PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Automated Analysis of Rat Behavior Using Deep Learning and Spatio-Temporal Visualization

Bernardo Luiz Bach

PROJETO FINAL EM GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO – CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Ciência da Computação

Rio de Janeiro, Novembro de 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Bernardo Luiz Bach

Automated analysis of rat behavior using deep learning and spatio-temporal visualization

Final Project

Final Project presented to the Computer Science Course of PUC-Rio in partial fulfillment of the requirements for the degree of Bachelor in Computer Science.

Advisor : Prof. Alberto Barbosa Raposo
Co-advisor: Dr. Jan Jose Hurtado Jauregui

Rio de Janeiro
November 2024

Acknowledgments

I would like to express my deepest gratitude to the many people who have supported me throughout this journey.

First, to my parents and my brother, your love, encouragement, and constant belief in me have meant the world. To my closest friends, for always being there for me, and to my girlfriend Karina, for her unconditional support and for motivating me when I needed it most.

I'm deeply grateful to my advisor Alberto Raposo for accepting this project idea, his guidance and putting me in contact with my co-advisor Jan Hurtado who helped me so much throughout the whole project.

I would like to also express my gratitude to Professors J. Landeira-Fernandez and Thomas Krahe for giving me the opportunity to work with projects from their lab. A special thanks to Talita Clerc for her help and collaboration along the way.

Abstract

Bach, Bernardo Luiz; Barbosa Raposo, Alberto (Advisor); Hurtado Jauregui, Jan Jose (Co-Advisor). **Automated analysis of rat behavior using deep learning and spatio-temporal visualization**. Rio de Janeiro, 2024. 35p. Final Project – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This project presents a multi-stage computational framework to streamline the analysis of rat behavior in conditioning experiments, a common procedure in neuroscience and behavioral research. Traditional manual analysis of video-recorded sessions, which document rats' responses to conditioned stimuli, is labor-intensive and prone to error. Our approach leverages deep learning to automate this process, enhancing both efficiency and accuracy in behavioral assessments. In the first stage, we use deep learning-based methods to segment key rat body parts and detect the rearing posture across video frames. To train these models, we developed a novel semantic segmentation dataset, enabling the use of CNN-based architectures with supervised learning. Next, our method extracts spatio-temporal descriptors from the segmented frames, allowing for precise quantification of behavior over time. In the final stage, we generate visual representations of these descriptors, creating a comprehensive view of behavior patterns such as freezing, rearing, and grooming. This method not only reduces the manual workload but also provides a robust, data-driven approach to understanding complex behavioral responses in animal models, opening avenues for more consistent, large-scale behavioral research.

Keywords

Rat; Semantic segmentation; Deep learning; Spatio-temporal descriptor.

Resumo

Bach, Bernardo Luiz; Barbosa Raposo, Alberto; Hurtado Jauregui, Jan Jose. **Análise automatizada do comportamento de ratos utilizando aprendizagem profunda e visualização espácio-temporal**. Rio de Janeiro, 2024. 35p. Projeto Final – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho apresenta um método computacional baseado em múltiplas etapas para simplificar a análise do comportamento de ratos em experimentos de condicionamento, um procedimento comum em pesquisas de neurociência e comportamento. A análise manual tradicional de sessões de vídeo, que documentam as respostas dos ratos a estímulos condicionados, é trabalhosa e sujeita a erros. Nossa abordagem utiliza aprendizado profundo para automatizar esse processo, aumentando tanto a eficiência quanto a precisão nas avaliações comportamentais. Na primeira etapa, utilizamos métodos baseados em aprendizado profundo para segmentar partes-chave do corpo dos ratos e detectar a postura de rearing ao longo dos quadros de vídeo. Para treinar esses modelos, desenvolvemos um novo conjunto de dados de segmentação semântica, permitindo o uso de arquiteturas baseadas em redes neurais convolucionais (CNN) com aprendizado supervisionado. Em seguida, nosso método extrai descritores espaço-temporais dos quadros segmentados, permitindo a quantificação precisa do comportamento ao longo do tempo. Na etapa final, geramos representações visuais desses descritores, criando uma visão abrangente de padrões comportamentais como freezing, rearing e grooming. Este método não apenas reduz o esforço manual, mas também oferece uma abordagem robusta, orientada por dados, para compreender respostas comportamentais complexas em modelos animais, abrindo caminho para pesquisas comportamentais mais consistentes e em larga escala.

Palavras-chave

Rato; Segmentação semântica; Aprendizado profundo; Descritores espaço-temporais.

Table of contents

List of figures

List of tables

1
Introduction

Behavior analysis of rats is an experimental procedure widely used in neuroscience, psychology, pharmacology, and genetics that serves as a vital tool for biomedical research [1]. Rats are used for these experiments because their genetics are similar to human genetics, and they do not produce expensive costs in experimentation. Different experiments are considered to analyze drugs, diseases, biological mechanisms, genetics, psychological conditions, or any external factor that can affect the entity's behavior.

One of these experiments involves placing a rat inside a conditioning chamber on two different occasions. The animals reside in their home cages and are brought into the chamber for the first session, during which they roam freely for 8 minutes. Afterward, they are subjected to 3 unavoidable shocks, each spaced 10 seconds apart, followed by an additional 3-minute period in the chamber. Finally, the rat is returned to its home cage. This initial session is intended to condition the animal to associate the new environment with danger.

The second phase of the experiment takes place 24 hours later, where the rat is returned to the conditioning chamber for another 8 minutes but without exposure to any aversive stimuli.

For some studies, the primary focus is on the animal's behavior during this second session, as it reflects how well the rat learned from the previous exposure to the chamber and how effectively it associated the context with pain and fear.

Some animals exhibit prolonged periods of immobility, which is referred to as "freezing behavior"—a recognized anxiety-related response [2]. Others engage in a higher frequency of actions such as grooming and rearing, which is often observed in animal models of TDHD and hyperactivity [2]. Multiple hypotheses can be derived from the study of these animal models. The effects of different substances can be tested to assess behavioral changes or damage to certain neuronal structures, which can be induced to determine the functions of those structures. These are examples of what can be tested using this type of procedure.

All events in this experiment are recorded on video and manually

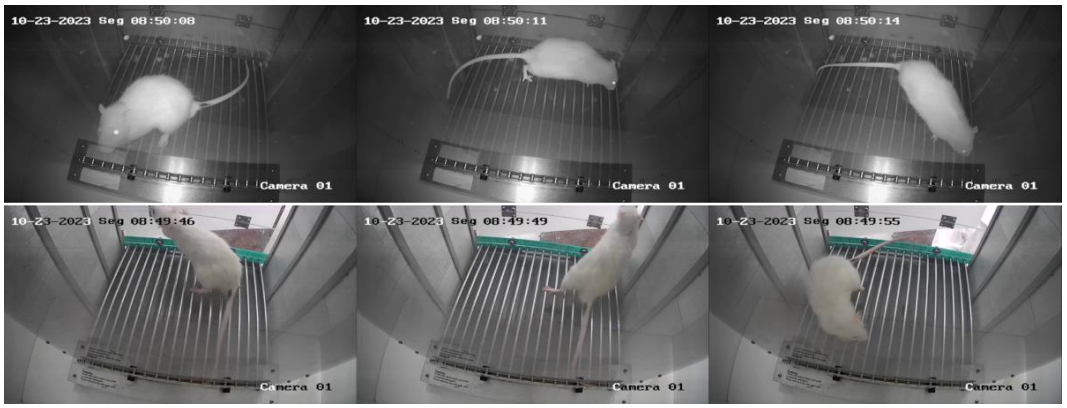

Figure 1.1: Sampled frames from two recorded videos. The first row represents a video generated using night vision. The second row represents a video generated using normal settings.

analyzed by specialists to report rat behavior. Figure 1.1 illustrates some of these videos, where we can see the rat within the chamber. Although these videos represent important documentation for the experiment, their manual analysis requires significant effort and is prone to human error. In this context, automated tools can assist this analysis, making it faster while minimizing possible errors.

In this project, we propose a multi-stage method that aims to support the rat behavior analysis in the specific experiment described above by providing visual information that is helpful for understanding and comparing different behaviors. The first stage of our method consists of segmenting rat body parts and identifying the rearing pose in all the video frames using deep-learning-based approaches. For these approaches, we create a novel semantic segmentation dataset that allows us to train encoder-decoder segmentation models considering supervised learning. The second stage processes these predictions to estimate spatio-temporal descriptors that are useful for defining rat movements. Finally, in the third stage, we compute visual representations to support the behavior assessment.

The rest of the document is structured as follows. Chapter 2 presents some related work relevant to our method. Chapter 3 describes the proposed methods in full detail. Chapter 4 shows some experimental results. Chapter 5 concludes this final project.

2
Related work

The analysis of laboratory rat behavior has been conducted for many decades. Software tools were developed to assist researchers in generating data from animal observations [3], but these tools often demand extensive manual input, requiring researchers to spend long hours manually analyzing animal behavior.

Since the late 20th century, various methods for automating this process have been developed using a broad range of computational vision tools that can facilitate animal behavior recognition, including techniques such as bounding boxes [4, 5], key point tracking [4, 6], thermal image segmentation [7], marker tracking [8] and in more recent years markerless tracking [6]

While early attempts to address this challenge relied on hand-coded heuristics, recent years have seen the predominance of computational models leveraging machine learning and deep learning [4, 6, 5, 7]. One of the earliest studies to automate rat behavior analysis with the use of machine learning was published by Rousseau et al. more than 20 years ago, in which they proposed the use of image processing techniques alongside neural networks [9].

DeepLabCut [6] is a state-of-the-art open-source tool that performs markerless tracking and pose estimation through deep learning and transfer learning. Its accuracy is competitive with both proprietary and other open-source alternatives [10].

For this project, we adopted a different approach to rat behavior analysis by utilizing semantic segmentation with a U-Net-based model. This technique allows us to precisely identify and separate from the background the interesting elements in an image, such as the rat's head, body, the initial part of the tail, and the platform on which the experiment took place.

To the best of our knowledge, no publications have been found that employ semantic segmentation for rat behavior analysis.

3
Method

We propose a multi-stage approach to assist in assessing rat behavior in neuroscience experiments. Specifically, we analyze video-recorded sessions where a single rat is placed in a controlled cage environment. The objective is to track the rat movement within the video and generate visual descriptors that facilitate behavioral interpretation. In the first stage, we focus on segmenting the rat body parts in each video frame. Additionally, since identifying whether the rat is in a rearing pose (standing on hind legs) is critical for behavioral analysis, we classify each frame to determine if the rat is in this posture. Both tasks are addressed using deep learning-based methods. In the second stage, we extract spatio-temporal descriptors from the video and the predictions of the models to capture key aspects of the rat movement. Finally, in the third stage, we produce visual summaries to support the analysis of behavioral patterns. The following sections provide a detailed description of each stage and the datasets used for our experiments.

3.1
Raw dataset

The dataset used in this study originates from INCog, a multidisciplinary neuroscience research group at PUC-Rio, specifically from the Behavioral Neuroscience Laboratory (LANEC), where they study animal behavior using rats.

All videos were recorded within a closed box, blocking all external light. Inside this box lies a conditioning chamber measuring 25 cm x 20 cm x 20 cm, with walls made of transparent material and a floor composed of multiple steel rods spaced 1.5 cm apart, illuminated only by a red light. The rats are all white and of similar size. The camera is placed adjacent to the conditioning chamber and mounted on the wall of the outer box. The camera angle is inconsistent throughout the videos due to the setup construction. As a result, sometimes, the camera is unable to capture the entire chamber.

During the procedure, one rat is placed inside the chamber, and for 15 minutes, it can freely move, and its behavior is recorded.

The dataset contains 81 videos with durations ranging from 5 minutes

to 3 hours. From these videos, we selected 24 to train data-driven models and the other 57 to test our algorithms.

3.2 Supervised learning datasets construction

In the first stage of our method, we aim to segment and classify the frames of the video using deep learning models to describe the rat body and if it is in a rearing pose. Appropriate datasets are essential to enable the training of these models under a supervised learning strategy for the target tasks. For constructing a semantic segmentation dataset, we first define the key structures of interest, specifically certain body parts of the rat that facilitate tracking its motion and orientation while simplifying the annotation process. These structures include the head, trunk, and base of the tail. Additionally, to track the rat location within the cage, we include the cage platform as a structure of interest. For the definition of a possible rearing pose, we consider a binary classification problem, where we need samples where the rat presents this position and samples where it does not.

3.2.1 Frame selection

We select a set of representative frames from the training videos, ensuring variety in the rat locations and poses. For this, we use a custom interactive tool that allows us to navigate through the video and export frames of interest in image format. This selection process also aims to achieve a balanced representation of frames with and without the rearing pose.

3.2.2 Annotation tool

We developed a sketch-based contour drawing annotation tool for delineating structures of interest on the selected frames. This tool enables users to draw and edit closed contours for specific structures with intuitive deformation interactions. Additionally, users can adjust standard window/level parameters for clearer visualization and use zoom and translation functions to focus on the target structure. The contours are saved as dense, high-resolution polygons within the image space. This tool also allows the user to label the selected frames, enabling the definition of whether the rat is in a rearing pose. A capture of this tool is shown in Figure 3.1, where we can see a selected frame and its corresponding annotations.

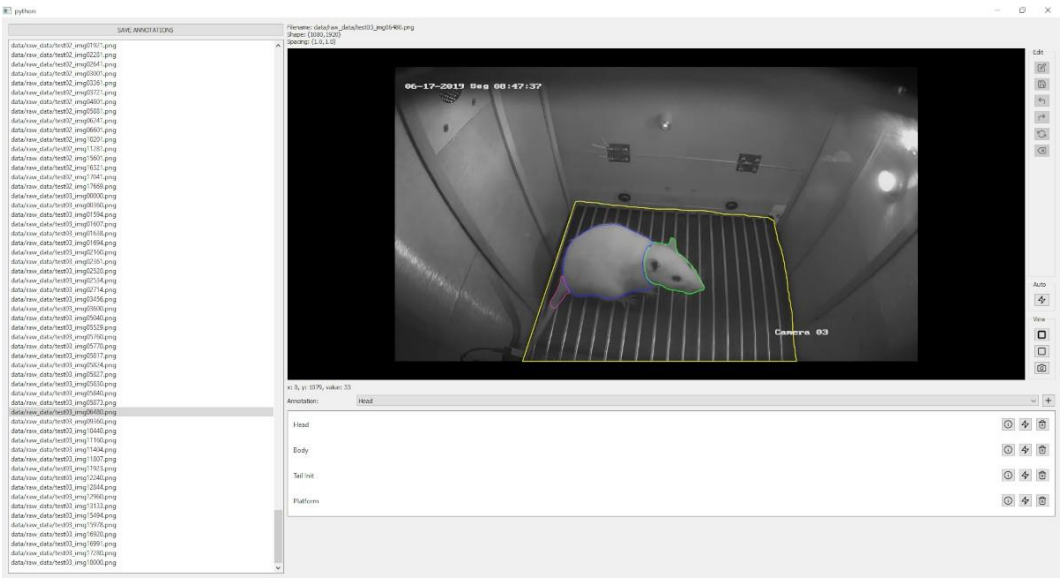

Figure 3.1: Annotation Tool.

3.2.3 Dataset export and split

Using the polygon annotations, we create multi-class label maps that assign a single structure of interest for each pixel. For this generation, we rasterize the annotated polygons into a single map considering the following operations: (1) fill the full label map with the background class label. (2) rasterize the platform structure polygon on the previous label map. (3) rasterize the trunk structure polygon on the previous label map. (4) rasterize the head structure polygon on the previous label map. (5) rasterize the tail init structure polygon on the previous label map. Then, these label maps can be converted to the probability maps expected as the outputs of the segmentation deep learning model by using one-hot encoding. An example of this label map is shown in Figure 3.2.

For the rearing pose labels, we create a one-hot encoding vector to indicate the presence or absence of the rearing pose. This vector assigns a value of 1 if the rat is in the rearing pose and 0 if it is not.

The samples are randomly split into training and validation sets, representing 70% and 30% of the full samples, respectively. The validation set is used to guide the training process and prevent overfitting. The same split is used for both problems, semantic segmentation and classification.

Figure 3.2: Dataset sample. Top: image. Bottom: label map. The platform structure is colored in yellow, the trunk structure is colored in blue, the head structure is colored in green, and the tail init structure is colored in magenta.

3.2.4 Annotation process

The complete annotation process was conducted by two annotators, following some rounds of experimental trials to refine and align the annotation methodology. A total of 728 frames were selected for annotation from 24 different videos. All frames were fully annotated using the custom annotation tool, and the corresponding segmentation and classification labels were exported as described. After splitting the data, we obtained 510 samples for the training set and 218 samples for the validation set. The training and validation sets contain 40% and 37% of frames with the rearing pose, respectively, ensuring a balanced representation of this behavior.

3.3 Stage 1: Deep learning-based analysis

In this stage, we propose using two deep learning models to analyze the video frames in a data-driven manner, leveraging their ability to automatically learn complex features and patterns relevant to our tasks. This approach

enhances both the accuracy and robustness of the frame analysis.

3.3.1 Semantic segmentation of rat's body parts

We aim to segment the body parts of the rat within the full video frames by using a deep learning model that follows a supervised learning scheme. We decided to use a semantic segmentation strategy because the experiment presents a repetitive environment with the presence or absence of a single common entity. We consider a U-Net [11] architecture with an InceptionResNetv2 [12] model as a feature extractor (backbone) for the segmentation task. The network input is a 3-channel image with dimensions 384×384 and intensity values ranging from -1 to 1. The network outputs a $384 \times 384 \times C$ per-pixel probability map, where C represents the number of classes, including an implicit background class for unannotated pixels. Since the segmentation task is approached as a multi-class per-pixel classification problem, the final layer employs a softmax activation function. For training, a hybrid loss function combining Categorical Focal Loss and Jaccard Loss is used. The training configuration includes a batch size of 4, a learning rate of 10^{-4}, and a maximum of 500 epochs, with early stopping applied using a patience of 50 epochs. During training, we use an aggressive data augmentation procedure that considers the following parameters. Rotation range: 90º, shift range: 25%, zoom range: 25%, random horizontal flip, brightness range: 50%, shear range: 5º, channel shift range: 150.

Figure 3.3 shows segmentation prediction examples on multiple frames of a test video. Notice that in most cases, the segmentation of the body parts seems to be accurate and consistent. The platform presents some noise; however, it is a static object, and our idea is to post-process the multiple segmentations to obtain a single representation.

Let us denote the predicted probability maps for each structure of interest by $\mathbf{P}_{\text{plat}}$, $\mathbf{P}_{\text{head}}$, $\mathbf{P}_{\text{trunk}}$, and $\mathbf{P}_{\text{tail}}$, where each one corresponds to the platform, head, trunk, and tail, respectively. By thresholding these predictions, we can obtain the binary masks $\mathbf{M}_{\text{plat}}$, $\mathbf{M}_{\text{head}}$, $\mathbf{M}_{\text{trunk}}$, and $\mathbf{M}_{\text{tail}}$, representing the platform, head, trunk, and tail, respectively.

3.3.2 Rearing pose classification

In addition to the segmentation of rat's body parts, we also use a deep learning model under a supervised learning strategy for the rearing pose classification. To avoid the inclusion of the full frame information, we focus

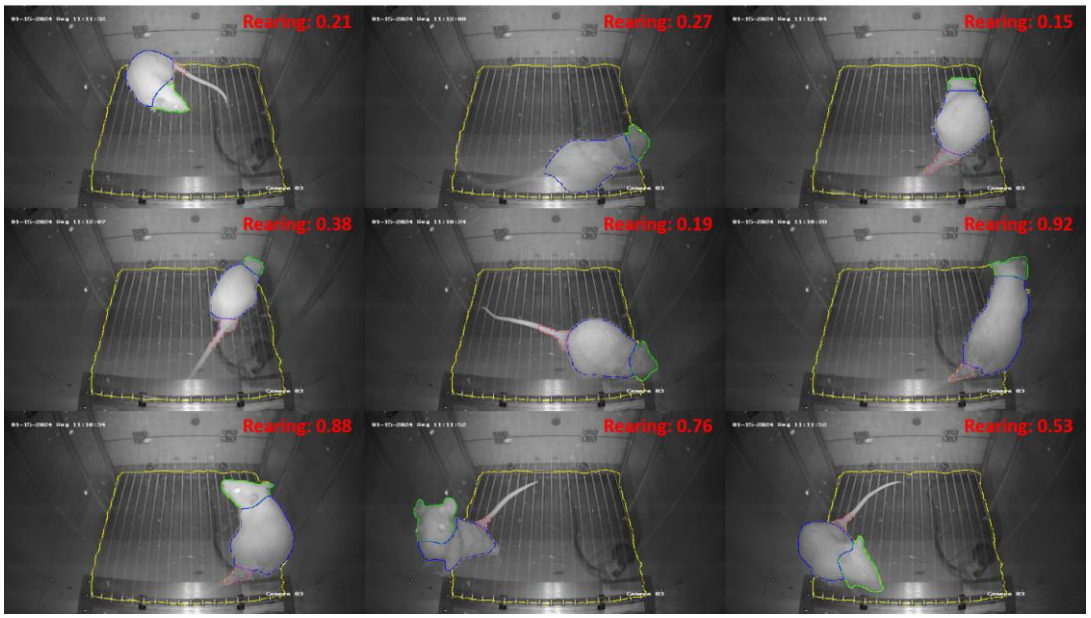

Figure 3.3: Structures of interest and rearing pose classification predictions on different frames of a test video.

on a region of interest defined by the segmentation of the trunk and head of the rat for this task. More precisely, during training, we use the rat's trunk and head segmentation label maps to define a dilated bounding box that captures a region around these structures. The dilation of the bounding box is controlled by a random offset between $0.05h$ and $0.25h$ pixel units, where h is the height of the frame. During inference, we use a bounding box based on the segmentation model predictions $\mathbf{P}_{\text{head}}$ and $\mathbf{P}_{\text{trunk}}$ with a fixed offset equal to $0.15h$.

We consider a EfficientNet-B0 [13] as the model for the rearing pose classification task with a single neuron in the final layer combined with a sigmoid activation function. The output for each sample is a single probability value defining if it presents a rearing pose or if it does not. The model input is a 3-channel image with dimensions 128×128 and intensity values ranging from -1 to 1. For training, we use a binary cross entropy loss function, batch size of 32, a learning rate of 10^{-4}, and a maximum of 500 epochs, with early stopping applied using a patience of 50 epochs. In this case, we use less aggressive data augmentation that includes a rotation range of 5º, a shift range of 5%, a zoom range of 10%, a random horizontal flip, a brightness range of 50%, a shear range of 5º, and a channel shift range of 150. Let us denote the predicted probability value as a rearing pose score s_{rear}. Figure 3.3 shows in red the rearing pose scores predicted over some frames of a test video.

3.3.3
Prediction on video

The two models are applied to the target video frames, considering a downsampling ratio to make the full video processing faster. More precisely, we uniformly sample a frame every 0.5 seconds, reducing the number of processed frames considerably. Thus, for the next stages, assume that we have the corresponding predictions for the uniformly sampled frames and consider a new frame rate for every computation. All the estimations presented in the next sections can be linearly interpolated to match the original resolution.

3.4
Stage 2: Spatio-temporal descriptors

In this stage, we compute descriptors that describe the video in the spatial and temporal domains. All these computations are performed considering time-based video downsampling and the resized frames for the segmentation model, i.e. 384×384 images.

3.4.1
Platform estimation

First, we estimate a quadrilateral shape to approximate the cage platform region. For this computation, we uniformly sample k frames from the full video and apply the structures of interest segmentation model over them, obtaining a set of platform binary masks $\mathbf{M}_{\text{plat}}^{(i)}$, where $i \in \{1, ..., k\}$. Then, we compute an average platform segmentation $\mathbf{M}_{\text{plat}}^{\mu}$ across the time as follows:

$$\mathbf{M}_{\text{plat}}^{\mu} = \left(\frac{1}{k} \sum_{i=1}^{k} \text{chull}\left(\mathbf{M}_{\text{plat}}^{(i)}\right) \right) > 0.5, \tag{3-1}$$

where the function chull($\mathbf{M}$) computes the Convex Hull binary image for an input binary image $\mathbf{M}$. The summation is a numerical operation, treating true values as one and false values as 0 rather than a logical operation. This averaging process is based on identifying persistent regions classified as the platform while minimizing interference from other structures and assuming convexity priors for the platform's shape, as expected in the given environment. Figure 3.4 shows in red an example of a platform mask $\mathbf{M}_{\text{plat}}^{\mu}$ estimated for the full video.

We estimate the contours of the mask $\mathbf{M}_{\text{plat}}^{\mu}$, represented as polygons, and select the largest one as the representation for the platform boundaries. Then, we use the Douglas–Peucker algorithm to fit a quadrilateral to this contour, generating a new polygon with four vertices, which represent the four corners

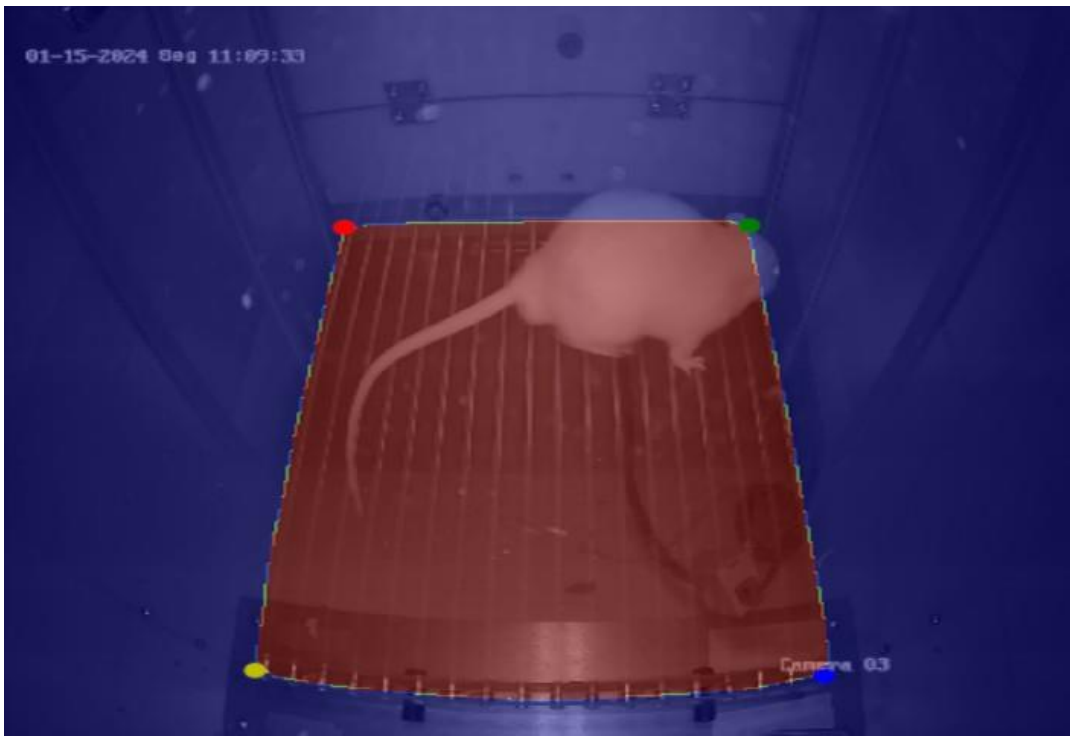

Figure 3.4: Estimation of the platform reference points. The red transparent shapes is $\mathbf{M}^{\mu}_{\text{plat}}$, and the colored points are the estimated corner points.

of the platform. Based on their position in the video, the corners are sorted to achieve a standard representation that allows us to match them with the corners of the real platform square.

Let us denote the sorted corners found on the video as $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, and $\mathbf{p}_4$. We match them with the set of sorted points $\mathbf{p}^{\pi}_1$, $\mathbf{p}^{\pi}_2$, $\mathbf{p}^{\pi}_3$, and $\mathbf{p}^{\pi}_4$ that represent the corners of the platform in the real space, following a 2D representation. With these correspondences, we find a homography matrix $\mathbf{H}$ using the least squares method. This matrix allows us to project any point from the video space to the real 2D space, enabling measurements on a real scale. More precisely, the corners $\mathbf{p}^{\pi}_1$, $\mathbf{p}^{\pi}_2$, $\mathbf{p}^{\pi}_3$, and $\mathbf{p}^{\pi}_4$ define a square with sides length equal to 20 mm, which are the real measurements of the platform.

3.4.2 Positions projection

The segmented masks $\mathbf{M}_{\text{head}}$, $\mathbf{M}_{\text{trunk}}$, and $\mathbf{M}_{\text{tail}}$ on all the frames are spatial descriptors of the rat occupancy in the video space. Similarly to object detection approaches, these shapes enable us to approximate the position of the rat through time. We compute an approximated position for the three rat body parts by processing and picking the centroid of the corresponding mask. For the frame and structure mask, we select the centroid of the largest

connected component if this component presents a considerable area (50 pixel units in the video space). Otherwise, we left the position for the corresponding structure mask empty. At the end of the full positions computation, i.e. over all the frames of the video, we interpolate the values of those that are empty. Additionally, we apply three iterations of a smoothing Gaussian filter with $\sigma = 2$ to the estimated positions on the temporal domain. These operations allow us to obtain smooth transitions and minimize the presence of outliers in the full estimated positions.

Let us denote any of the estimated positions for each frame and for each structure as the point $\mathbf{x}$, which is defined in the video space. To obtain its projection $\mathbf{x}^{\pi}$ in the real space defined by the platform quadrilateral, we apply the following:

$$\mathbf{x}^{\pi} = \mathrm{cc}\,(\mathbf{H}(\mathrm{hc}(\mathbf{x}))), \tag{3-2}$$

where the function hc($\mathbf{x}$) converts $\mathbf{x}$ to homogeneous coordinates and the function cc($\mathbf{x}'$) converts $\mathbf{x}'$ from homogeneous coordinates back to Cartesian coordinates. Thus, with this processing, we obtain the reference positions $\mathbf{x}^{\pi}$ for each rat body part and for each video time step in the real space. These coordinates can be used to compute direction vectors that describe where the rat is looking. For a given position $\mathbf{x}^{\pi}_{\mathrm{trunk}}$ for the trunk and a given position $\mathbf{x}^{\pi}_{\mathrm{head}}$ for the head, we can estimate the unit direction vector equal to $(\mathbf{x}^{\pi}_{\mathrm{head}} - \mathbf{x}^{\pi}_{\mathrm{trunk}})/|\mathbf{x}^{\pi}_{\mathrm{head}} - \mathbf{x}^{\pi}_{\mathrm{trunk}}|$.

3.4.3 Descriptors

Using the projected positions, segmentation predictions, rearing pose classification predictions, and the video content, we create some descriptors useful to assist the analysis of the rat behavior. It is possible that the rat is not present in all the frames in some videos. For this reason, we create a presence descriptor $\mathbf{d}_{\mathrm{pres}}$ with scalar values in the range [0, 1], where 0 indicates the rat absence and 1 indicates its presence. For each frame, the descriptor value d_{pres} is defined as follows:

$$d_{\mathrm{pres}} = \begin{cases} 1, & \text{if area}(\mathbf{M}_{\mathrm{head}} \vee \mathbf{M}_{\mathrm{trunk}}) > 10 \\ 0, & \text{if otherwise} \end{cases}, \tag{3-3}$$

where the function area measures the area of a given mask in pixel units.

We use the positions of the trunk projected in the real space to define a spatial descriptor that represents the positions of the rat through time. Let us define this descriptor as $\mathbf{D}_{\mathrm{pos}}$, which is useful to map the different positions where the rat was in the video. By measuring the displacement of the rat over

time, we can compute a speed descriptor that maps the locomotion of the rat within the cage. It is useful to identify the regions of the video where the rat presented fast displacements. Let us denote this descriptor as $\mathbf{d}_{\text{loc}}$, which maps the rat speed in mm/s multiplied by the values of $\mathbf{d}_{\text{pres}}$. The latter is to define that there is no locomotion in frames where the rat is not present. By thresholding this descriptor, we obtain a piece-wise constant descriptor $\mathbf{d}_{\text{locth}}$ with values in the range $[0, 1]$ that represent the regions of the video where the rat presented considerable motion. We use a threshold value of 5mm/s.

The locomotion of the rat within the cage is important for analyzing its behavior; however, the rat also exhibits various localized movements during activities such as rearing, grooming, and other fine motor actions. These smaller, specific movements provide additional insights into behavioral patterns, stress responses, and general health, complementing the broader analysis of its locomotion. For this reason, we also create a descriptor of local motion based on image differences between neighboring frames in the video. The intuition is that these image differences are potential indicators of object motion in the video when the camera is static. Thus, let us denote as $\mathbf{I}_i$ the current frame and as $\mathbf{I}_{i-1}$ the previous neighboring frame, considering a time distance of approximately 1s. Also, consider two reference masks $\mathbf{M}_i$ and $\mathbf{M}_{i-1}$, which are the union of the corresponding rat's head and trunk masks. For both frames, we define a unique region of interest $\mathbf{M}_{\text{roi}}$ that is the dilation of the union of the reference masks of both frames, i.e. $\mathbf{M}_{\text{roi}} = \text{dilation}\,(\mathbf{M}_i \cup \mathbf{M}_{i-1})$. Then, we compute a difference image $\mathbf{I}_{\text{diff}}$ focused on this region as follows:

$$\mathbf{I}_{\text{diff}} = |\mathbf{I}_i - \mathbf{I}_{i-1}| \odot \mathbf{M}_{\text{roi}}, \tag{3-4}$$

where $\mathbf{M}_{\text{roi}}$ represents the element-wise multiplication. From this difference image $\mathbf{I}_{\text{diff}}$, we select the areas that represent considerable motion by applying a threshold of 5. The latter results in a new local motion mask $\mathbf{M}_{\text{lm}}$ that is post-processed using binary opening, binary closing, and small connected component removal. Finally, considering the local motion mask $\mathbf{M}_{\text{lm}}$, we define a local motion score d_{lm} as follows:

$$d_{\text{lm}} = \min(\text{area}(\mathbf{M}_{\text{lm}})/150, 1). \tag{3-5}$$

By computing these values for every frame of the video, we generate the local motion descriptor $\mathbf{d}_{\text{lm}}$.

Freezing is one of the key actions in rat behavior analysis, as it often indicates fear, anxiety, or heightened attention in response to a stimulus. Recognizing freezing behavior can provide insights into the animal's emotional state, response to environmental changes, or reaction to experimental condi-

tions. By accurately identifying and quantifying freezing episodes, researchers can assess the impact of pharmacological treatments, environmental stressors, or neurological disorders on the rat's behavior, aiding in studies of anxiety and other behavioral or cognitive conditions [2]. To recognize the freezing action, we propose the binary descriptor $\mathbf{d}_{\text{freez}}$, which presents values of 1 in regions of the video where the rat presents the freezing action and 0 otherwise. This descriptor is computed as follows:

$$\mathbf{d}_{\text{freez}} = \text{sccr}((1 - \mathbf{d}_{\text{lm}}) \odot (1 - \mathbf{d}_{\text{locth}}) \odot \mathbf{d}_{\text{pres}}), \tag{3-6}$$

where the function sccr is a small connected component removal operation applied in 1D, used to avoid considering possible freezing regions with short times, i.e. lower than 4s. The intuition of this descriptor is to select those regions where the rat is present; it does not present locomotion in the cage and presents minimal local motion.

Rearing is also a key action in rat behavior analysis, as it reflects exploratory behavior and curiosity, often indicating the rat's interest in its environment or response to novelty. Rearing, where the rat stands on its hind legs, can provide insights into cognitive function, sensory perception, and general activity levels. Tracking rearing behavior helps researchers assess spatial awareness, investigate neurological health, and understand reactions to environmental stimuli, making it an important measure in studies on learning, memory, and anxiety. In the first stage, we compute rearing pose scores for each frame, denoted as s_{rear}. Let us denote the set of full scores as the vector $\mathbf{s}_{\text{rear}}$. Then, the rearing action descriptor $\mathbf{d}_{\text{rear}}$ is defined as follows.

$$\mathbf{d}_{\text{rear}} = \text{smooth}(\mathbf{s}_{\text{rear}}) \odot \mathbf{d}_{\text{pres}} \odot (1 - \mathbf{d}_{\text{freez}}), \tag{3-7}$$

where the function smooth applies two iterations of a Gaussian filter with $\sigma = 2$. The intuition is to ignore those classification scores defined in regions where the rat is not present or is presenting a freezing action.

Grooming is another key action in rat behavior analysis, as it serves as an indicator of the rat's physiological and emotional state. Grooming behavior can reflect baseline self-maintenance activities, responses to stress, or reactions to environmental changes. By observing patterns, frequency, and duration of grooming episodes, researchers gain insights into the animal's stress levels, coping mechanisms, and even neural function. Analyzing grooming behavior can thus be crucial for studies on anxiety, depression, neurological disorders, and the effectiveness of therapeutic interventions [2]. Although this action can be difficult to identify in a computational setting, we propose a grooming descriptor $\mathbf{d}_{\text{groo}}$ that indicates regions of the video where we can find high-

frequency local motion similar to that produced in a grooming action. Thus, we compute this descriptor $\mathbf{d}_{groo}$ as follows:

$$\mathbf{d}_{groo} = \mathbf{d}_{lm} \odot (1 - \mathbf{d}_{locth}) \odot (1 - \mathbf{d}_{freez}) \odot (\mathbf{d}_{rear} < 0.5) \odot \mathbf{d}_{pres}, \tag{3-8}$$

where the intuition is to select those regions of the video where the rat is present, it is not presenting a rearing action, it is not presenting a freezing action, it is not presenting locomotion within the cage, but it is presenting considerable local motion.

3.5 Stage 3: Visual analysis

In this stage, we aim to generate visual representations that are useful for assisting the rat behavior analysis. As a first representation, we propose a trajectory map that shows the rat positions through the video in a normalized space with real scale. Thus, we can plot the coordinates of the descriptor $\mathbf{D}_{pos}$ as vertices of a continuous polyline within the target space. Figure 3.5 shows an example of this polyline that represents the trajectory of the rat through the video. Also, using the descriptor $\mathbf{D}_{pos}$, we can generate a heatmap that highlights the regions of the cage where the rat spent the most time. For this, we map the time spent for each coordinate of $\mathbf{D}_{pos}$ onto a regular grid, then apply smoothing to create a more continuous representation of the rat's movement patterns. Figure 3.5 shows an example of this heatmap, revealing a high-density region near the bottom-left corner of the cage, indicating that the rat spent a significant amount of time in this area.

The descriptors $\mathbf{d}_{loc}$, $\mathbf{d}_{locth}$, $\mathbf{d}_{freez}$, $\mathbf{d}_{rear}$, and $\mathbf{d}_{groo}$, are especially useful to recognize actions and patterns in the full video. These descriptors can reduce the effort by making the user focus on specific regions of the video instead of analyzing it frame by frame. Thus, to produce a visual representation for these descriptors, we consider a horizontal color bar plot. Each sample in the horizontal direction represents a frame of the video, while the color maps the corresponding descriptor value. For the bar coloring, we consider a blue-white-red color map that uses the corresponding descriptor maximum and minimum values for normalization. Figure 3.6 shows an example of this plot for the different descriptors. Notice how the red regions in the plot are potential indicators of any of the behaviors of interest.

All of these tools can significantly reduce the effort required by the user in analyzing multiple videos, streamlining the process of behavior assessment, automating time-consuming tasks, and allowing for more efficient data extrac-

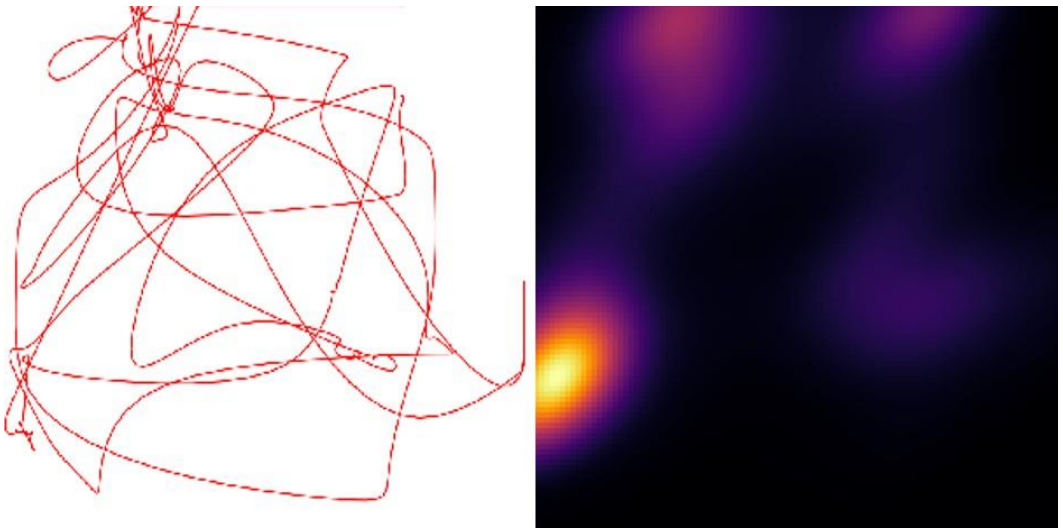

Figure 3.5: Trajectory plot and heatmap example.

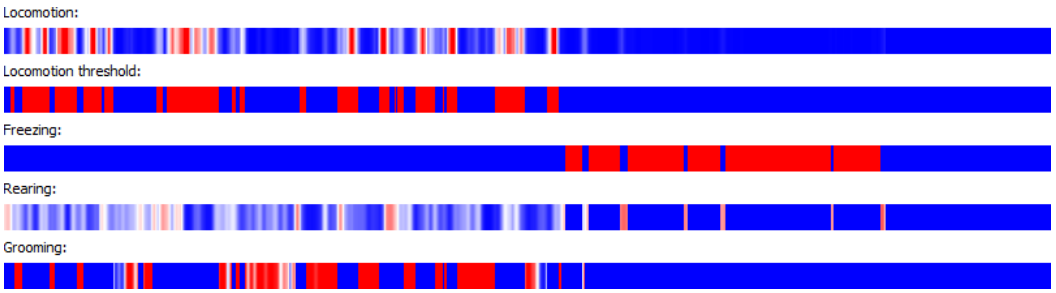

Figure 3.6: Visualization of descriptors.

tion and interpretation.

4
Results

4.1
Training log

Using the parameters specified in the previous section for semantic segmentation, we trained the corresponding deep learning model, considering the validation set to select the best weights. Figure 4.1 shows the loss function evolution on training and validation sets, where it is possible to notice the corresponding convergence. Differently, for the rearing classification, Figure 4.2 shows a more chaotic behavior on the validation set. Although we tried different parameters to make more stable the training procedure, we noticed that this could be caused by two factors: a small dataset and the complexity of differentiating a rearing pose from other poses. We hypothesize that with more annotated and highly representative data, we can achieve more stable and efficient training.

4.2
Numerical results

Table 4.1 presents the numerical results obtained over the validation set in the rat's body parts segmentation problem. We consider six metrics typically used for semantic segmentation tasks: precision, recall, accuracy, Dice coefficient, intersection over union (IoU), and Hausdorff distance in pixel units in the original video space. We show the metric values per-structure of interest and average. From this table, we can see that the tail init seems to be the most challenging structure. In contrast, the head and the trunk present high Dice and IoU values, suggesting that they will be useful for tracking. On the other hand, the platform segmentation seems to be accurate.

For the rearing classification task, we evaluated performance using standard binary classification metrics. Figure 4.3 presents the confusion matrix, where the dominance of true positives (TP) and true negatives (TN) over false positives (FP) and false negatives (FN) is apparent. Figure 4.4 displays the Receiver Operating Characteristic (ROC) curve, with an Area Under the Curve (AUC) of 0.98—indicating near-optimal classifier performance, as 1.0

Figure 4.1: Training log of semantic segmentation model.

Table 4.1: Baseline approach IoU results

Metric	Head	Trunk	Tail init	Plat-form	Mean
Precision	0.7923	0.8978	0.5620	0.9648	0.8042
Recall	0.7627	0.9179	0.6530	0.9551	0.8221
Accuracy	0.9974	0.9957	0.9987	0.9805	0.9931
Dice	0.7673	0.9059	0.5920	0.9594	0.8062
IoU	0.7746	0.8984	0.6034	0.9227	0.7998
Hausdorff	39.8167	42.4154	38.2144	102.4516	58.0611

represents a perfect model. Additionally, the classification model achieved an accuracy of 0.94, precision of 0.91, recall of 0.93, and F1-score of 0.92, all of which suggest strong overall performance. Despite these promising results, there remains room for further improvement.

4.3 Visual results

Figure 4.5 shows some rat's body segmentation prediction results on selected samples of the validation set. We can see that the model closely approximates the ground truth annotations, confirming the numerical values presented above. Also, we can see that the model seems stable and does not present structures that are too noisy. However, in the last row sample, we can see that the model misses the prediction of the rat tail init.

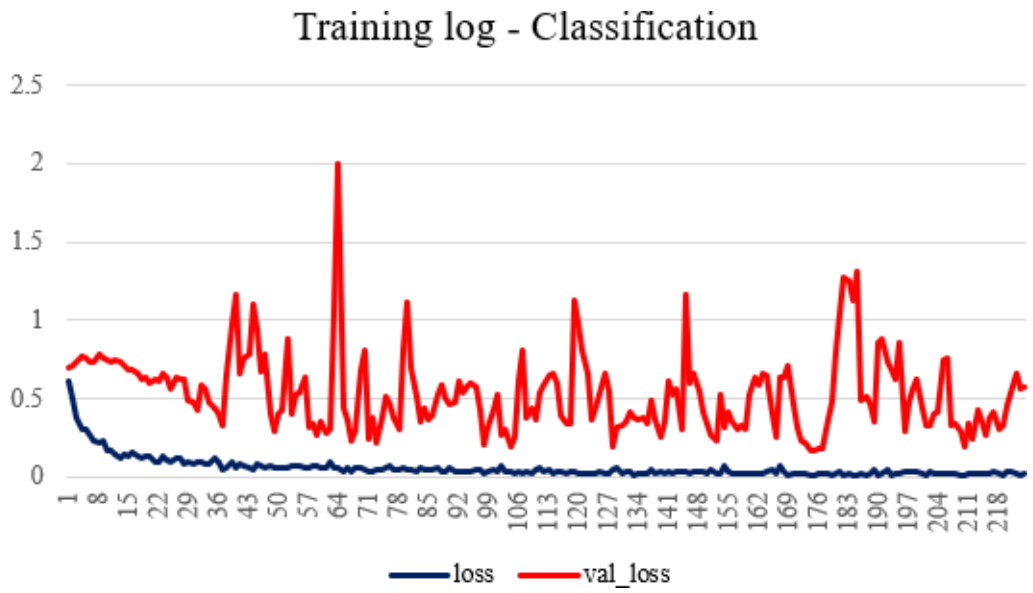

Figure 4.2: Training log of classification model.

4.4 Implementation details

4.4.1 Programming language

We chose to use the Python programming language due to its simplicity and vast machine learning libraries, allowing us to concentrate on the problem solving aspect of this complex theme.

4.4.2 Libraries

For this image processing and segmentation tasks, we chose various Python libraries to optimize performance and usability.

For machine learning and GPU acceleration, it integrates TensorFlow and Keras for model building and training, while ONNXRuntime ensures compatibility with different machine learning frameworks and improves inference speed. We also used pre-trained architectures from the Segmentation Models library to support efficient segmentation workflows.

Data manipulation and numerical operations are handled by Pandas and NumPy, with Matplotlib and Scikit-learn used for visualization and evaluation of model performance.

The graphical user interface is built using PyQt and VTK, allowing users to interact with the software easily and visualize data.

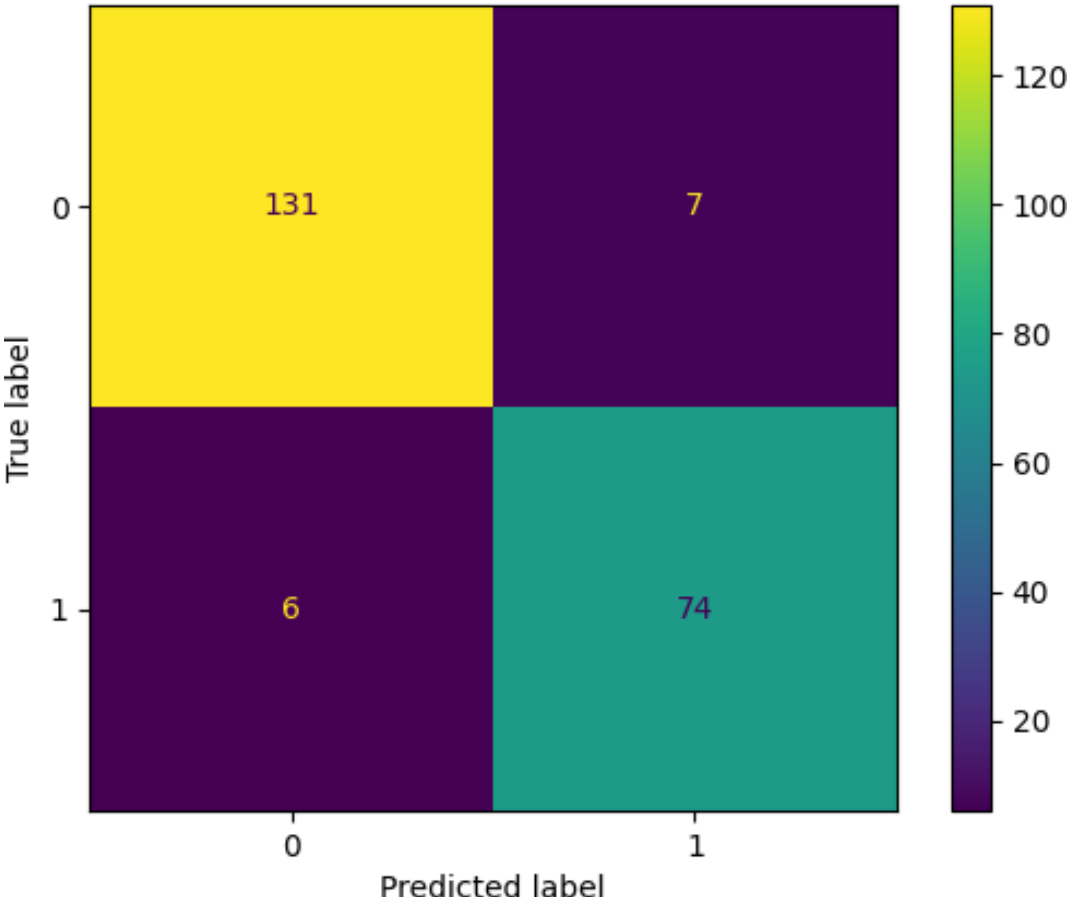

Figure 4.3: Confusion matrix.

We relied on OpenCV (cv2) and scikit-image to manage image processing and transformations.

4.4.3 Annotator

For this project, an annotation tool was developed to facilitate the task of labeling images from the dataset. This tool enables users to draw and classify the contours of four regions of interest in the image: the rat's head, trunk, the base of the tail, and the platform. Additionally, it was employed to classify images for rearing pose estimation.

The annotator includes features that allow users to adjust lighting, contrast, and exposure, as well as zoom and pan across the image. These adjustments are not saved and are intended solely to aid in better visualizing the image during the annotation process. Each annotation is recorded as a set of coordinates within the image, forming dense, high-resolution polygons for each of the four previously mentioned structures.

Annotations are saved in CSV format, with each row containing the

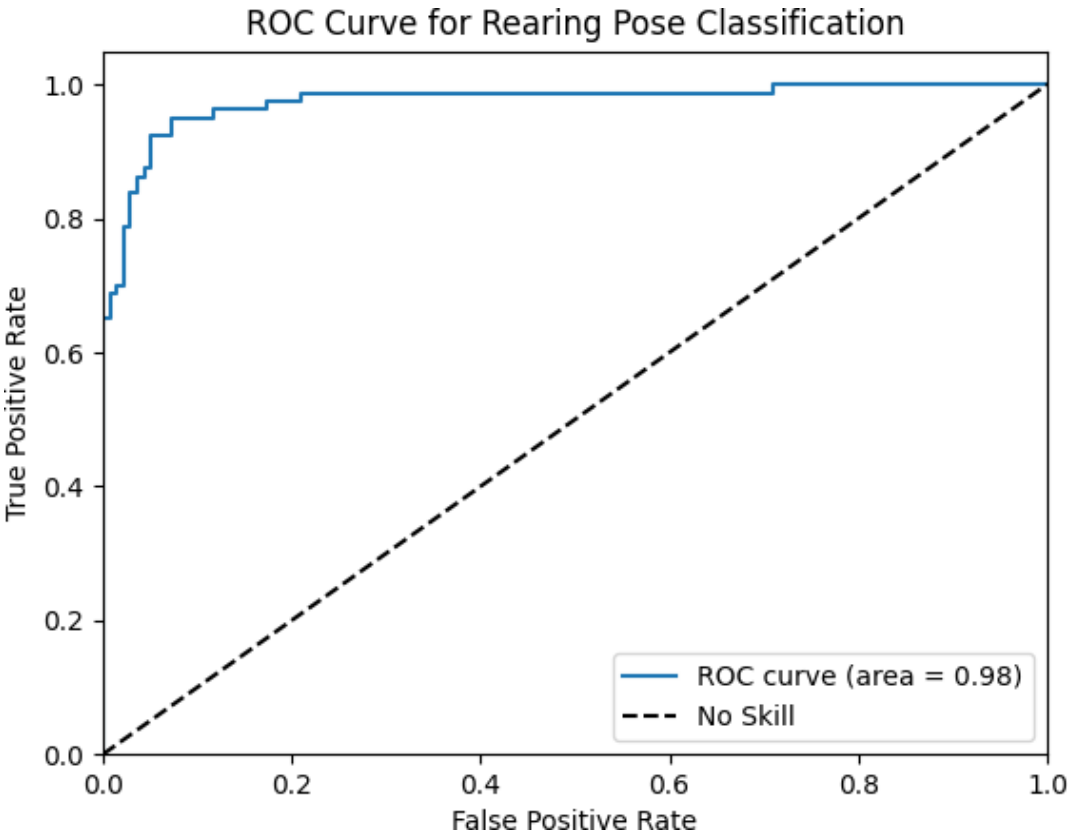

Figure 4.4: ROC Curve.

following fields: 1) Instance ID: an identifier matching the image with a unique annotation; 2) Image ID: the original image file name; 3) Annotation ID: the label identifier for the annotated region; 4) Data: the coordinates of the drawn polygon; and 5) Time and Date.

4.4.4 Exporting the dataset

The processing of the raw dataset occurs alongside that of the annotations. First, the raw dataset is randomly split into training (70%) and validation (30%) groups. From the annotation data, the coordinates of each polygon are used to apply a rasterization algorithm, creating a multi-class map for each structure of interest present in the original image. Finally, the image is resized to 384×384, resulting in a labeled dataset for our segmentation model.

4.4.5 Training Script for Segmentation

The training process begins by checking for the availability of GPUs. If no GPUs are available, the system defaults to using the CPU. A batch generation process follows, where both unprocessed and labeled images, resized during

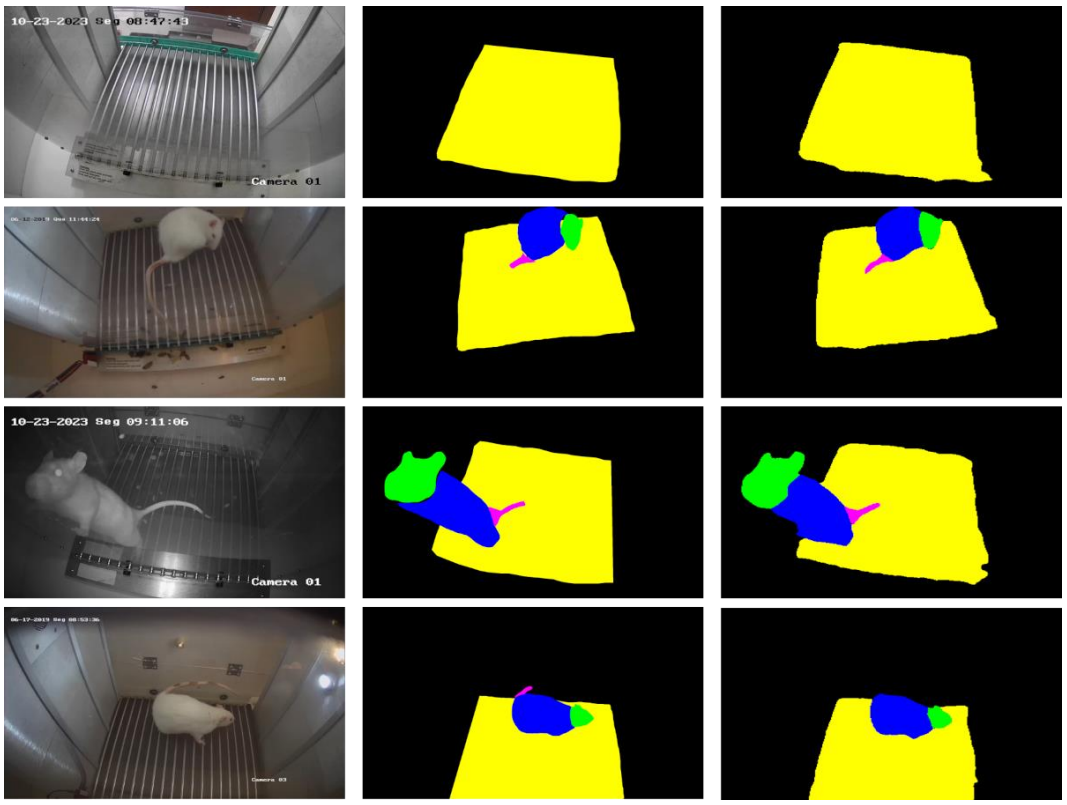

Figure 4.5: Visual results on some samples of the validation set. First column: input image. Second column: ground truth annotation. Third column: prediction.

a previous data manipulation phase, are retrieved to form the training and validation sets. Data augmentation occurs during this batch generation, where the training subset is subjected to rotations up to 90º, horizontal and vertical shifts, zoom, horizontal flipping, brightness variations, shear deformation, and channel shifting.

For the training, we opted for a U-net architecture segmentation model combined with InceptionResNetV2 as a feature extractor. To achieve pixel-wise classification for multi-class segmentation tasks, a softmax activation function was selected.

Although the U-net architecture is widely used in image segmentation problems, it has a significant limitation in its ability to learn more complex aspects of the model [14]. By incorporating the Adam optimizer, we aim to mitigate these limitations and further focus the deep learning model on the region of interest (ROI) in the images [15].

For model fitting, we implemented early stopping set to 50 epochs based on the validation loss. Model checkpoints were used to save weights during training, and logs of the loss value and Intersection over Union (IoU) scores were recorded. The training runs for a maximum of 500 epochs but may stop

earlier if early stopping is triggered.

4.4.6 Training Script for Binary Classification

The training script begins similarly to the segmentation training, with a check for GPU availability. The dataset used consists of images and their corresponding classifications. The augmentations applied to the images are largely the same as in the previous task, with the exception that the rotation is limited to 5º. This is because the task involves classifying rearing behavior, where the rat stands up, making orientation a key variable. To further enhance the model's focus on relevant features, a region of interest (ROI) extractor was implemented to create a bounding box around the rat, limiting the region processed in each training batch.

The model used for this task is a convolutional neural network (CNN) with a sigmoid activation function. The backbone of the model is EfficientNetB0, chosen for its lightweight architecture, as a more robust feature extractor is unnecessary for this task. The loss function employed is binary cross-entropy, appropriate for measuring a binary classification output. Early stopping, model checkpointing, and logging mechanisms are the same as in the previous setup, with the training set to run for a maximum of 500 epochs.

4.4.7 Evaluation

To evaluate the performance of the segmentation model, we compare the images labeled by the annotator with those labeled by the model. Six metrics are used to assess the model's effectiveness: Intersection over Union (IoU), Dice coefficient, accuracy, precision, recall, and the Hausdorff distance.

To calculate these metrics, we iterate through each pair of images, consisting of the original annotated image and the corresponding model-generated prediction. For each pair, we evaluate the performance for each class individually.

Binary classification, on the other hand, requires a different evaluation approach and set of metrics. For this task, we compute accuracy, precision, recall, and the F1-score based on the labeled data and the predictions generated by the model.

4.4.8 Demonstration GUI

The demonstration GUI consists of an ethogram composed of five bars with heatmaps, two visualizations of position descriptors and a frame-by-frame video slider. As the user move the slider, the displayed video frame updates, allowing for precise examination of specific moments.

The five horizontal behavior bars correspond to the distinct behavioral categories we are measuring and are color-coded using a heatmap gradient, where color intensity represents the intensity of each behavior, with red indicating high activity and blue indicating low.

1. Locomotion: represents when the animal move and the intensity of its movement is color coded from blue to white to red. Its calculated by the difference in pixel area of the head and trunk masks combined in between two consecutive frames.

2. Locomotion Threshold: A binary classification representing whether the rat is moving a significant amount or just subtle movements. We can use this metric to help determine if the rat is performing grooming or freezing actions.

3. Freezing: A few set of hand coded heuristics are used to characterize freezing. If the rat has a low locomotion score and its movement is bellow the locomotion threshold for more than five seconds its considered freezing behavior.

4. Rearing: heatmap calculated directly from the output of the classification model

5. Grooming: Also coded as heuristics the grooming behavior is measured by the amount of localized movement while there is no freezing, rearing or locomotion.

There are two graphs containing the position heatmap and the trajectory the rat moved during the test. The trajectory is drawn by interpolating the position of the animal every frame.

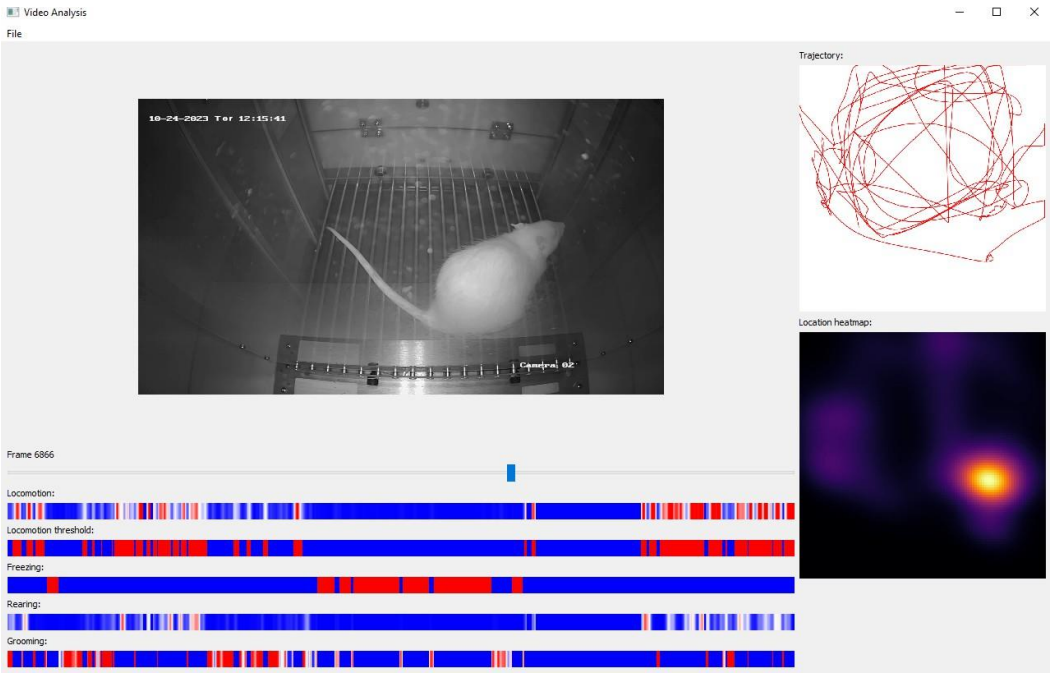

Figure 4.6: GUI capture.

5
Conclusion and future work

The automatization of rat tracking and behavior analysis represents a significant advancement in the field of neuroscience and behavioral research. In this work, by leveraging modern computer vision techniques and machine learning algorithms, we present an automated system that offers improved accuracy, efficiency, and objectivity over traditional manual methods, allowing for consistent, high-throughput data collection and analysis.

A key component in this advancement is the use of image segmentation that allows precise detection and isolation of rats from complex backgrounds, ensuring more reliable tracking and behavioral categorization. The U-Net based model, with its strong ability to capture fine details and spatial hierarchies, significantly enhances the accuracy of segmentation, even in challenging environments. This leads to more accurate identification of specific behaviors, such as grooming, rearing and freezing which are essential for behavioral analysis.

This technology accelerates research while opening new opportunities for studying a wider range of behaviors and conditions, ultimately contributing to a better understanding of neurological diseases, cognitive functions, and the effects of interventions. As this system continues to evolve, we believe the potential for scalability and application in diverse experimental environments can further enhance behavioral research.

Bibliography

[1] VAN MEER, P.; RABER, J.. **Mouse behavioural analysis in systems biology**. Biochemical Journal, 389(3):593–610, 2005.

[2] DE FREITAS, T. D. S. C.. **Caracterização Comportamental dos Ratos Cariocas com Baixo Congelamento: Uma Avaliação de um Potencial Modelo de TDAH**. PhD thesis, PUC-Rio, 2024.

[3] TEJADA, J.; CHAIM, K. T. ; MORATO, S.. **X-plorat: A software for scoring animal behavior in enclosed spaces**. Psicologia: Teoria e Pesquisa, 33, 2017.

[4] CHEN, Z.; ZHANG, R.; FANG, H.-S.; ZHANG, Y. E.; BAL, A.; ZHOU, H.; ROCK, R. R.; PADILLA-COREANO, N.; KEYES, L. R.; ZHU, H. ; OTHERS. **Alphatracker: a multi-animal tracking and behavioral analysis tool**. Frontiers in Behavioral Neuroscience, 17:1111908, 2023.

[5] NILSSON, S. R.; GOODWIN, N. L.; CHOONG, J. J.; HWANG, S.; WRIGHT, H. R.; NORVILLE, Z. C.; TONG, X.; LIN, D.; BENTZLEY, B. S.; ESHEL, N. ; OTHERS. **Simple behavioral analysis (simba)–an open source toolkit for computer classification of complex social behaviors in experimental animals**. BioRxiv, p. 2020–04, 2020.

[6] MATHIS, A.; MAMIDANNA, P.; ABE, T.; CURY, K. M.; MURTHY, V. N.; MATHIS, M. W. ; BETHGE, M.. **Markerless tracking of user-defined features with deep learning**. arXiv preprint arXiv:1804.03142, 2018.

[7] MAZUR-MILECKA, M.; RUMINSKI, J.. **Deep learning based thermal image segmentation for laboratory animals tracking**. Quantitative InfraRed Thermography Journal, 18(3):159–176, 2021.

[8] MAGHSOUDI, O. H.; TABRIZI, A. V.; ROBERTSON, B. ; SPENCE, A.. **Superpixels based marker tracking vs. hue thresholding in rodent biomechanics application**. In: 2017 51ST ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS, AND COMPUTERS, p. 209–213. IEEE, 2017.

[9] ROUSSEAU, J.; VAN LOCHEM, P.; GISPEN, W. ; SPRUIJT, B.. **Classification of rat behavior with an image-processing method and a**

neural network. Behavior Research Methods, Instruments, & Computers, 32:63–71, 2000.

[10] BÜHLER, D.; POWER GUERRA, N.; MÜLLER, L.; WOLKENHAUER, O.; DÜFFER, M.; VOLLMAR, B.; KUHLA, A. ; WOLFIEN, M.. **Leptin deficiency-caused behavioral change–a comparative analysis using ethovision and deeplabcut**. Frontiers in neuroscience, 17:1052079, 2023.

[11] RONNEBERGER, O.; FISCHER, P. ; BROX, T.. **U-net: Convolutional networks for biomedical image segmentation**. In: MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION–MICCAI 2015: 18TH INTERNATIONAL CONFERENCE, MUNICH, GERMANY, OCTOBER 5-9, 2015, PROCEEDINGS, PART III 18, p. 234–241. Springer, 2015.

[12] SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V. ; ALEMI, A.. **Inception-v4, inception-resnet and the impact of residual connections on learning**. In: PROCEEDINGS OF THE AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, volumen 31, 2017.

[13] TAN, M.; LE, Q.. **Efficientnet: Rethinking model scaling for convolutional neural networks**. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 6105–6114. PMLR, 2019.

[14] RAMESH, S.; KANCHANA, J. S. ; DAVID NEELS PONKUMAR, D.. **Hybrid u-net and adam algorithm for 3dct liver segmentation**. In: 2023 7TH INTERNATIONAL CONFERENCE ON I-SMAC (IOT IN SOCIAL, MOBILE, ANALYTICS AND CLOUD) (I-SMAC), p. 752–757, 2023.

[15] JABER, M. M.; ABD, S. K. ; ALI, S. M.. **Adam optimized deep learning model for segmenting roi region in medical imaging**. In: PROCEEDINGS OF INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND INTELLIGENT SYSTEMS: ICETIS 2021 VOLUME 2, p. 669–691. Springer, 2022.