
- Include videos from different animals, lighting conditions, and camera angles
- Reduce excessive data augmentation
- Verify annotation consistency

Issue 3: Validation loss is unstable or highly noisy

Description:

- Validation loss fluctuates strongly between epochs
- No clear trend is visible

Likely causes:

- Inconsistent annotation rules
- Very small validation set
- Class imbalance

How to fix:

- Review annotation logic for all classes
- Ensure the validation set is at least 20 percent of the data
- Add more examples of underrepresented classes

Issue 4: The model predicts scratching when no scratching is present

Description:

- Scratcher reports Itch during grooming, rearing, or post-scratch behavior

- False positive itch events are frequent

Likely cause:

- Incorrect labeling of the Itch class

Most common annotation mistake:

- Labeling the downward paw movement after scratching as Itch

Correct behavior reminder:

- Itch must only be labeled when the paw is approaching or contacting the nape
- All post-scratch movements must be labeled as Others

How to fix:

- Revisit annotated frames around scratching bouts
- Correct any frames where post-scratch behavior was labeled as Itch
- Retrain the model after corrections

Issue 5: The model misses scratching events

Description:

- Scratching occurs in the video but is not detected
- False negatives are common

Likely causes:

- Too few Itch examples in the dataset
- Very short scratching bouts
- Low visual contrast around the paw

How to fix:

- Annotate more frames containing scratching behavior
- Include multiple examples of short and long scratching bouts
- Avoid aggressive cropping or blur augmentations

Issue 6: Scratcher behaves strangely or timing appears incorrect

Description:

- Predictions appear shifted in time
- Scratching events appear longer or shorter than expected

Likely cause:

- Frame rate mismatch

Explanation:

- Scratcher assumes exactly 30 frames per second
- Any dataset extracted at a different frame rate will cause temporal misalignment

How to fix:

- Confirm that frames were extracted at exactly 30 frames per second
- If unsure, re-extract frames from the original videos at 30 frames per second
- Reannotate and retrain if necessary

Issue 7: COCO JSON export fails or cannot be loaded

Description:

- Export fails in Roboflow
- Scratcher cannot read the dataset

Likely causes:

- Incomplete dataset generation
- Interrupted download
- Incorrect export format

How to fix:

- Ensure the dataset version finished generating successfully
- Export only in COCO JSON format
- Re-download and fully extract the archive
- Do not rename files inside the dataset folder

Issue 8: One class dominates predictions

Description:

- Most frames are predicted as a single class
- Other classes are rarely detected

Likely cause:

- Severe class imbalance

How to fix:

- Check the number of annotated frames per class
- Add more annotations for underrepresented classes
- Avoid oversampling one behavior type during annotation

General best practices to avoid issues

- Always extract frames at exactly 30 frames per second

- Use only the three defined classes
- Follow class definitions strictly
- When unsure, label as Others rather than Itch
- Prioritize annotation consistency over speed
- Regularly review annotations during the process