AUTOMATED RECOGNITION OF RODENT SOCIAL BEHAVIOR

Malte Lorbach

Malte Lorbach - Automated Recognition of Rodent Social Behavior

AUTOMATED RECOGNITION OF RODENT SOCIAL BEHAVIOR

MALTE LORBACH

Cover: The three Rs – Replacement, Reduction and Refinement. Defined by W. Russell and R. Burch in 1959 under the title "The principles of humane experimental technique", the three Rs are the most important ethical guidelines for designing and conducting animal research. They encourage the use of research methods that replace animal testing by alternative techniques, reduce the number of animals, and minimize pain and suffering.

Automated Recognition of Rodent Social Behavior
PhD thesis, Utrecht University, the Netherlands

ISBN 978-90-393-6872-5
COVER Malte Lorbach
PRINT ProefschriftMaken || www.proefschriftmaken.nl

AUTOMATED RECOGNITION OF RODENT SOCIAL BEHAVIOR

AUTOMATISCHE HERKENNING VAN SOCIAAL GEDRAG VAN KNAAGDIEREN

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof.dr. G.J. van der Zwaan, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op woensdag 8 november 2017 des middags te 12.45 uur

door

MALTE THORBEN LORBACH

geboren op 2 juni 1986
te Osterode am Harz, Duitsland

PROMOTOR Prof. dr. R. C. Veltkamp
COPROMOTOR Dr. ir. R. W. Poppe

This thesis was partly accomplished with financial support from the European Research Council under the European Union's Seventh Framework Programme, ERC grant agreement no. 317259.

Contents

1 Introduction

Measuring social behavior of rodents is a key process in various research disciplines. In neuroscience, for example, rodent social behavior is studied to understand the pathology and development of neurological conditions such as Huntington's disease. Social behavior is also relevant when monitoring animal welfare as the absence of social interaction can indicate an unhealthy living environment.

The purpose of measuring behavior is to be able to compare it, either to some desired state (e.g., a healthy environment) or among animal populations (e.g., healthy versus diseased). This requires quantifying the behavior objectively which is typically achieved by counting occurrences of specific actions and interactions, or timing their durations and frequencies. The basis of such quantification is the annotation of the behaviors in either live observations or video recordings.

The level of detail of the annotations determines how rich the resulting measurements are. For example, counting occurrences yields only one measure per observation while annotating start and end times of each interaction also allows measuring the duration and frequency. Naturally, there is a trade-off between the level of detail and the time it takes a human to annotate at that level, which may range from approximately three to twelve times the length of the observation [3, 18, 145].

In general, annotating videos manually is a time-consuming task and it is desirable to alleviate some of the manual effort by automating it. Besides saving time, automated annotation is consistent, produces replicable output and can operate for extended periods of time without suffering from fatigue. Its efficiency allows reanalyzing previous data under new hypotheses, saving both time and animals. Automation therefore contributes directly to the goal of the three Rs [118], the ethical guideline for animal research which aims to replace animal testing by alternative techniques, reduce the number of animals, and refine existing methodology. One of the main limitations of current automated methods is their rigidity with respect to varying environmental and experimental factors. In comparison, humans have exceptional perception and interpretive skills which allow them to abstract from irrelevant environmental factors. These skills still seek to be matched by machines.

The first automated methods for behavior measurements focused on quantifying movements and location preferences of single rodents by tracking the rodent's location in a video recording. These methods allowed for measures such as the distance traveled or the time spent outside the shelter [121, 135]. Later on, advances in digital image processing led to better recognition of the visual information in the videos which allowed to annotate the start and end times of specific actions such as walking and rearing. To recognize and annotate rodent actions in videos, an algorithm needs to compute for each video frame which action is performed. This is a classification task that can be addressed by using a computational model that distinguishes actions based on input features. The features are an abstraction from the visual information in the video and are typically derived from the tracked locations, such as velocity, or directly from the video image, such as body shape. This approach works well for measuring single animal behavior but it lacks the ability to annotate *interactions* among rodents.

Automatically annotating social behavior brings new challenges. First and foremost, the locations of multiple, visually similar rodents have to be tracked. This is particularly difficult as the animals move quickly and are similar in appearance. Although tracking remains to be a challenging task, recent advances in video analysis led to reductions in the number of errors made [113]. Now the tracking of multiple rodents is more robust and accurate, which allows us to take the next step and consider the automated annotation of specific interactions.

We continue with an explanation of the research context in Section 1.1 and the focus of our work in Section 1.2. A summary of the contributions of this thesis is presented in Section 1.3.

1.1 Research Context: PhenoRat

This research was initiated by the EC FP7 Marie Curie *PhenoRat* project. In this project four graduate fellows, two with a neuroscience background and two with a computer science background, collaborated to study the social behavior of rats. The two neuroscientists, Elisavet Kyriakou and Giuseppe Manfré, investigated rat models for Spinocerebellar Ataxia type 17 [74–76] and Huntington's disease [85], respectively. For behavior analysis automated measurement methods were used. Developing and improving these automated methods was the main focus for the computer scientists. The work involved investigations of novel algorithms for tracking the locations of multiple rats in video as well as the automated classification

of rat social behavior based on those location trajectories. The latter is the focus of the research presented in this thesis.

PhenoRat was managed by University Tübingen and Noldus Information Technology bv in cooperation with Utrecht University and Radboud University Medical Center Nijmegen. It ended on October 31, 2016. The work presented here was conducted as part of the *PhenoRat* project and continued afterward at Utrecht University.

1.2 Scope of this Thesis

Our main goal is to enable neuroscientists and biologists to quickly annotate videos of rodent social behavior experiments with support of automated annotations. Our focus lies on the efficient training of the involved classification model. We approach this problem as an interactive annotation task in which the researcher, the user of our method, starts annotating examples of the relevant interactions while an algorithm learns to distinguish the interactions from those examples. Once the algorithm has learned sufficiently, it can propagate the annotations throughout the remaining videos automatically and thereby alleviate the user from much of the manual effort. Typically, manual annotations are obtained by labeling interactions sequentially as they occur in the video. This may not be the best way for the algorithm to learn however as it may take a long time until examples of rare interactions are encountered. To avoid the time-consuming task of manually searching for such examples in the videos, we instead aim at finding potentially informative examples automatically. The task for the user is then reduced to labeling the selected examples.

To be able to investigate and develop an interactive annotation method, we first require a better understanding of the technical aspects that play a role in automated annotation of rodent interactions. Therefore, we analyze various aspects with respect to their influence on the quality of the automated annotations in Chapters 4 to 6. We then leverage these insights in our interactive method which we present in Chapter 7.

For most of our work, we assume that the animal locations have been tracked with a suitable algorithm that maintains the identities of the subjects. As tracking errors are still common in practice, a systematic examination of the effect of these errors on the annotation accuracy is part of our work (Chapter 5). We do not develop novel learning or classification algorithms but use established, off-the-shelf methods. In fact, we generally treat the classifier as a black box that may be replaced by another suitable algorithm in practice.

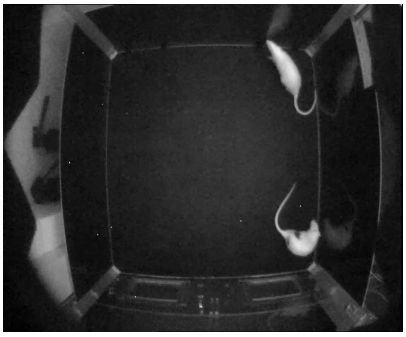


Figure 1.1: Two rats in observation cage recorded from top view perspective.

We apply the presented work to social interactions between two rats that can freely move within a controlled, confined space (the observation cage) as shown in Figure 1.1. Social behavior is stimulated by a short (24-48h) social isolation period – an experiment often conducted by neuroscientists to study the social behavior of rats with a neurological condition [45, 130]. We do not consider experiments in which the movement of one or more animals is constrained as for example in the three-chamber approach test [91]. This choice is the result of the collaboration with the neuroscientists in the PhenoRat project (Section 1.1).

The interactions of the rats in the experiments include inspection behavior in which the rats repeatedly approach each other, sniff their partners and then retract. Following, social grooming, and play behavior is also observed. Although we demonstrate our work primarily on one particular set of interactions, we do not tailor our methods towards this set. In particular, we refrain from meticulous tuning of methods for the purpose of increasing accuracy measures and attempt to draw conclusions regarding the general nature of interactions rather than specific categories. In Chapter 3 we discuss in more detail the characteristics of rodent interactions performed by different species, including rats and mice.

1.3 Contributions of this Thesis

We present our work on automatic annotation of rodent social interactions in video. Our goal is to develop an interactive annotation method that learns to recognize interactions from user-provided examples and so reduces the manual annotation effort otherwise required. We make several contributions to achieve this goal, which we summarize below. Unless oth-

erwise specified, all contributions have been made by Malte Lorbach, the author of this thesis.

1. We first introduce a new research dataset (RatSI) that allows us to study rat social behavior recognition. The dataset includes manual annotations which serve as a ground truth for learning the classification model and evaluating the automatically generated annotations. It comprises nine videos with a total duration of 135 min. Since there is no other rat social behavior dataset publicly available yet, we make RatSI available to the research community. The videos and annotations were obtained by Elisavet Kyriakou for a social interaction test within the PhenoRat project. Compilation, processing and publication of the data was done by the author. (Chapter 3)

2. We systematically analyze the characteristics of rodent social behavior in two datasets (RatSI and CRIM13) and thereby enhance our understanding of the challenges of automated annotation (Chapter 4). We discuss topics related to learning, applying and evaluating behavior classification models.

   We first examine how often and for how long specific interactions occur in the considered datasets. We then analyze how the they are annotated by human observers and whether the interactions show aspects of ambiguity that lead to variance in the labeling. Finally, we explore various features that are essential for distinguishing the considered interactions automatically, such as the relative pose and distance, and show how they can be extracted from the animal trajectories. This yields a basic feature set applicable to a wider range of rodent behavior data.

3. The challenging task of tracking the locations of multiple, similar rodents may introduce errors in the data that we learn from and that we use to distinguish interactions. In Chapter 5 we systematically analyze to what degree those errors affect the accuracy and whether they affect some interactions more than others. In particular we demonstrate that the current tracking method limits the recognition of interactions that happen in close contact. In this chapter we also introduce our first classification method for social interactions that may serve as a baseline for other work.

4. It is desired that a classification model can be applied to a variety of experiment settings and behaviors without the need to retrain it. Despite the fact that variations in the environment and the behavior can occur unexpectedly and are sometimes beyond our control, the

applicable scope is rarely evaluated in the literature. To study the risk of loss in accuracy that are due to behavior variations, we conduct cross-dataset classification experiments in Chapter 6. An analysis of the classification models highlights the properties that limit the applicable scope, and shows the potential for adaptation and transfer to other settings.

5. In a scenario in which automated annotation is not possible because no suitable classification model is available, the human observer is left with manually annotating videos. Our goal is to alleviate as much of the manual effort as possible by training a new classification model while the observer is annotating. As the model becomes more accurate over time, it may take over the annotation task when it achieves a satisfying performance. The human observer can then stop annotating manually and save valuable time.

   In order to reduce the time until the classifier can take over the annotation, we may guide the human observer to annotate particularly useful or rare examples first. For instance, interactions that are easy to distinguish will only require a few examples until they are recognized accurately, while visually similar cases may require more. In Chapter 7 we present our interactive annotation framework which realizes the above goals and allows the annotation of rodent behavior videos with substantially reduced manual effort. We validate the efficiency of the approach with human annotators in a user study.

1.4 Thesis Outline

We begin with an overview of the literature on measuring rodent social behavior in ethology and behavioral neuroscience, and the automation of such measurements in Chapter 2. We introduce our research datasets in Chapter 3, followed by an analysis of their main properties concerning the recognition of social behavior in Chapter 4. We continue with a systematic analysis of the effects of varying tracking quality in Chapter 5. Cross-dataset application of trained classification models is investigated in Chapter 6. Finally, we develop an interactive annotation framework and evaluate it in a user study in Chapter 7. We conclude with a discussion of our findings and future directions in Chapter 8.

2 Measuring Rodent Social Behavior: an Overview

Research into the social behavior of rodents, be it for experimental or observational purposes, involves comparing behavior among different animal populations or against some desired state. The key to make systematic comparisons is to describe the behavior quantitatively. Traditionally, researchers have formalized the description of behavior by defining categories of behavioral events and then counting or timing their occurrences in either live observations or video recordings [86]. For a quantitative comparison, the occurrences are then analyzed in terms of their frequency or their mean and total durations.

Another form of describing behavior is to measure some aspect of the observed animal numerically such as its velocity or its location. Although such measures are quantitative by definition, they typically do not provide meaningful insights on their own. Instead they need to be combined with other information to derive more meaningful quantities. For example, it could be of interest to use the velocity to derive an activity profile over the course of a day-night cycle [22, 30], or to use the locations to determine the latency until the animal dares to explore an open, unprotected area [11, 117]. Methods that directly measure such continuous aspects of behavior require some form of automation when determining them by hand is imprecise or infeasible (e.g. a velocity profile).

In this chapter, we give an overview of methods for measuring rodent social behavior used in literature. We provide an introduction to traditional, manual measuring methods, although the main body is dedicated to automated methods. We discuss the advantages and disadvantages over manual methods and elaborate on the challenges of automation, on current limitations and open questions. To give a comprehensive overview, we discuss all components involved in automated annotation, although the focus in the thesis lies on learning classification models efficiently with reduced manual effort.

2.1 Manual Measuring Methods

Measuring behavior based on a categorical classification of involves three steps: defining the categories, annotating observations in terms of those

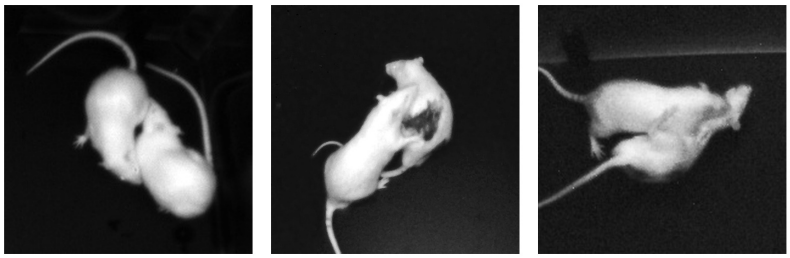


Figure 2.1: Examples of inspection and play behavior.

categories, and analyzing the annotations statistically. Let us look at each step in detail.

2.1.1 *Defining Behavior Categories*

For the traditional annotation methods, the behavior categories are defined by a textual description of what constitutes certain behaviors and how they differ from other, similar behaviors. The description serves as training material for the annotators and as documentation of the methodology behind the behavioral study. A good description is clear, unambiguous and provides a sufficient distinction between the behaviors that are most alike. Any room for interpretation as to whether or not a certain category applies to an observed situation leads to subjectivity in the annotations and is therefore to be avoided [80].

Writing a good behavior definition is a challenging task. As a guide to how behavior may be categorized, Martin *et al.* [86] distinguish between defining categories in terms of either the structure of behavior or its consequences. A description of structure includes the visual and physical appearance with respect to posture and movement of the animal, for example *scratch head with front paws*. A description of the consequences deals with the effects of the behavior on the environment, another animal or the subject itself irrespective of how it is performed specifically. For example, *digging a hole* describes the consequence of the action (a hole that was not there before) without specifying the exact circumstances such as whether front or hind paws are used.

Defining meaningful categories with respect to a particular research question is a process that often involves consulting the literature for standard ethograms and conducting preliminary experiments. By annotating the observations of those experiments and thereby identifying ambiguous and missing information, the behavior definitions can be expanded, im-

proved or rewritten before the actual study is launched. Typical categories used to describe rodent social behavior include *approaching*, *following*, *oral* and *anogenital inspection*, *social grooming* and a variety of play or antagonistic behavior [99, 139, 148]. Figure 2.1 shows some examples.

2.1.2 *Manually Annotating Behavior*

Once the behavior categories are defined, an observer can annotate the rodent's behavior in either live observations or video recordings. In controlled environments, the behavior is often recorded as it allows replaying and watching fast activities in slow-motion to ensure no events are missed. Figure 2.2 depicts two ways in which an annotation can be made [2]: as a *point event* that merely indicates the occurrence of a behavior in time, or as a *state event* that indicates the continuous performance of a behavior from its beginning until its end. The latter allows computing the event duration in addition to the frequency.

Although there are still cases in which annotations are made with pen and paper, for example during field observations, in most laboratory and other controlled environments the use of annotation software is standard practice. Annotation software allows the observer to label events by key strokes on the keyboard while watching the video. Features such as automatically ending a state event once a new one is started, make labeling mutually exclusive and exhaustive categories more efficient. Furthermore, it is often possible to score additional information alongside an event such as who initiates an interaction. Integrated analysis tools simplify the computation of statistics about the annotated observations.

Despite the fact that annotation software increases the efficiency, manual annotation remains a time-consuming task. The time needed depends on the number of categories, the occurrence frequency of the behaviors (several per minute or a few per hour), the level of detail of the annotations (point or state events, and optional information), and the experience

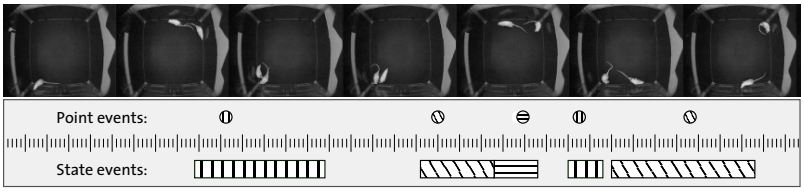


Figure 2.2: The occurrence of a specific behavior can be annotated as either a point in time (point event) or a continuous performance (state event).

of the annotator. The time may add up to twelve times the duration of the video for creating temporally highly accurate annotations [145]. Behavior annotation demands a high level of concentration from the observer over a long period of time. The fatigue that inevitably sets in during long observation sessions can lead to mistakes as events may be missed, confused or inaccurately labeled. Regular breaks are necessary which increases the total annotation time further. The limited capacity of human annotators makes long-term observations of days or weeks virtually impossible unless a team of annotators is employed to work simultaneously.

As discussed in the previous section, the quality of the category definitions is an important factor to mitigate subjectivity in the annotations. But the interpretation of the definitions is not the only reason why two annotators may label the same videos differently [49]. Other factors include preconception about the tested animals, variation in scoring behavior transitions and ambiguity of the video material. Preconceptions can occur if the annotator is aware of the assignment of experiment groups such as a treatment and control group [114]. Scoring the transition between two state events, that is the time when one behavior ends and the following begins, often causes variations as it is difficult to define and detect the exact point in time objectively. Finally, the video recordings can be ambiguous when essential information is hidden due to occlusion, bad lighting, focus or simply due to low video resolution. Clearly, ambiguity is not limited to video recordings and may as well occur in field observations where animals are not always in plain sight and close enough to see.

Technical issues with the recordings can be resolved by improving the recording facilities, for example by installing multiple cameras at different view points. Such solutions come with a higher cost for equipment, installation effort and technical challenges such as time synchronization among cameras. Temporal variations may be reduced by thorough training of the annotators but is likely to remain to some degree. Similarly, preconceptions can be mitigated by hiding the group assignments from the scorer but cannot always be eliminated completely as treatment and disease could leave visible marks on the animals.

Total objectivity of human annotators may be impossible to achieve in practice, but we can take measures to ensure a high quality. Monitoring the quality is a first important step that helps to decide whether we can trust the conclusion drawn from a behavior study [54]. A way to make such a decision is to ask whether we would arrive at the same conclusion if another annotator had labeled the behavior using the same methods. This question relates to the *inter-annotator reliability*. The reliability can be expressed in terms of the percentage of agreement [19] or its chance-corrected version

called Cohen's kappa [23]. In any case, at least two sets of annotations for the same videos from different annotators are required. In order to contain the additional annotation effort, only a subset of the observations is labeled by multiple annotators in some studies and the results are then extrapolated to the rest of the study [59]. Typical figures of the reliability in literature for open field rodent behavior range from 60% to 70% [3, 18, 59]. Note that it is unclear what an acceptable value for the reliability is as the exact figure depends not only on the agreement but also on the occurrence frequencies of the different categories and in particular whether they are skewed or balanced [23]. Moreover, for state event sequences with a beginning and an end, a more complex analysis of agreement that first aligns the state events could in fact be more accurate than percentage agreement scores that do not take the event durations into account [105].

In general, researchers aim at minimizing subjectivity and ambiguity by meticulously defining what a certain behavior constitutes and by using an adequate acquisition setup. Despite this effort, it takes a considerable amount of time to train annotators and to gain sufficient experience as a group of annotators to produce annotations with a high level of agreement. In appreciation of the possibility that some disagreement will always remain, monitoring the disagreement is crucial to behavior research and the refinement of the methodology.

2.1.3 *Analyzing Annotated Behavior*

Once the desired behaviors are annotated, they can be analyzed with respect to the study's research question. The question often concerns the *phenotype* of an animal population which describes behavior that is characteristic for that population. Researchers typically want to explore whether there are significant differences in the phenotypes across populations (e.g., healthy versus diseased). Which measures are considered part of the phenotype depends on the exact research question. Common types include the frequency and the duration of the occurrences of each behavior category and how they change over time (short-term over a day-night cycle or long-term over the entire lifetime). More comprehensive phenotypes involve a temporal analysis of particular behavior sequences as they for example occur in grooming [64] and mating behavior [60]. They may further include properties of group behavior and social hierarchies [129]. Discussing the details of such analyses is beyond the scope of the thesis. We refer to literature from behavioral biology for further reading, for example [79, 86].

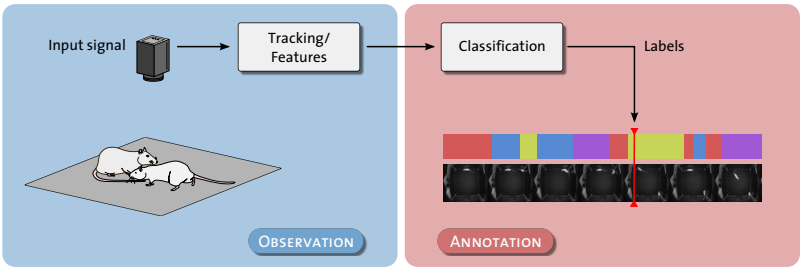


Figure 2.3: Components of an automated annotation system.

2.2 Automatically Annotating Behavior

In the previous section, we discussed the traditional, manual approach to define, annotate and analyze rodent behavior. We highlighted the main limitations of human annotators, namely: subjectivity and annotation time. Automated annotation methods are an attractive alternative to manual labeling. Automation can alleviate much of the manual effort and enable consistent and repeatable analysis of hours of behavior observations. Other than for annotating behavior categories, computers can also be employed for directly measuring kinematic quantities such as velocity, activity or posture. We consider these complementary to the categorical approach and refer to other literature for an overview [34, 40, 109].

In our review of automated annotation methods, we focus on work that considers behavior of freely moving rodents (e. g. open field tests). We do not consider test setups that focus on one specific aspect of behavior such as the elevated plus maze [52] or the three chamber social approach test [94].

Conceptually, automated annotation methods have of two main components as illustrated in Figure 2.3, namely: a) an *observation* component which extracts meaningful information (features) from the input signal such as a video, and b) an *annotation* component which uses a classification algorithm to label the behavior based on the extracted features. Let us examine these components in more detail.

2.2.1 *Observation*

The observation component involves acquiring the input signal, such as a video, and then extracting features from that input signal, for example, the location of the animals or their posture. The input signal can be acquired from a variety of hardware devices and the choice depends on the behavior

that is to be measured. Rearing behavior, for example, can be measured by infrared light-beams that detect when a rodent crosses one of the horizontally arranged rays [62, 91]. Accelerometers detect vibrations that allow for measuring the repetitive motions of grooming behavior [17]. A rough localization can also be achieved with an observation platform equipped with force sensors [51].

For recognizing social interactions, the locations of the animals relative to each other is important information. In literature, methods for determining the animal locations use either radio-frequency identification (RFID) or video cameras. In an RFID setup, every animal carries a small transponder implanted under the skin in the neck. If such a transponder is stimulated by a radio wave with a particular frequency, it returns a unique radio signal. This signal identifies the animal and can be detected by antennas to perceive the animal's presence. RFID setups are often used for larger cages with multiple compartments. By placing antennas at strategic positions such as doors between compartments, it is possible to track the animals' visits to certain areas [46, 56, 81] and to perform operant conditioning tasks [6, 68, 150]. It is also possible to arrange multiple antennas in a grid in the cage floor to enable location tracking within a compartment [147]. However, the physical placement of antennas limits the spatial resolution of the tracked locations to about 10 to 40 cm [147]. The low resolution prevents detecting specific social interactions such as *following* or *inspection*.

The majority of recent work on rodent location tracking relies on using video cameras and computer vision algorithms to track the animals. Video-based methods have the advantage that they are non-intrusive to the animals and are available at low cost. Moreover, video images allow extraction of not only the locations at a high spatial and temporal resolution but also of the animal's posture and appearance which leads to a richer behavior description.

2.2.1.1 *Video-based location tracking*

The tasks of tracking a single rodent and tracking multiple rodents simultaneously in a video have different requirements. Although we consider them to be disjoint tasks, they do share a few basic concepts which we want to elaborate. Generally, the goal is to locate the rodents in every video image. That requires the separation of pixels that belong to an animal from the non-animal pixels. Let us consider the animal pixels to be the *foreground* while everything else, the cage, the floor and other objects, is part of the *background*. The segmentation of foreground and background is typically one of the first steps of any tracking algorithm irrespective of the number of animals.

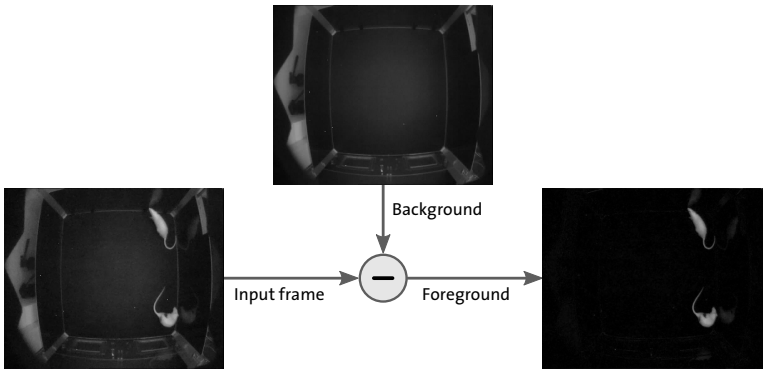
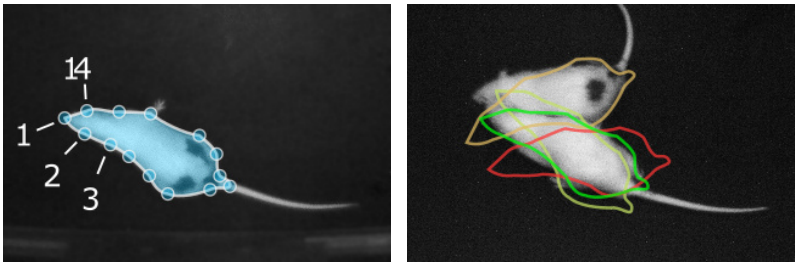


Figure 2.4: A stored background image is subtracted from the video frame to segment the foreground (the animals).

In the simplest case, both camera position and background are static and do not change over the course of a video. Then it is possible to keep a reference image of the background (without animals) and subtract that image from every video frame as illustrated in Figure 2.4. After subtraction only foreground pixels have non-zero values, which separates the animal from the background and allows calculating its location in the image [57]. In practice, the reference image is not always a perfect representation of the background as lighting may change slightly and objects and cage bedding (e. g. sawdust) may shift. Background subtraction then fails to create a crisp separation of the foreground and thus causes inaccurate localization. Small or gradual changes in the background can be addressed by updating the reference image from several past video frames. By averaging over those frames, moving foreground objects are eliminated [37]. It has also proven useful to relax the condition for background pixels after subtraction and also filter pixels whose value is not exactly but close to zero [92, 135]. Finding a suitable threshold for this filtering problem can be challenging as lighting conditions often vary across videos. Other image processing techniques such as blob detection and morphological operations can further improve the results [28].

Background subtraction aims at identifying the foreground by elimination. Because there is no model of the foreground involved, it is suitable for detecting any object that is sufficiently different from the background. If a foreground model is available, though, it may improve the detection as other objects such as feces and shifted bedding become easier to eliminate. Moreover, using a foreground model is also applicable to less controlled environments in which the background is not static such as in field obser-



(a) Training image for a rodent shape model (b) Different model parameters result it good and bad fits in a test image

Figure 2.5: Several key points, encoding local texture and global shape information, form together an appearance model of a rodent.

vations. In rodent tracking, the knowledge of the foreground object (i. e. the rodent) can be utilized to improve the accuracy of the tracking. Then the task is to model the appearance of a rodent and use that model to detect a matching instance in the video image [156].

The appearance of the rodent can be described in terms of the shape of its contour and the intensity values of the enclosed pixels. In the case of rodents, but also for other deformable objects such as human faces or fish, the appearance variations are constrained by the body structure. *Active shape models* (ASMs) [25] exploit this fact and statistically model both global (shape) and local appearance information (texture). The parameters of the model are learned from examples images of the object to find, in our case the rodent. Finding a rodent in a novel image then involves fitting the model to different locations in the image and selecting the instance that best fits the learned parameters as illustrated in Figure 2.5. A greedy search over all parameter combinations and locations is typically intractable and it is more efficient to restrict the search to suitable candidates. Twining *et al.* [141], for example, first perform a rough background subtraction leaving only the rodent and potential artifacts such as reflections as candidates. The search is then initialized with the fit from the previous frame and this hypothesis is iteratively updated until it converges to the rodent's contour. Similarly, de Chaumont *et al.* [31] manually define a shape model based on multiple geometric primitives (e.g., ellipses) as well as their relative displacements, and fit the model to the image using a physics engine.

Note that most rodent behavior experiments are conducted in the animal's active phase during the night. Video recordings are typically monochrome as they are made under near-infrared (red-light) illumination which appears invisible to the rodents. Color information is therefore not available for tracking.

2.2.1.2 *Tracking Multiple Rodents under Occlusion*

Both foreground segmentation and model-based detection work well for tracking the location of single rodents. If multiple rodents are observed, the tracking algorithm has the additional task of separating the individuals and identifying them consistently across video frames. Typically the number of animals present is known. Therefore, in images in which the rodents are far away from each other, finding each individual is conceptually identical to finding a single rodent in some sub-region of the image. However, if they are close, in contact or occlude each other, the separation becomes more challenging.

Approaches that rely on foreground segmentation need to separate the extracted foreground pixels into multiple parts, each corresponding to one rodent. Given that the animals used in experiments are often from the same genetic background, they have the same fur color. When the animals are in contact, their contours can be ambiguous. Information from previous or future frames, for example the optical flow, can indicate different motion directions and thus help separating the animals [16, 63].

Although optical flow may not solve situations with similar motion directions, motion in general is a useful cue for dealing with occlusion. Model-based tracking approaches can benefit directly from using motion cues when they are incorporated in the model. Mayya *et al.* [89] for example define local motion models for several body parts and then use a Kalman filter to follow the most likely trajectory of each animal.

A popular tracking method that combines both appearance and motion cues is *particle filtering* [15, 67, 103]. A particle filter maintains a large number of particles, each representing a hypothesis for a rodent's location. In every video frame, each particle updates its location according to a motion model which describes the typical movement of a rodent from one frame to the next. Because the update is stochastic, each particle moves differently. After the location update, the particles are evaluated based on the current video frame and an observation model which describes the appearance of a rodent. Particles that did not follow the trajectory of a rodent will contradict the observation model and are thus discarded as invalid hypotheses. The remaining particles again update their locations and the next frame is processed. After a few frames, only valid hypotheses that represent the true location of the rodents are left.

The key components of a particle filter are the observation and motion models. These are formulated probabilistically. The observation model allows evaluating a location hypothesis against the current observation using a likelihood function. Branson *et al.* [15] incorporate in the observation model for mice both foreground segmentation features (a rotated and

scaled ellipse) and shape features from contour templates. Their motion model represents typical displacements by an autoregressive model which can be sampled to generate random particle updates. Pistori *et al.* [103] use a random walk model for the same task.

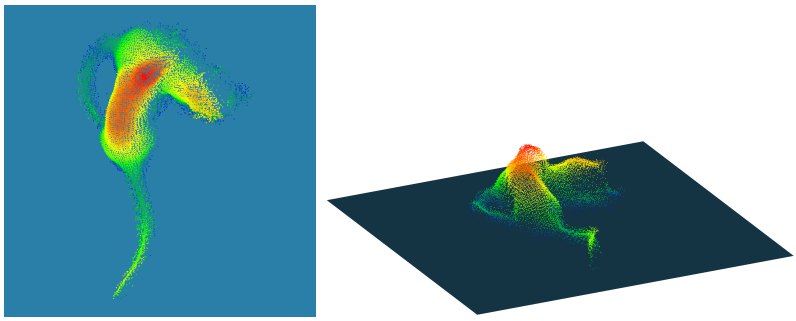
The ability to correctly solve occlusion situations with a particle filter depends predominantly on how the observation model deals with only partially visible animals. Due to the highly deformable shape of rodents, it is difficult to explicitly model occlusions. As we have discussed before, incorporating motion cues in the observation model can partly compensate for the missing appearance information [16, 63].

2.2.1.3 *Maintaining Identities of Multiple Rodents*

Separating the animals in the image is one of the challenges that arises from tracking multiple animals. Another is maintaining the correct identities throughout the entire video. Maintaining identities is highly relevant when the sociability of individual animals is measured. For example, it may be of interest to determine the initiator and the receiver of an interaction to assess social hierarchies [1, 8, 142, 147].

To maintain identities, the tracking algorithm needs to associate the location of an individual in one video frame with its corresponding location in the next frame. This is a simple task as long as the animals are separated by a distance larger than their displacement between the two frames. Then each new location can be linked to the closest location from the previous frame. In contact and occlusion situations, this distance cue is unreliable. Instead, temporal cues such as the smoothness of motion may be used to propagate the identities according to a smooth continuation of the movement [123]. However, the assumption of smooth motion is sometimes violated as well. For example, in fast-paced playing or fighting interactions, the erratic movements of the rodents can cause the identities to be confused. Therefore, approaches that rely solely on distance and temporal continuation risk to propagate confusions through the rest of the video without any chance for recovery.

To prevent error propagation and to automatically correct identity confusions, visual appearance features that distinguish the individual animals are needed. These can be natural features such as fur color [55] or body size [92]. It may also be possible to detect small differences in fur patterns [101] or in body temperature using a high-resolution thermal camera [48]. If the animals lack any obvious, visual feature that would distinguish them or if the differences are too small, artificial features can be added. Shemesh *et al.* [129] applied fluorescent markers to the fur that light up in different colors under UV light, and Ohayon *et al.* [93] dyed the fur



(a) Top view (b) Side view

Figure 2.6: 3D point clouds of two rats recorded with a depth camera.

with different patterns of strokes and dots. Under daylight conditions colored tags can be used to track and identify the animals [104]. As natural features are not always available, tracking with artificial markers, including RFID tags, is still the only reliable option to maintain identities in social behavior experiments.

2.2.1.4 *Markerless Tracking in 3D Images*

Applying artificial markers requires handling the animals prior to experimentation which can induce stress and consequently alter the behavior. It is therefore generally preferred to achieve identification without markers. Recent advances in 3D imaging could present a new approach to tackle the occlusion challenge and therefore make identification more robust. With 3D imaging the animals can be represented as three-dimensional volumes instead of two-dimensional shapes as illustrated in Figure 2.6. Using the extra dimension, occlusions can be resolved with higher accuracy as the contours in the contact regions are better visible. 3D videos of rodents can be acquired from depth cameras [55, 90, 153] or from multiple video cameras with different viewpoints [120, 128].

An additional advantage of using 3D imaging to observe the rodents is that they allow the reconstruction of the 3D pose. The pose can be represented by the location of specific points on the body such as nose, tail base and ears [120] or larger body parts such as head, trunk and hip [87, 88]. From this 3D pose, elevated activities such as rearing or crawling over another animal can be derived more easily, which eliminates some of the limitations of the standard top view cameras. Because of the potential to create richer descriptions of behavior, the use of 3D imaging for rodent observations is gradually receiving more attention.

Yet, a number of challenges limit the applicability of 3D cameras in practice. One is the increased requirement for data storage. 3D images are multiple times larger in file size than 2D images due to the lack of appropriate compression algorithms. An experiment of a few hours of observations easily requires several terabytes of disk space. Furthermore, processing 3D images is more complex, has a higher demand on the computational resources and many established image processing techniques are restricted to 2D images. This makes the development of tracking algorithms more challenging and time-consuming. Finally, most depth cameras provide 3D images in the form of point clouds representing the observed surfaces. They do not provide appearance information such as illumination, contrast and texture. Given that such information may be needed to uniquely identify the animals after potential confusions, an additional, 2D camera may have to be installed, synchronized and registered (i. e. calibrated) with the 3D camera [55]. The 2D camera has the additional benefit that the researcher can validate observations in the more familiar 2D videos but obviously further increases both storage and computational requirements.

2.2.1.5 *Evaluating Tracking Quality*

Despite playing such a crucial role in behavior analysis, the quality of rodent tracking is rarely evaluated systematically because the *ground truth* to compare with is time-consuming to generate manually. As there is no benchmark dataset publicly available, literature comprises only few quantitative evaluations [55, 119] or comparisons among methods [143]. The lack of quantitative analyses raises the questions what influence the tracking quality may have on automated annotation methods. We address this question in Chapter 5 and conduct a systematic analysis of the effects of tracking quality on the annotation accuracy.

2.2.1.6 *Deriving Features for Behavior Classification*

Once the locations of the rodents are tracked, features can be derived from those tracks. These features represent relevant aspects of the behaviors and enable the subsequent inference algorithm to classify and annotate the behaviors. Therefore, the choice of features is often geared toward the relevant behaviors. A basic feature set for social interactions comprises the velocity of each rodent as well as the distances and orientations between them [18, 48, 61].

Additional features can be included if a detailed pose representation can be extracted from the trajectories. For non-social behaviors it has been shown that extracting a pose representation based on multiple body parts

improves the accuracy of automated annotation [33]. It is likely to improve social behavior recognition as well. Extracting the pose requires that the tracking algorithm not only provides the center point locations of the animals [4, 18, 147], but also locations from other body points such as the nose or tail-base [31, 48]. Detailed distance and motion information allow for annotation of interactions between specific body parts such as *nose-nose inspection*. We perform an experiment to demonstrate the effect of pose information on annotating social behavior in Chapter 5.

In case it is not possible to reconstruct the pose representation from tracking data, one may instead use features derived directly from the images. For example, tracking a large number of key points in the video, such as corners and edges, yields a global description of the motion in the video. This approach has been applied to a single mouse [38, 59]. Similarly, the optical flow can be computed to describe the motion of body parts [145]. Despite the successful application on individual rodents, comparable image features only led to a minor improvement for annotating social behavior [18]. Presumably, detailed body motion is less informative for the considered social interactions which are mostly related to the trajectories of the animals relative to each other. Furthermore, in the social setting it is more difficult to associate the key points with the correct animal, in particular in contact situations where it would be most useful.

As mentioned before, developing a feature extraction method typically requires the design of features for specific behaviors and involves a certain amount of manual tuning to reach satisfying annotation results. Methods that automatically learn suitable feature representations and classification models jointly from a large set of annotated training data, are becoming increasingly popular for computer vision applications in general and thus also for rodent behavior recognition. We briefly discuss such *end-to-end* recognition frameworks in Section 2.2.3.

2.2.2 *Annotation by Classification*

We now turn to the annotation component. The task of the annotation component is to label the occurrences of behaviors in a recording. It infers the labels from the features provided by the observation component. The final labeling output should facilitate further behavior analysis, for example, to examine occurrence frequencies and mean or total durations of specific behavior categories. Therefore, the annotation component must label the occurrence of a behavior and its beginning and end time. The automatically generated annotations are thus directly comparable to annotations obtained from a human observer.

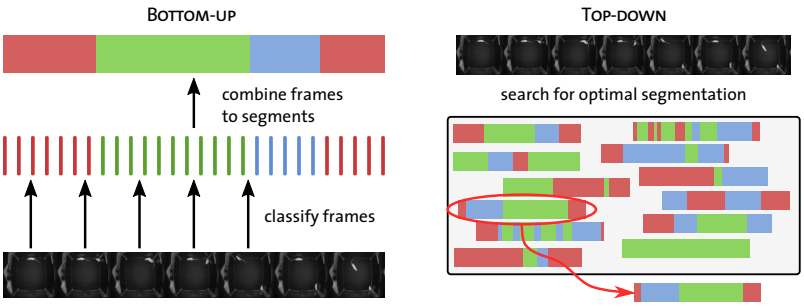


Figure 2.7: Bottom-up (frame-based) and top-down (event-based) classification schemes.

The behavior categories, as defined by the investigator of the behavior study, are typically mutually exclusive and exhaustive [2, 100, 137, 139]. That is, they are defined such that at any time in the video, there occurs exactly one behavior that can be labeled. This allows us to formulate the annotation task as a classification problem. Every video frame, represented by a feature vector $\boldsymbol{x}$, is classified into one discrete behavior category $\hat{y}$. The classification can be achieved by expressing the behavior categories in terms of mathematical models that create a mapping from features to categories: $f_\theta : \boldsymbol{x} \mapsto \hat{y}$, where θ are parameters of the model f. For instance, a simple threshold function could map the distance between two animals to two discrete states, either *being in proximity* or *at a distance* [27, 31]. Much of the early work on rodent behavior annotation used such simple rule-based models [134, 135]. They are attractive because the limited number of parameters makes them easy to work with. The parameters are determined manually in such a way that the experimenter is satisfied with the model output for some given input [66, 147]. Naturally, this approach requires a considerable amount of trial and error, and the manual tuning complexity increases with the number of parameters.

With machine learning, manual tuning can be replaced by a learning algorithm [13]. Learning algorithms determine the model parameters automatically and in a fraction of the time needed with manual tuning. This speed-up allows for training more complex classification models that can distinguish between multiple, fine-grained behaviors. Popular models used for rodent behavior classification include neural networks [53, 115], decision trees [48, 55], and support vector machines [42, 59]. We will discuss the training of such classifiers with rodent data in Section 2.2.2.1.

Approaching the annotation problem at the frame-level allows us to use many off-the-shelf classifiers because every frame is considered an inde-

pendent sample. Clearly, consecutive video frames are not actually independent and the context given by surrounding frames can potentially inform and improve the classification. For example, to classify *approach* behavior it may be more informative to compare the distance between the animals at the beginning of the approach and at the end, rather than to rely on measuring the change of distance in adjacent frames. The difficulty to incorporate such context information is because frame-level classification lacks the explicit notion of a behavior segment (state event), that is, a consecutive sequence of frames showing one behavior. Although information from surrounding frames can be incorporated, the temporal distance at which this happens is arbitrary. Burgos-Artizzu *et al.* [18], for instance, incorporate confidence scores of the classifier from a fixed window of up to 615 frames irrespective of the behavior category. A fixed window cannot facilitate behaviors with different durations.

In order to incorporate contextual information over an extended and variable period of time, the annotation task has to be approached from a different perspective. Instead of proceeding from frames to segments in a bottom-up approach as illustrated in Figure 2.7, we can choose a top-down approach and segment the video directly into labeled state events. The task is then to find the optimal temporal segmentation of the entire video. This allows the classifier to incorporate information at variable temporal distances, for instance, from behavior transitions such as the beginning and the end of an approach [42]. The family of classification models for such a task are *structured prediction algorithms*, named after their structured output (in our case the labels and the start and end times of each state event in a video). An example is the structured support vector machine (SVM^{struct}) [140].

The main disadvantage of the top-down classification approach is its high computational complexity. The number of segmentation possibilities increases exponentially with both the number of categories and the length of the video, which renders exact inference of the optimal segmentation often intractable. Although efficient algorithms such as dynamic programming [9] (as used in [42]) or belief propagation [97] may be used in some cases, the computational requirements remain higher than for bottom-up classification. Furthermore, a structured classification model has often more parameters to learn than a frame-level classifier and generally requires more training examples to ensure sufficient accuracy [73].

Although classifiers based on Hidden Markov Models (HMMs) [106] are conceptually top-down approaches, the Markov assumption limits the temporal context that can be incorporated to only two consecutive temporal units. This limited context can only be informative if the HMM is applied

to an intermediate representation on a higher temporal scale than individual video frames. That is, a temporal unit should be longer than one frame. Two consecutive video frames do not convey much contextual information when the video frame rate is much higher than the rate of transitions between behaviors [4, 147]. A possible approach is to learn *action units* or *movemes* [3] as an intermediate behavioral representation and to combine the HMM with a context free grammar to model the temporal structure of those units [73]. Learning such an intermediate representation together with the corresponding temporal model requires a large amount of training data.

In the rodent behavior recognition literature, the majority of work takes the bottom-up classification approach. Its simplicity and the wide availability of implemented, off-the-shelf classifiers makes it attractive. Moreover, despite their limitations, frame-based models are still powerful models whose capacities may not even be fully exploited yet. In light of other challenges such as tracking multiple rodents, deriving meaningful features and learning from relatively small, unbalanced datasets, incorporating structured temporal context does not appear to be of the highest priority yet.

2.2.2.1 *Learning Classification Models*

We have previously introduced the classification model as a mapping from a feature vector to a behavior label whose function is controlled by a set of parameters θ. In this section we discuss approaches to automatically determine the optimal parameters for a given classification problem using machine learning. For this task, we will interchangeably use the terms *learning a model* and *training a classifier*.

The learning algorithms used for training rodent behavior classifiers are predominantly supervised [4, 18, 42, 48, 53, 55, 59, 115]. That is, the optimal parameters $\hat{\theta}$ are inferred from annotated training examples. The set of examples includes the behaviors that we expect the classifier to recognize when it is applied in practice. Ideally, every behavior category is represented by a large number of examples as the classifier will generally become more accurate with every additional example. The training set can be obtained either from videos of previous studies that have already been annotated and analyzed with a traditional method, or from novel videos that are recorded and manually annotated specifically for this purpose. Using already annotated data saves both time and animals needed for the experiments. However, manual corrections are sometimes needed to increase the temporal accuracy of the annotations before they can be used for training. In both cases, we have to appreciate the fact that we have limited control over how the animals behave and how often they display certain actions.

Since some behaviors occur rarely, it can be a tedious task to obtain a sufficient number of examples from every category.

The learning algorithm uses the training set to determine the optimal parameters $\hat{\theta}$. The optimal parameters are the ones that reproduce the labels of the training examples with the fewest errors. The algorithm can achieve a low error simply by remembering every example and the corresponding label in a large database. Obviously, this algorithm has remembered but it has not learned. In other words, it has not generalized from the empirical examples to the natural variation of the behavior categories and it would fail to produce a reasonable classification for an unseen example. To evaluate whether a classifier has learned to generalize, a common technique is to remove a subset of the examples and train the classifier on the remaining part. We can then count the mistakes on the removed examples to compute the expected error for unseen examples. A more robust estimation of the true error can be achieved by repeating this experiment with different parts removed and then averaging the error over repetitions. Such a procedure is called *cross-validation* in the machine learning domain and it is used extensively in rodent behavior classification [48, 59, 61, 145].

Cross-validation can also be used to analyze particular generalization capabilities concerning for instance different video acquisition environments, animal strains or gender [59, 145]. The examples to remove are then chosen such that they contain a particular characteristic that is otherwise absent in the training data. The evaluation on the removed examples describes the ability of the classifier to deal with unseen scenarios such as a different experiment setup. Because the majority of work on rodent behavior classification is based on examples from one environment, validation across environments is typically not performed. The lack of such a validation can be critical in practice. In Chapter 6 we demonstrate with a cross-dataset classification experiment how variation in behavior caused by the animal's age affects the classification error.

Both learning and evaluating rodent behavior classifiers are based on examples that have been labeled by a human annotator. As we have found in previous sections (Section 2.1), human annotators are subject to disagreement which affects both learning and evaluation. During learning we may encounter examples that would have been labeled differently by another annotator. This can cause the learning algorithm to find parameters that are optimal for the examples labeled by one annotator but not optimal when labeled by another. This is an undesired property as the training examples should be free of such disagreement. As a consequence, the expected error, which is computed from the removed examples in cross-validation, is subject to the level of disagreement in those examples. We

find that two factors contribute to the expected error: the true inability of the classifier to recognize a behavior and the uncertainty that is due to disagreement among annotators. Therefore, the expected error depends on the level of disagreement in the test examples. This effectively puts an upper bound on the achievable accuracy of the classifier.

Suppose we knew which examples exhibit disagreement among annotators. We could account for those examples during training and evaluation, for example by simply omitting them or by weighing them differently in the calculation of the error. The simplest way of determining disagreement is to let multiple annotators label the same training examples. Obviously, the manual effort multiplies with the number of annotators which quickly renders this approach unattractive. Alternatively, a single annotator can be asked to not only label the behaviors but also assign a weight to each example according to its importance. Kabra *et al.* [61] evaluate their annotation framework based on labeled video frames that have additionally been assigned three different scores: important, unimportant and unknown. They then evaluate the classification error only on frames marked as important and thus obtain a better estimate of the true expected error. The same scheme could be extended to the classifier training so as to learn only from the important examples. On the one hand, it can be argued that forcing the user to reason about the importance of an example decreases the chance of annotation mistakes. On the other hand, disagreement caused by differently interpreted behavior categories cannot be exposed in this way.

2.2.2.2 *Learning with Human in the Loop*

Until now we have assumed that training a behavior classifier is a task that has to be completed once, whereafter it can be used to annotate videos. The scenario in which this is the case is restricted to situations in which no substantial changes are made to either environment or animals. As we have mentioned in the previous section, modifications to the acquisition setup such as lighting or viewpoint, or testing a novel animal population that displays unseen behavior variations, can affect the classification accuracy. Moreover, classifiers are trained to detect a fixed set of behavior categories and alterations to the definitions or an extension of this set is typically not possible without retraining. In such scenarios, the trained classifiers cannot be used and the videos need to be labeled manually.

Although we may not be able to avoid manual labeling completely, we can aim to reduce the manual work and support the labeling by automation. In fact, any annotation made by a human observer may be used for training a new classifier. Therefore, it is possible to train the classifier at the same time as the human annotates the videos: the human is included

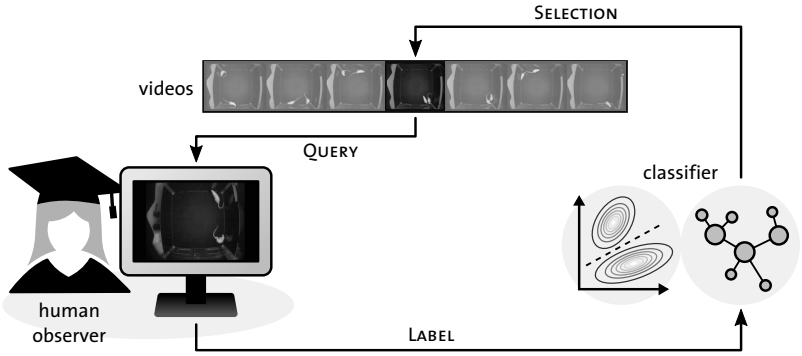


Figure 2.8: Active learning loop for automated annotation of rodent behavior.

in the loop. The more examples the human annotates, the better we can train the classifier. As soon as the classifier has learned sufficiently, it may take over the annotation task from the human. The crucial decision in this scenario is to decide when to let the classifier take over. This involves a trade-off between the time-investment of continuing the manual labeling on the one hand and the potential increase in accuracy of the classifier on the other. Stopping early may result in misclassifications while stopping late wastes valuable time because the classifier has already converged to its maximum accuracy.

The usual way of manually annotating a video is to label the behaviors sequentially as they occur. This creates a sequence of labeled examples that is inefficient for training an accurate classifier. A large number of videos need to be labeled before sufficient examples of behaviors that occur infrequently are included. Instead, it is desirable to make a more balanced and varied selection of examples across videos that includes all behaviors.

The *human in the loop* concept is implemented in the Janelia Automatic Animal Behavior Annotator (JAABA) framework [61]. The user of the framework labels a number of video segments while a classifier is repeatedly trained. The classifier then propagates the annotations to the unlabeled parts of the videos which in turn are judged and possibly corrected by the user. This loop continues until the user is satisfied with the generated annotations. Because the user chooses which examples to label, the quality of the final annotation depends on this selection. The criterion based on which humans choose examples may not necessarily coincide with what is informative for the classifier. Moreover, to find suitable segments the user has to browse through the video which takes additional time and increases the risk of missing relevant examples.

In our interactive annotation framework that we present in Chapter 7, we select examples automatically and actively query the user to provide a label. This *active learning* loop is illustrated in Figure 2.8. Because the selection is derived from the current classification model, the examples are informative from the perspective of the classifier rather than the user. The interaction of human and machine makes this learning paradigm powerful [84, 125]. We exploit the perception and generalization skills of the human while leaving the complex, objective computations to the machine.

2.2.3 *End-To-End Recognition*

Automated annotation is based on information (features) that is extracted from the tracked animal locations or directly from the video frames. The features are typically designed to capture relevant aspects of the behavior to be annotated. For instance if we want to classify *approach* behavior, then a meaningful feature could be the change of distance between the animals. Designing features for every behavior category can be a time-consuming task as it is not always clear what information should be extracted and how to encode it in the features. Finding appropriate features often requires a certain amount of trial and error.

A way to circumvent manual feature design is to learn the features simultaneously with the classification model from labeled training data. Such an end-to-end solution has as input signal the video image and as output the behavior label. The ability to map directly from image to label comes from a complex, non-linear model with a large number of parameters. The complexity of the model makes it powerful but difficult to learn [10]. The learning algorithm requires a large number of training examples to find an accurate model and the demand for computational resources is high. In computer vision, such models have been successfully applied to recognize objects in images [50, 71, 132] using millions of training images [36].

End-to-end methods for visual inference tasks typically involve *convolutional neural networks* (CNNs). The architecture of such networks is particularly suited for learning visual features. It allows learning hierarchical feature layers, ranging from low-level information, such as edges and corners, to mid-level representations of objects parts and finally complete scenes. Because the lower levels represent general visual features, which are largely independent of the specific inference task, they can often be reused and transferred into networks for related tasks [50]. The ability to transfer previously obtained knowledge can drastically reduce the time and the number of examples needed to learn a model.

CNNs are also applied to recognize human actions in videos [44, 131]. Processing videos requires learning not only appearance features but also temporal features that represent motion [138]. This further increases the model complexity and the number of required training examples.

In the domain of rodent behavior recognition, CNNs have recently been considered to classify individual behavior of rats and mice [69, 108]. The proposed models are not yet powerful enough to replace annotation methods based on designed features. Besides the already mentioned task of modeling motion in such networks, another challenge is the transfer of networks from the human to the rodent domain. Networks trained for object detection or human action recognition typically learned from man-made and natural scenes which are intrinsically different from the laboratory environments in which the rodent itself is the main subject of interest. If instead a suitable network is to be trained from scratch, a large number of labeled training videos need to be obtained first.

2.3 Conclusion

The desire to automate rodent behavior analysis has a long history and solutions to measure behavior of individual rodents have been around for several years. Only recently, with advances in machine vision, research on recognizing specific interactions between rodents has gained momentum. There are several challenges including location tracking and pose estimation of multiple rodents, representing pose and motion with suitable features, reasoning about interactions and their temporal extent, and learning as well as validating classification models. Previous work shows progress in tackling one or more of these challenges, but is often applied to only one set of videos and behaviors. The comparison and benchmark of the proposed methods is hindered by the scarcity of publicly available social behavior datasets. As a consequence, there is a demand for new datasets and more importantly for a unified recognition framework that is proven to be applicable to a variety of experiment settings and behaviors. The research on such a framework can build on the insights from previous work and combines it to identify and tackle the prevailing challenges of rodent interaction recognition.

The work in this thesis is primarily related to learning rodent behavior classifiers and aims at annotating behavior videos interactively with reduced manual effort. Much of this work relies on understanding the properties of rodent social behavior videos. Before we turn to the analysis of these videos, we first introduce our publicly available rat social behavior dataset in the following chapter.

3

Rodent Social Behavior Datasets

The research on automatic annotation of rodent behavior requires experimenting on realistic behavior data, in our case video recordings with the corresponding true annotations. However, publicly available datasets of rodent social behavior are scarce. In fact, only two datasets with mice are currently available, the Caltech Resident-Intruder Mouse dataset (CRIM13) and the Mice Behaviour Analysis dataset (MBADA), and no rat datasets. Therefore, we have compiled two rat social behavior datasets for our research, namely: the Young Rats (YR) dataset that was composed from existing recordings and the rat social interaction (RatSI) dataset that we newly acquired within the PhenoRat project. The RatSI videos were recorded for a longitudinal study on the social behavior in a rat model for Spinocerebellar ataxia type 17 (SCA17), a neurological disorder associated with motor and cognitive impairment. RatSI is made publicly available to the research community (`http://www.noldus.com/innovationworks/datasets/ratsi`).

In this chapter we introduce YR and RatSI and also briefly discuss the two mouse datasets. Here we only describe the content of the datasets, how they were acquired and what animals were used. We will discuss in more detail the characteristics of the observed behavior and the differences between the species in the following chapter.

3.1 Our Young Rats Dataset

The YR dataset is a collection of short video clips each containing one social interaction between two rats. The locations of the rats are manually corrected in every video frame which makes YR a controlled and polished dataset. These properties are ideal for studying tracking and feature-related aspects in absence of other factors such as label noise. The fact that only selected clips are annotated, however, makes it unsuitable for investigating temporal aspects.

We compiled YR from five top view videos of an open-field social interaction test provided by Suzanne Peters, Utrecht University. The recordings were originally obtained and annotated for a study on play behavior of young rats [102]. The provided annotations could not be used directly for our research as they are not temporally accurate. Specifically, the start and end points of the interactions as annotated do not consistently coincide

Table 3.1: Description of behavior classes used in YR and RatSI.

Allogrooming	Grooming another rat's fur
Approaching	Moving toward another rat in a straight line
Following	Chasing another, moving rat within a tail length distance
Moving away	Moving away from another rat in a straight line
Nape attacking	Snout or oral contact directed at neck region, possibly with biting/pulling fur in that region
Pinning	Actively restrain another rat on its back
Social nose contact	Non-incidental nose-body contact (e.g. inspection)
Solitary	Any activity not directed at another rat
Other (RatSI only)	Any interaction not covered by another category
Uncertain (RatSI only)	Ambiguous or occluded interactions

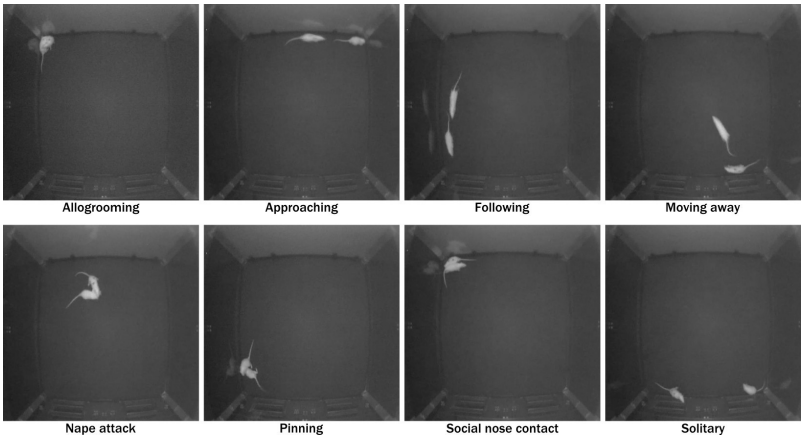


Figure 3.1: Example frames of the interactions in YR dataset.

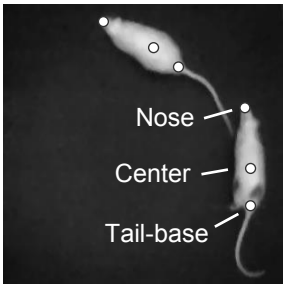


Figure 3.2: Three points on the body of every rat are tracked in YR and RatSI.

with the video but may be shifted by a varying delay of up to one second. Such delays occur when the researcher annotates while watching the video without subsequent refinement. Because the annotator needs to see at least a few frames of an interaction before he or she can decide on the category, the true beginning of the interaction is missed. While annotation delays only have a minor influence on a subsequent analysis in terms of counts and frequencies, we require frame-accurate annotations to be able to develop and evaluate our automated annotation method.

To generate frame-accurate annotations from the original data, we randomly select ten events of each interaction category per video (400 segments in total) and correct their start and end times. In total this yields 12.6 min of accurately annotated video. Note that some interactions succeed each other immediately, forming one longer clip of two or more interactions.

3.1.1 *Behavior Classes*

We include seven interaction categories and additionally one category that covers all solitary actions. The categories are briefly described in Table 3.1 (see the detailed definitions in Table A.1 in Appendix A) and example frames are shown in Figure 3.1. The annotated interactions are related either to the trajectories of the animals such as *approaching* and *following*, or to the contact between the rats such as *allogrooming* and *nape attacking*.

3.1.2 *Tracking*

The locations of the rats, including three body points (nose, center of body mass, and tail-base), were tracked with Noldus EthoVision XT 11. See Figure 3.2 for an illustration of the tracked points. Body point locations and identity errors were corrected manually within the selected clips including a 0.6 s (15 frames) margin before and after each interaction. For the analysis of tracking quality in Chapter 5, this correction is achieved in two steps. First, the identities are corrected, keeping potentially inaccurate tracking locations. In the second step, the body point locations are also corrected.

3.1.3 *Experiment Protocol & Animals*

The recordings of YR are part of a larger social interaction study adopting the following protocol. On two separate days before the recordings, the rats were individually introduced to the cage arena for thirty minutes. Forty-eight hours before the test, the rats were isolated to stimulate a de-

sire for social interaction. Each rat was then put in the recording cage together with a familiar sibling. Recordings last for thirty minutes and are made from a top view perspective in a 90x90 cm Noldus PhenoTyper® 9000 cage with standard top unit (image resolution 704 × 576 px, 25 fps) without bedding or accessories. The videos are monochrome as they are recorded under red-light conditions during the animals' active, dark phase.

One group of ten naive Sprague Dawley males, 5 weeks old, were used. The animals were housed socially under reversed day-light cycle conditions and water and food were available ad libitum. The experiments were performed in adherence to the legal requirements of Dutch legislation on laboratory animals (WOD/Dutch "Experiments on Animals Act") and were reviewed and approved by an Animal Ethics Committee ("Lely-DEC"). For more details, refer to the original study [102].

3.2 Our RatSI Dataset

Our Rat Social Interaction (RatSI) dataset consists of nine fully-annotated videos of 15 min each. The recordings and annotations were made by E.I. Kyriakou at the Radboud University Nijmegen Medical Center, who conducted the experiments to study a rat model for SCA17 [65, 76]. The contents of the recordings are comparable to YR, except that the animals are older (9 months instead of 5 weeks) and therefore larger. As RatSI contains continuous videos, it represents a realistic use case for automated interaction recognition.

During the acquisition of RatSI, the annotator was instructed to score the start and end points of interactions accurately without the delays found in YR. While such accurate annotations are more time-consuming to obtain, they do not require correction afterward. The interactions are mutually exclusive and therefore cannot overlap in time. Hence, the start of a new interaction coincides with the end of the previous. In addition to time and category, the annotators also scored which animal plays the active role in the interaction. The active animal is defined as the rat that is for example *following* or *moving away*. When both animals perform the same interaction together, for instance when both *approach* each other, it is scored as such. The rats are easily identified by the black marking on the back of one rat. In Chapter 4 we discuss the quality of the obtained annotations.

3.2.1 *Behavior Classes*

RatSI includes the same behavior categories as YR, described in Table 3.1. Because each frame of the videos in RatSI is annotated, there are two ad-

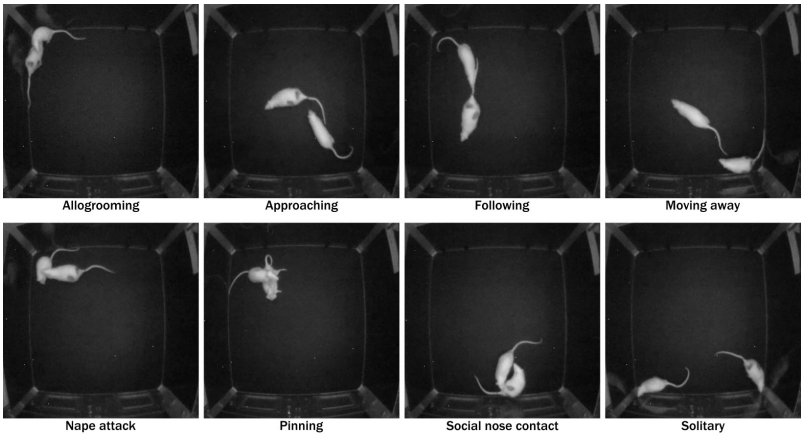


Figure 3.3: Example frames of each behavior in RatSI dataset.

ditional categories, namely: *other* and *uncertain*. *Other* is annotated for interactions that are expected to occur but are not relevant for the study. In RatSI these include amongst others boxing or wrestling. *Uncertain* is reserved for cases in which the interaction is not clearly visible due to occlusion, speed or video resolution. Example frames from RatSI videos are shown in Figure 3.3. Note that the rats are larger compared to YR (Figure 3.1); the size of the cage is identical.

3.2.2 *Tracking*

The animal locations and body point positions (nose, center of body mass, and tail-base) have been tracked throughout the videos using a tracking algorithm that is based on Noldus EthoVision XT 12, extended by a customized rat identification algorithm. The identification algorithm uses appearance differences (here reinforced by black markers) to distinguish and maintain the identities up to a few errors per video which we correct manually afterward. Note that the identification algorithm is under development to facilitate markerless identification and is therefore not included in the official EthoVision XT 12 version. The resulting tracking data is more accurate than the uncorrected data in YR. Thus we do not manually correct the body point locations in RatSI. In occlusion situations, locations can be inaccurate and nose and tail-base points can be confused. Such confusions are automatically corrected based on animal's movement and typically last for only a few frames.

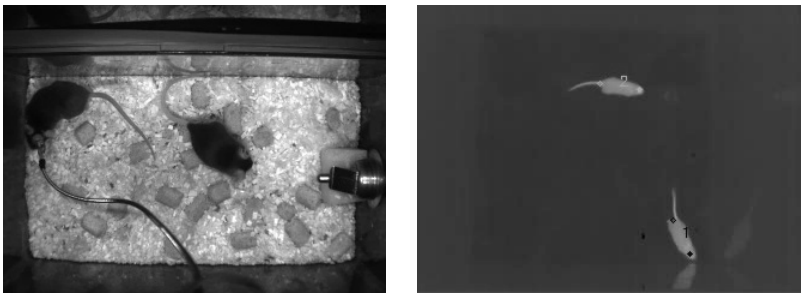
3.2.3 *Experiment Protocol & Animals*

The recorded experiments are part of a social interaction study adopting a protocol similar to the one used for YR. Three days before the recordings, the rats were individually introduced to the cage arena for twenty minutes. Twenty-four hours before the test, the rats were isolated to stimulate a desire for social interaction. Each rat was then put in the recording cage (a 90x90 cm Noldus PhenoTyper® 9000) together with another, unfamiliar rat. The recordings start with the introduction of the second animal and are made from a top view perspective (image resolution 704×576 px, 25 fps) without bedding or accessories. The videos are monochrome as they are recorded under red-light conditions during the animals' active, dark phase.

For testing, naive male rats, 9 months old, of two genotypes were used: SCA17 ($n = 8$) and wild-type-like (Sprague Dawley, $n = 10$). The animals were housed in pairs under reversed day-light cycle conditions and water and food were available ad libitum. Testing was performed during the animals' active (dark) phase. All experiments were performed after approval of the Ethical Committee for Animal Experiments of the Radboud University Nijmegen Medical Center for compliance to ethical standards and use of laboratory animals according to EU-guidelines.

3.3 Public Mouse Datasets

Although rats and mice are similar in the evolutionary sense, they have different behavioral traits. In particular, rats display different social behavior patterns such as complex play-fighting which is not seen in mice [98, 100]. Rat and mouse datasets are therefore not simply interchangeable in the search for automated measuring methods. Nonetheless, mouse behavior datasets can be a valuable source of information for studying the inherent characteristics of trajectory-related interactions. Together, rat and mouse datasets pose an interesting task for domain adaptation or transfer learning [95], that is, the transfer of classification models from one species to another. In this section, we give an overview of the two publicly available mouse social behavior datasets: CRIM13 and MBADA. We will use CRIM13 in later chapters. Note that a third mouse behavior dataset, the SCORHE dataset (`https://scorhe.nih.gov`), contains several short recordings of two and three mice but no annotations of their interactions.



(a) CRIM13 (b) MBADA

Figure 3.4: Examples frames from mouse datasets.

3.3.1 *CRIM13*

The Caltech Resident-Intruder Mouse (CRIM13) dataset [18] comprises approximately 88 hours of annotated video recordings of a resident-intruder experiment. This experiment is similar to the social interaction test of our rat datasets in that two animals are interacting freely in an observation cage. The main difference is that in CRIM13 the observation cage is the home-cage of one of the mice, the male resident. The second mouse is placed in the cage as an intruder whereupon the resident will typically inspect the intruder's gender and engage in attack (if male) or courtship (if female).

The original study, from which the material has been collected, is concerned with the neurophysiological mechanisms in the mouse brain. The researchers used optogenetic stimulation of neurons to manipulate the level of aggression. This stimulation requires a physical connection with the mouse brain which is always visible in the videos as seen in the lower left corner in Figure 3.4a. The cable partly occludes the view on the mice from the top view perspective. Although CRIM13 additionally contains videos recorded from the side, the animal location tracking is based on the top view alone.

3.3.1.1 *Behavior Classes*

The behavior of the resident mouse is annotated in the same way as in RatSI, that is, with the beginning and end points of mutually exclusive, exhaustive categories. Only the behavior of the resident mouse is annotated. The considered behaviors, listed in Table 3.2, include seven interactions, five solitary actions, and two background categories for unspecified, *other* behavior and for *human* intervention (placing and retrieving the intruder).

Table 3.2: Behaviors included in the CRIM13 and MBADA mouse datasets.

CRIM13		MBADA	
Solitary	Social	Solitary	Social
Drink	Approach	Stand alone	Above
Eat	Attack	Walk alone	Following
Clean	Copulation		Nose2Body
Human	Chase		Nose2Genitals
Up (Rearing)	Circle		Nose2Nose
Other	Sniff		Stand together
	Walk away		

There is some similarity between the categories of CRIM13 and RatSI, including *approach*, *chase* (*follow*), *sniff* (*social nose contact*) and *walk away* (*moving away*). The *other* category in CRIM13 comprises otherwise unspecified behavior, the majority of which is when the mouse is standing and walking through the cage. These would be labeled *solitary* in RatSI.

3.3.1.2 *Tracking*

The mice locations are tracked in the top view videos by an undisclosed algorithm giving access to the center point of each mouse in x, y coordinates. The accuracy is not evaluated empirically. Qualitatively, the accuracy of the locations seems reliable, although noise clearly increases in occlusion situations, e.g., during *copulation* and *attack*. A few identity swaps occur but persist only briefly. CRIM13 does not provide the locations of body parts.

3.3.2 *MBADA*

The second mouse dataset is the Mice Behaviour Analysis (MBADA) dataset [48]. It comprises 5.2 hours of video recordings of two or three mice in a square observation cage. MBADA is recorded with a thermal infrared camera allowing to measure the mice' body temperatures. If the temperatures are sufficiently different, it can help to identify the subjects without external markers. An example frame is shown in Figure 3.4b.

3.3.2.1 *Behavior Classes*

The included behaviors, listed in Table 3.2, are primarily defined by the distance between specific body points. Merely *following* is determined by

relative motion. The distribution of the frames among the behaviors in Figure 3.5 shows that the recordings contain mostly unlabeled and solitary behavior. Only 4% of the dataset (12.6 min) are social interactions.

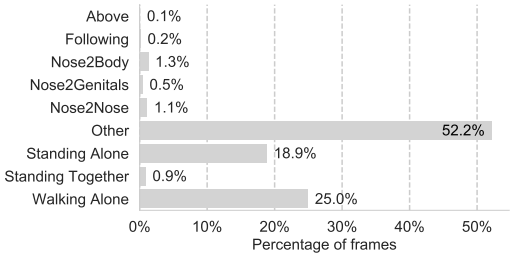


Figure 3.5: Behavior distribution in MBADA (scorer 1), 5.7×10^6 frames (5.23 hrs).

3.3.2.2 *Tracking*

MBADA provides the tracking of the nose and tail-base points as well as the center of body mass. The tracking algorithm first performs background subtraction to locate the animals. If not all animals are well separated, the body points are assigned in an iterative *Expectation Maximization* process that places the nose and tail-base points on the edges of the major axis of the segmented, elliptical shapes. Under the assumption that body points do not move far between subsequent frames, the identities of the mice are assigned. After occlusions, which can lead to incorrect identification, a heat signature extracted from the thermal image is used to reestablish the identities. The accuracy of the tracking is not empirically evaluated. Qualitatively, a few identity swaps occur, often after incidental contact, but body points seem to be assigned accurately and correctly in most of the frames. Accuracy degrades in contact and occlusion situations similar to all other considered datasets.

3.4 Conclusion

We introduced two rat interaction datasets, YR and RatSI, which enable us to investigate the automated recognition of rat social interactions. We have made our larger dataset, RatSI, publicly available with the goal to support further investigations by other researchers and to enable the benchmark and comparison of automated measuring methods. We believe that our dataset can help in particular those researchers who lack the facilities to

acquire annotated behavior recordings on their own and those who wish to validate their methods.

Throughout this thesis, we will repeatedly refer to CRIM13 as a point for comparison between rat and mouse social behavior. Some interactions in CRIM13 resemble those in YR and RatSI, and thus make such comparisons possible. In contrast, the fact that only the locations of the mice are tracked in CRIM13, but no additional body points, poses a challenge for automatically recognizing interactions such as *attack* and *copulation*. As far as MBADA is concerned, the low amount of motion-related interactions leads us to the conclusion that MBADA is not sufficiently diverse for our investigations.

4

Analysis of Rodent Social Behavior in RatSI and CRIM13

In this chapter we investigate rodent social behavior from a computational point of view. Specifically we examine properties of social behavior datasets that are related to learning, applying and evaluating behavior classification models. The knowledge that we gain from these investigations will allow us to make informed choices in later chapters.

Regarding learning and evaluating, we first study the composition of the RatSI and CRIM13 datasets by measuring how often specific interactions occur based on human annotation. As we have discussed in Section 2.2.2.1 the occurrence frequency is valuable information as the absolute dataset size is misleading if some interactions occur rarely. We further examine whether occurrence frequencies change over the course of an experiment. Then, we look at how the interactions are annotated by human observers and examine the variance in their labeling with an inter-annotator reliability study. By quantifying the disagreement in the annotations, we can better judge the classification performance (Section 2.2.2.1). Finally, we explore a possible feature representation for the animal trajectories that facilitates the automated classification of the interactions.

The main contribution of this chapter is the systematic analysis of the RatSI and CRIM13 datasets. Additionally, we introduce a set of features that captures various aspects of rodent interactions and that forms the basis for our classification models in the following chapters. We use this analysis to identify the characteristics of the considered data which helps us to address them appropriately in the classification method. Naturally, the specific characteristics depend on the type of experiment and behaviors, and the results are limited to the RatSI and CRIM13 datasets. Other researchers may use this chapter as inspiration to analyze their own datasets.

We begin with the analysis of occurrence frequencies in Section 4.1, followed by a study of the inter-annotator agreement in Section 4.2. In Section 4.3 we introduce the feature set and discuss its properties. We draw conclusions from the results in Section 4.4.

4.1 Composition of Rodent Social Behavior

We consider the behavioral composition of a dataset as a set of statistical properties. Specifically, we include the number of occurrences of each behavior, how the frequencies of these occurrences change in the course of a video, and whether occurrences of one behavior may be conditioned on an immediately preceding behavior.

4.1.1 *Occurrence Frequencies of Interactions*

We take a probabilistic perspective and model the occurrence of a behavior as a random variable. Let E_a be a random variable that represents the occurrence of an event: an interaction of category a. Then, $P(E_a)$ is the prior probability that a given interaction observed in a video belongs to the category a. $P(E_a)$ provides us with the base rate with which a behavior will occur in a video on average. We determine the prior probability for every behavior in the dataset simply by counting how often it has been annotated. We want to note that this view is strongly simplified as it assumes that every occurrence is independent of any external factors such as the time of day, the experiment duration or preceding interactions. Furthermore, we treat an occurrence of an interaction as a single event irrespective of its duration. Later, we will take the duration into account and examine the prior probabilities with respect to single video frames.

In Figure 4.1 the prior probabilities of the occurrence of an interaction in the datasets are shown in terms of percentages. In RatSI the most occurring interactions are *social nose contact*, *approach*, *following* and *moving away* (each covering approximately 10-20% of the behaviors). Interactions with close contact (*allogrooming* and *nape attacking*) occur less frequently (around 4%) and *pinning* rarely (less than 1%).

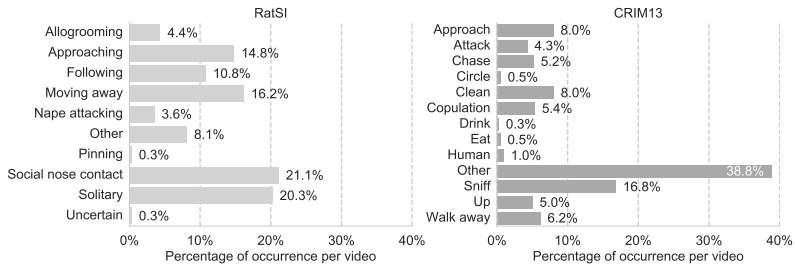


Figure 4.1: Prior occurrence probabilities of behavior events expressed as percentages.

In CRIM13 the interactions between animals are slightly more balanced, each typically covering approximately 4-8% of the behaviors. The exceptions are *sniff* with 16.8% at the upper end, and *circle* with only 0.5% at the lower end. All solitary behaviors together are approximately 52.6% of the dataset with the majority (38.8%) being labeled as *other*. Note that part of the solitary actions happen in the beginning of each video when the intruder mouse is not yet present in the cage.

Apart from a few exceptions in both datasets, the occurrences of the interactions are relatively balanced. Nonetheless, the solitary actions occur at least as frequently as the most occurring interaction and thus account for a substantial portion of the datasets. The prior probability for solitary behavior is therefore structurally higher than for most interactions.

The ratio between social and non-social behavior in the dataset is important for the automated recognition. Recognizing interactions in a large pool of non-social, background activities is more difficult than distinguishing between only interactions. Given the higher prior probability, the recognition may become biased towards predicting solitary actions as it will, on average, cause fewer errors. The corresponding classification model should address the skewed priors, for example by artificially balancing training and test data.

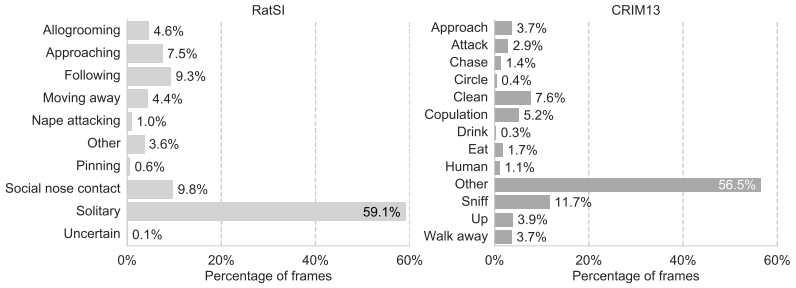


Figure 4.2: Distribution of behaviors among the frames of RatSI and CRIM13.

Let us now take the duration of the behaviors into account. Let F_a be a random variable that represents whether in a given video frame a particular interaction a is occurring. Since behaviors span multiple video frames, the corresponding prior probability $P(F_a)$ then also reflects the typical duration of the behavior. To determine the value of the prior probability for each behavior, we count the number of frames that have been annotated accordingly. The results are shown as percentages in Figure 4.2.

When durations are considered, the imbalance between the solitary actions and interactions becomes more extreme. In both datasets approxi-

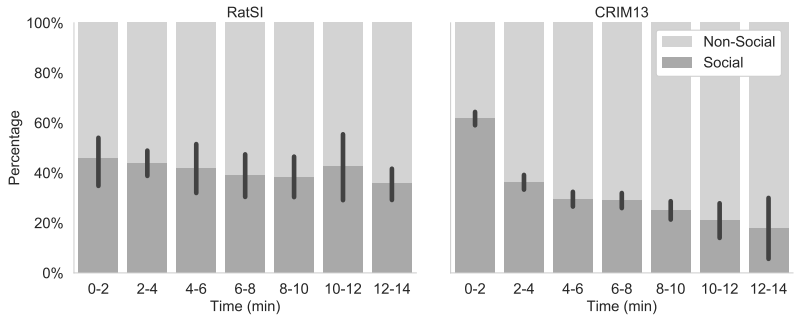
mately 60% of all frames belong to a solitary category leaving around 40% of interesting and relevant interactions. The increased imbalance is due to solitary behaviors that are in comparison considerably longer than the interactions. Also among the interactions, the figures vary. In RatSI, *pinning* and *nape attacking* are the least represented interactions with 0.6% and 1.0%, respectively. Similarly, *circle* and *chase* together account for less than 2% of all frames in CRIM13.

To summarize, the social activities of the rodents in the datasets are fewer in number than their solitary actions. Consequently, there is a substantial imbalance among the prior probabilities for the occurrence of each behavior. Such an imbalance presents challenges for automated recognition methods in both training and prediction. Small classes that are underrepresented in the set of training examples can lead to an inaccurate classification model as relevant variations of the behavior may be missing. During prediction the classifier may be biased towards predicting the majority class as this will, on average, cause fewer misclassifications. We need to address the imbalance with appropriate measures, for example by weighing the smaller classes relatively higher during learning and evaluation.

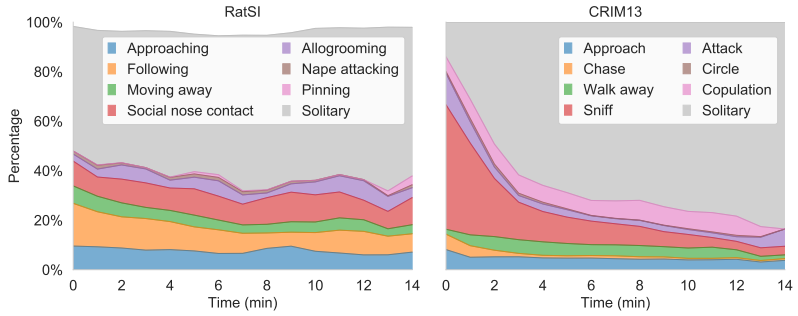
4.1.2 *Temporal Structure*

In the previous section, we found that the ratio between social and non-social behavior is approximately 40% to 60%, averaged over an entire dataset. We are interested in whether this ratio changes over the duration of a video (in our datasets up to 15 min) as this would affect the prior probability. Additionally, we examine whether the occurrence of one interaction makes it more likely for another interaction to occur right afterward. Such increased or decreased occurrence probabilities could be indicative for a temporal dependency across interactions.

We first address the ratio between social and non-social behavior. We compute the ratio over time as follows. Every video is partitioned into intervals of two minutes: {[0…2 min), [2…4 min), . . . }. Within each interval we count the number of social and non-social frames. We average the resulting ratios over the videos while keeping the partitioning into intervals. The CRIM13 videos start and end with only the resident mouse in the cage, where thus only solitary behavior can be observed. Since the time of introduction and removal of the second mouse varies across videos, we crop the videos before partitioning to the part where both mice are present. This ensures that we do not erroneously overestimate the amount of non-social



(a) Social vs. non-social behavior across videos (RatSI: 9, CRIM13: 161) in time bins of two mins. Error bars show standard error of mean.



(b) All behaviors except *unknown* and *other* labels. Areas show mean value across all videos.

Figure 4.3: Occurrence of behaviors over the course of the experiment videos.

behavior. RatSI videos start with the introduction of the second rat and therefore do not require any pre-processing.

Figure 4.3a shows the ratio within each interval averaged across all videos. In RatSI the percentage of social behavior seems to decrease slightly over time but the large variance among videos does not allow for a conclusive judgment. In contrast, there is a clear decline in social activity in CRIM13 from approximately 60% in the first two minutes to around 20% in the end just before the intruder mouse is removed. Clearly, interactions do not occur evenly in time and therefore the prior probabilities are not constant.

If we break down the social behaviors into separate interaction classes, we see that *sniffing* behavior is the main contributor to the social activity in CRIM13 (Figure 4.3b). Its gradual decline also decreases the overall amount of social behavior over time. In RatSI no such clear distinction is found as all interactions vary only modestly in frequency over time. These

activity profiles highlight a structural difference between the experiment setups of RatSI and CRIM13. In CRIM13 an intruder is placed in the homecage of the resident mouse which induces strong reactions in the resident. The thorough inspection (*sniffing*) of the intruder plays a dominant role in this first reaction. In RatSI the rats meet on neutral ground after having been isolated. Their desire for social interactions is expressed in a range of behaviors whose frequencies do not decline as strongly as in CRIM13 in the first 14 minutes.

Besides long-term changes in the interaction frequency, some interactions may have a more direct relationship and regularly occur in succession. To analyze whether interactions are related temporally, we count how often two behaviors occur immediately after each other. This allows us to examine transition probabilities, that is the probability that a specific interaction will occur right after another one. It is beyond the scope of this chapter to perform a comprehensive sequential analysis of rodent behavior and we refer to other literature [5] for studying sequences of higher order and statistically analyzing such sequences.

We formulate the probability of a transition from one interaction to another as the conditional probability $P(E_1|E_0)$. As before, the random variables E_0 and E_1 represent the event that a given behavior belongs to a specific category. Compared to the prior probability of a single event $P(E_1)$, the added constraint is that event E_1 occurs immediately after event E_0.

To determine the conditional probabilities, we first count how often each possible sequence of two consecutive interactions occurs in the dataset. For K behavior categories, this yields $K^2 - K$ values. We subtract K because a behavior event cannot repeat itself by definition (e.g., *approach-approach*). We then divide these values by the number of occurrences of the first behavior, which results in the empirical measure of the conditional probability:

$$P(E_1|E_0) = \frac{N(E_0E_1)}{N(E_0)} \tag{4.1}$$

where $N(E_0E_1)$ counts the occurrences of the E_0E_1 sequence and $N(E_0)$ the occurrences of behavior event E_0.

We display the probabilities for RatSI and CRIM13 in the transition matrices in Figure 4.4. The matrix contains the conditional probabilities for the transition from an interaction in a row to an interaction in a column: $P(\text{column}|\text{row})$. Note that each row sums to one since there is always a succeeding action.

In both datasets, a number of transitions appear more often than others. In RatSI, for example, an approach is succeeded by social nose contact with a 63% chance. In CRIM13, after a mouse walks away from the other,

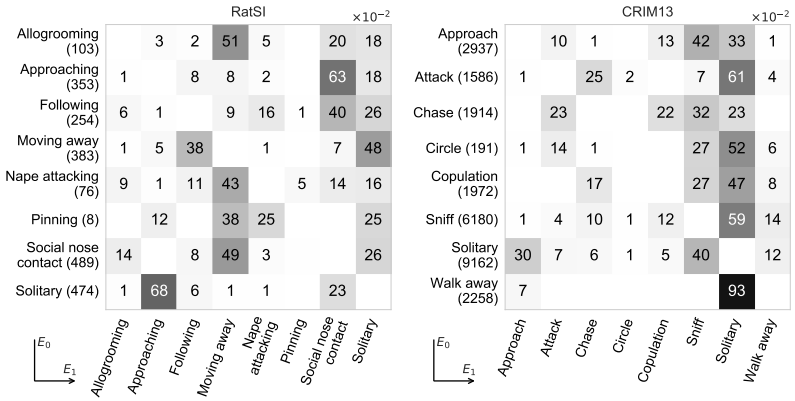


Figure 4.4: Transition matrix showing the conditional probabilities $P(\text{column}|\text{row})$. Values < 0.005 omitted. Values in brackets are absolute counts $N(E_0)$.

it almost always (93%) engages in solitary activities, unless it turns around straight away to approach the other mouse again (7%). Similarly, some transitions barely occur at all. For one part this is simply due to logic: a rat cannot approach another if it already established social nose contact. The other part is formed by behavioral patterns: the mice in CRIM13 apparently circle each other before attacking, but they never circle before copulation.

When we consider the automated recognition of interactions, a model of the temporal relationship between interactions is potentially helpful. It could disambiguate two similar interactions based on what happened before or after. Such a model could be derived from a transition matrix. We want to emphasize however that the transition probabilities as shown here are biased by the chance of each behavior occurring individually. For instance, solitary activities are more frequent than social interactions and are therefore just by chance more likely to occur after any other event. Consequently, the *solitary* columns contain relatively high values in every row. A proper temporal transition model only encodes those relationships that exceed (or fall short of) the level of chance occurrence [5]. The model is then independent of the prior probability of each behavior individually.

In this thesis we refrain from using a temporal model for two reasons. First, the temporal model has to be learned from behavior data and therefore only reflects the relationships that were present or absent in that data. The model can become invalid when the animals grow older or when we use animals from another genetic background. We want our recognition method to be applicable to data that exhibit different temporal relationships from the training data. Second, the sequential analysis of behavior

and the discovery of abnormal temporal patterns is often part of the study of animal behavior [20, 32, 144]. Incorporating a model of *typical* temporal patterns in the measuring method and eventually encountering *abnormal* temporal patterns can have precarious effects and may jeopardize the results of the behavior analysis.

4.2 Manual Annotations and Inter-Annotator Agreement

Up to now we have considered the manual annotations in the datasets as the *ground truth* and have assumed that every video frame shows exactly one behavior which is labeled as such. As we have discussed in Section 2.1.2, these assumptions do not generally hold in practice. Sometimes two annotators label the same frames differently. In this section, we assess the inter-annotator agreement in our own dataset.

To estimate the agreement, we need to compare the annotations from different annotators for the same videos. Because the available datasets have only one set of annotations, we asked three collaborating neuroscientists to annotate additional videos. The videos are comparable to the RatSI videos in terms of acquisition environment and animals, and the same behaviors are annotated. Note that at the time of this experiment, the RatSI videos were not available yet.

We analyze the agreement among the three annotators on a frame-by-frame basis as well as on an event-basis. We measure the overall agreement in terms of the percentage of frames for which the annotators agree. Studies that report the annotator agreement typically report the percentage agreement. In addition, we investigate whether the level of ambiguity is different for certain behavior categories. We further measure the temporal variations of annotated behavior events among the annotators. These additional measures allow us to gain novel insights into the ambiguity of social interactions.

4.2.1 *Experiment*

The participating annotators scored three videos, each with a length of 5 minutes (about 200 behavior events per video), containing rat social interactions identical to the interactions in RatSI. The same behavior definitions were followed (see Table A.1 in Appendix A for details). The instructions that were given to the annotators emphasized the importance of precisely scoring start and end times so as to avoid systematic delays. On the one hand, labeling with high temporal accuracy is more time-consuming because the annotator often needs to seek backwards through the video in

order to determine the precise start of an interaction. On the other hand, it ensures that annotations, videos and features align perfectly in time, which is required if the data is used for training a behavior classifier. The annotations were obtained using Noldus The Observer XT 12, with which all annotators were familiar before the experiment.

4.2.2 *Results*

The results are broken down into three parts, each taking a different perspective on the inter-annotator agreement. First, we look at the total agreement among each pair of annotators – the most common measure for the agreement. We then provide a novel analysis of the ambiguity of specific behavior categories and finally the temporal variations in the annotations.

4.2.2.1 *Overall Agreement Among Annotators*

Counting the frames on which annotators agree gives us a measure of agreement in terms of an overall percentage. The large fraction of solitary behavior in the videos, which we assume to be less ambiguous and therefore easier to label in agreement with others, may dominate the overall percentage and hide from us insights into the most relevant behaviors: the social interactions. To sketch a more comprehensive image, we isolate the social behaviors and measure the agreement among them.

Lacking a ground truth that tells us when interactions occur in the video, we rely on the given annotations to determine the relevant, social parts of the videos. We take an inclusive approach and consider a frame to potentially contain an interaction if at least one of the annotators has labeled it as one. We thereby include the *other* and *unknown* labels because they are mostly assigned to interactions rather than solitary activities. By this criterion 67.5% of the frames are considered social behavior.

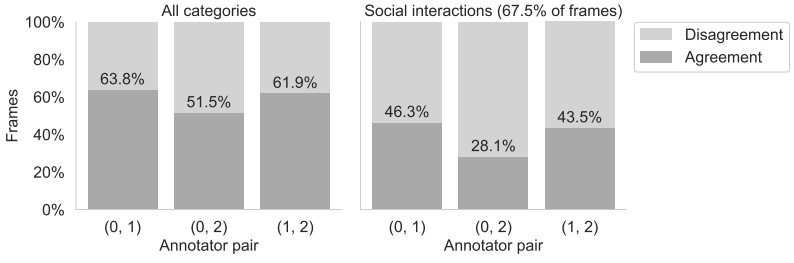


Figure 4.5: Percentage of frames in which an annotator pair agrees, considering the whole videos (left) or only potential interaction frames (right).

We first look at the percentage of frames in which a specific *pair* of annotators agrees. Figure 4.5 shows, on the left, the ratio between agreement and disagreement among all behavior categories including solitary activities and, on the right, the ratio among the social interactions. On average, two annotators agree in 59.1% of the frames considering all behaviors and in 39.3% of the interaction frames. The agreement on the social interactions is considerably lower than on solitary behavior. Annotator pair (0,2) generally exhibits a lower agreement than the other two pairs which achieve comparable percentages.

The percent agreement is subject to overestimation because it does not take into account the chance level of agreement. In other words, two annotators that label a video with entirely random annotations would still show a certain level of agreement. A popular measure that corrects for chance level agreement is Cohen's κ [23]. It is widely used to determine the inter-annotator agreement in behavior studies. Cohen's κ is computed as the fraction of agreement that exceeds the level of chance agreement:

$$\kappa = \frac{p_o - p_e}{1 - p_e}. \tag{4.2}$$

Here, p_o is the observed agreement which is equal to the percentages reported in Figure 4.5, and p_e is the expected chance agreement if the annotators would randomly assign labels. If the observed agreement is equal to the chance agreement, then $\kappa = 0$. Perfect agreement would result in $\kappa = 1$.

The chance agreement is subject to the prior probabilities with which each annotator picks a random label. These probabilities are computed from the observed labels. If the perceived prior probabilities differ between the annotators, then this difference alone accounts for some fraction of the disagreement. Cohen suggests to compute this fraction and report the corresponding value $\kappa_{\max}$ that can maximally be achieved under the different priors (thus < 1 if priors differ). He describes the intuition behind a potentially low $\kappa_{\max}$ as an indication for imprecisely defined behavior categories that leave room for different interpretations. $\kappa_{\max}$ is calculated the same way as κ but instead of the observed agreement the best possible agreement $p_{\max}$ that satisfies both observers' priors is used:

$$\kappa_{\max} = \frac{p_{\max} - p_e}{1 - p_e} \quad \text{with} \tag{4.3}$$

$$p_{\max} = \sum_{i=1}^{K} \min\left(P_A(F_i), P_B(F_i)\right) \tag{4.4}$$

where $P_A(E_i)$ and $P_B(E_i)$ are the prior probabilities which which the two observers assign the category i to a random frame, respectively.

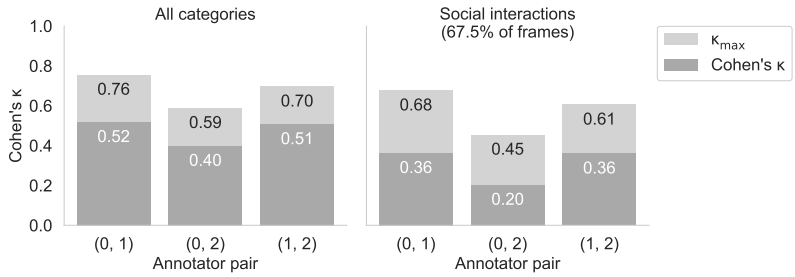


Figure 4.6: Agreement in terms of Cohen's κ, considering the whole videos (left) or only potential interaction frames (right).

The values of κ and $\kappa_{\max}$ for the annotator pairs in our experiment are given in Figure 4.6. Compared to the percent agreement, the values are lower but show the same pattern. Annotator pair (0,2) exhibits the lowest agreement, while the other two pairs are on par. Notably, $\kappa_{\max}$ is lower when considering only the social interactions. Following Cohen [23], we may argue that the social behaviors are prone to more subjectivity than the solitary category.

4.2.2.2 *Ambiguity of Interactions*

Instead of considering each pair of annotators separately, we can also look at agreement from another perspective and analyze the level of ambiguity in the videos. To estimate the ambiguity we consider the annotators as experts casting a vote on every frame. We can then count the frames for which all, two or none of the annotators agree. When all annotators agree, the frame can be considered clear and unambiguous. With one deviating vote, there may be some doubt and ambiguity about the behavior. Finally, if none of the annotators agree, each casting a different vote, the behavior in that frame is apparently difficult to categorize.

Figure 4.7 illustrates the percentages of frames falling in each of the three agreement categories. As before we compare the percentages among all behavior categories with only the social interactions. Almost half of the videos, 48% of the frames, is relatively clear and all annotators agree, while there is some ambiguity in a third of the frames causing one annotator to vote differently. Less than 20% are truly ambiguous. The parts of the videos that show social interactions seem to be more ambiguous: omitting the solitary behavior from the analysis decreases the portion in which all annotators are in agreement to 34.6%. The portions in which the annotators disagree partially and entirely increase to 41.7% and 23.7%, respectively.

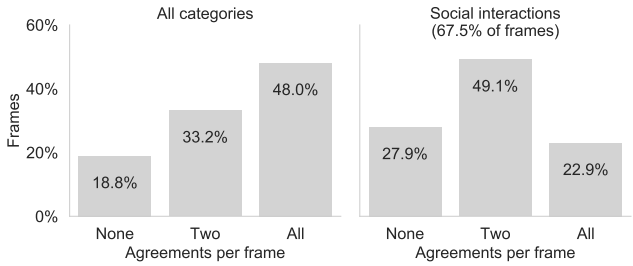


Figure 4.7: Percentage of frames in which none, two or all three annotators agree. Considering the whole videos (left) or only social behavior (right).

Irrespective of the considered behavior categories, in roughly 80% of the frames at least two annotators decided for the same label. This demonstrates that in a substantial portion in the videos there exists at least some agreement about the behavior of the animals. Although *some* agreement is no guarantee for having found a legitimate set of annotations, we value this outcome positively as it shows that the visual information in the video frames enables humans to recognize the behavior; information that can potentially be exploited by a computational recognition method.

Until now we have used an aggregated measure of agreement, summarizing both time and behaviors into one overall percentage. We now want to draw a more detailed image and assess the agreement and the ambiguity for specific interaction categories. For each category we look for interactions that are scored by at least one annotator and then measure how often one or two other annotators agree. Because the interactions occur with different frequencies and durations in the videos, we normalize the agreement for each category independently. The left plot in Figure 4.8 shows the number of frames assigned to each category by at least one annotator. In the right plot we display the three levels of agreement for each behavior.

Following and *solitary* not only occur most often, they also exhibit the highest agreement. In 68% and 85% of the frames, respectively, at least one other annotator also scored *following* or *solitary*. On *pinning* two annotators agree more than 55% of the time, although there are only few frames in which all three annotators agree (14%). For all other categories the annotators disagree more often than they agree. More than two thirds of the frames are scored by only a single annotator at a time. In fact, one of the annotators never agreed on any *allogrooming* frame. Clearly, this annotator had a different interpretation of what constitutes *allogrooming* compared to the other two.

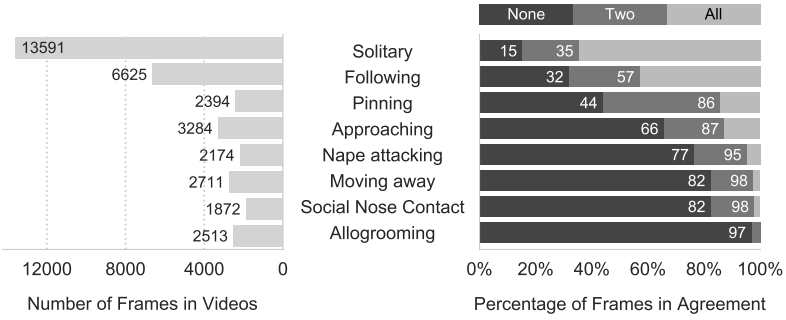


Figure 4.8: Level of agreement broken into behavior categories. Left: number of frames labeled by at least one annotator. Right: percentage of frames in which none, two or all three annotators agree.

4.2.2.3 *Agreement on Behavior Events and their Temporal Variations*

As a final aspect of this inter-annotator agreement study, we want look at the agreement among behavior events. In contrast to video frames, comparing events is a more complex matter. Suppose two observers have annotated the exact same interaction. Even if they both recognized the interaction and assigned the same behavior label, their view on when exactly the interaction begins and ends may still vary. The question is when we can consider two event annotations in time to be equivalent.

To get an intuitive idea about the different levels of agreement of event annotations, we show a number of examples in Figure 4.9. The examples have been annotated with the same behavior label but they vary in the amount of overlap. Here, we compute the overlap as the intersection di-

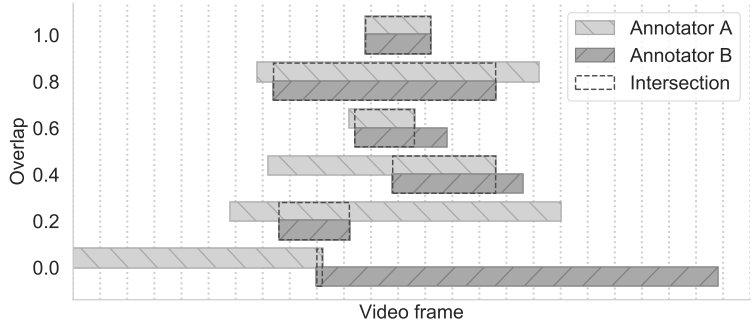


Figure 4.9: Examples of annotations with varying overlap (intersection / union).

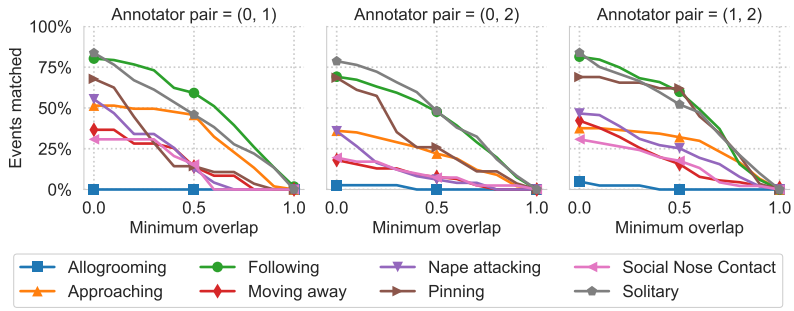


Figure 4.10: Percentage of events in agreement with varying overlap criteria. Overlap is intersection over union.

vided by the union. That is, the number of frames for which the annotations overlap divided by number of frames annotated by either observer. Although this is only an illustrative exercise, it demonstrates the different types of temporal variations that we could encounter in our data. In particular, we observe cases in which both start and end time are delayed (examples with overlap 0.4 and 0.6), and cases in which one annotation is a segment of another (examples with overlap 0.2 and 0.8). We further find perfect alignment with overlap 1 but also almost disjoint events without considerable overlap.

When computing the average agreement for event annotations, the desired overlap is a free parameter. We vary this overlap threshold to sketch the complete image. Figure 4.10 shows the percentage of events that are considered equivalent under an increasingly restrictive overlap criterion. The percentages relate to the total number of events (sum of both annotators), listed in Table 4.1.

Naturally, the number of events that are successfully matched decreases as the overlap criterion becomes more restrictive. In accordance with our findings at the frame-level, *solitary* and *following* events achieve the highest agreement. *Allogrooming* events are rarely annotated by two observers and essentially do not coincide with the annotations of the third observer. For most behaviors except *solitary*, *following* and *pinning*, more than half of the annotated events do not match any event of another observer.

4.2.3 *Conclusion*

All in all, the results from this section demonstrate that rat social interactions exhibit a considerable amount of ambiguity that makes annotation a challenging task. Although in total more than three quarters of the videos

Table 4.1: Number of annotated events per annotator.

Annotator:	0	1	2
Allogrooming	2	8	74
Approaching	62	43	138
Following	135	88	79
Moving away	23	48	132
Nape attacking	21	26	77
Pinning	26	30	28
Social Nose Contact	15	24	67
Solitary	132	77	84

are labeled in agreement by at least two of the three annotators, some of the interactions, such as *allogrooming*, show a remarkable low level of agreement. We want to note that the annotators that participated in this study all had experience with rat social behavior but had only little training in annotating it according to the given definitions. Thorough training and communication among the annotators would improve the consistency of the annotations within the group and likely lead to higher agreement scores. Nonetheless, we were able to identify three factors responsible for the disagreement among annotators: a) different interpretation of what constitutes a behavior category and possibly vague boundaries, b) varying attention to detail and the tendency to lump multiple interactions in immediate succession into one, and c) ambiguity of the exact beginning and end of an interaction.

4.3 Feature Representation

When human annotators score behaviors in a video, they make labeling decisions by interpreting the visual information in the images. Depending on the specific behavior they may consider the animals' pose, their relative position, and their movements. In this section we aim to verify that visual information extracted *computationally* from the video images indeed aligns with the behaviors as scored by the annotator. This investigation is a first step to demonstrate which interactions may be recognized automatically based on tracked animal locations and body parts.

Specifically, we analyze the distribution of the feature values extracted from the tracking data in the datasets. Comparing the distribution within and across interaction categories gives us an idea whether those features are meaningful for automated classification. To this end, we first conceptu-

ally group the interactions into two groups: trajectory-related and contact-related interactions. Trajectory interactions are mainly expressed in terms of whole-body motion, that is the movements of the rats through the space and in relation to each other. *Approaching* and *following* are examples for trajectory-related interactions. Contact interactions express coordinated motion or manipulation of specific body *parts*. In a nape attack, for instance, a rat attempts to reach the neck area of the other rat with its snout or paws to bite and pull the fur. Hence, the relative positioning and detailed motion of body parts convey crucial information for differentiation. Table 4.2 lists the interactions assigned to each cluster.

Note that the chosen grouping is based on the visual properties of the behaviors rather than their function in an ethological sense. From a functional perspective the two contact-related interactions *nape attacking* and *sniffing*, for example, fulfill different purposes and are therefore dissimilar.

Table 4.2: Clustering of trajectory- and contact-related interactions.

Trajectory		Contact	
RatSI	CRIM13	RatSI	CRIM13
Approach	Approach	Allogrooming	Attack
Following	Chase	Nape attacking	Copulation
Moving away	Circle	Pinning	Sniff
	Walk away	Social nose contact	

4.3.1 *Trajectory-related Interactions*

We first consider trajectory interactions such as *approaching*, *moving away* or *following*. During such interactions the animals move in order to get closer or further away from each other or to maintain a certain distance while moving along a similar path. They differ in how the distance between the animals changes over time, in the velocity with which they move and in the orientation relative to each other. We will first show how to derive these basic trajectory features from the tracking data and then analyze whether their values coincide with what we expect for each interaction.

The tracked animal locations that are provided with the datasets allow us to compute velocity and orientation of each rodent and the distance between them. Let us denote the position of a body point at time t by $\boldsymbol{p}(t) = \left[\begin{smallmatrix} p_x \\ p_y \end{smallmatrix}\right]$. To identify a specific body point of an animal (c for center point, n for nose point, b for tail-base point) we indicate the body point in subscript and the animal in superscript. For example, the center point of

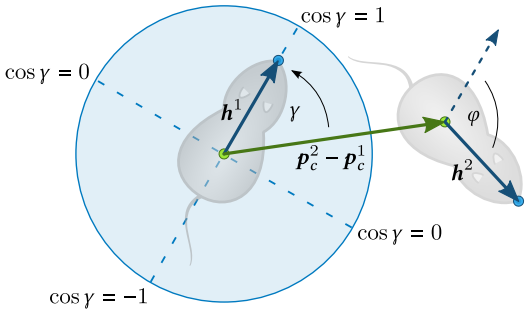


Figure 4.11: Relative orientation between two rodents captured by the relative angle φ and relative heading direction γ.

animal 1 is denoted by $\boldsymbol{p}_c^1(t)$. For the sake of clarity we omit the time (t) unless it is necessary to distinguish between values of different frames.

For the distance between the animals we consider the Euclidean distance between their center points, given by

$$d_{cc} = \left\| \boldsymbol{p}_c^1 - \boldsymbol{p}_c^2 \right\|. \tag{4.5}$$

The change of the distance over time is determined by the difference of the values between consecutive frames divided by the time between frames Δt:

$$d'_{cc}(t) = \left(d_{cc}(t) - d_{cc}(t - \Delta t)\right) / \Delta t. \tag{4.6}$$

Velocity is estimated by the displacement of the center point between two consecutive frames:

$$v_c(t) = \left\| \boldsymbol{p}_c(t) - \boldsymbol{p}_c(t - \Delta t) \right\| / \Delta t. \tag{4.7}$$

Similarly we can compute distances among any combination of body points as well as velocities of body points other than the center. Note that the velocity features involve a division by the time between two frames, which makes the features invariant to the video frame rate of the considered dataset. RatSI and CRIM13 videos are recorded at a rate of 25 fps.

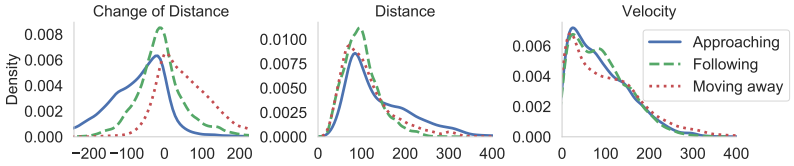
The orientation of the animal informs us about the direction in which it is facing. Choosing the right reference frame to express this direction is particularly important for interactions. For *approaching* and *moving away*, for example, it is the orientation *relative* to the other animal that conveys the most essential information. Global reference frames such as the image axes or the cage walls are typically less relevant. We consider two features to represent the relative orientation, namely: the relative angle φ and the

relative heading direction γ. As depicted in Figure 4.11, φ is the absolute angle between the two animals' head vectors $\boldsymbol{h}^1$ and $\boldsymbol{h}^2$. It ranges from 0 to 180 degrees when the animals are facing the same or opposite directions, respectively. The head vectors are derived from the center and nose points, that is, $\boldsymbol{h}^i = \boldsymbol{p}_n^i - \boldsymbol{p}_c^i, i \in \{1, 2\}$.

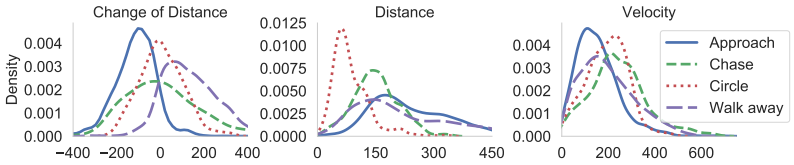
The relative angle φ captures the orientation from a global perspective but it does not convey information from the perspective of a specific animal. In contrast, the second orientation feature, γ, captures where in one animal's environment the other animal is (e.g., in front, behind, next to). It is invariant to the orientation of the other animal and can hence disambiguate between towards or away from another animal. The values of γ cover the full 360 degree circle and thus also distinguish between left of and right of. We can eliminate the left/right notion from the feature by computing $\cos(\gamma)$ which then ranges between -1 and 1, and is symmetric with respect to the head vector $\boldsymbol{h}$.

Note that the head vector cannot be derived from the nose point in CRIM13 because only the center point location is tracked. For CRIM13 we estimate the head vectors from the direction of movement, assuming that the mice typically walk forward as suggested by the dataset authors [18]. The disadvantage of estimating orientation from movement is that it is impossible if the mouse is not or barely moving. In addition, tracking noise decreases the accuracy of the estimate especially during slow movements. We compute the direction of motion from the head vector that is now given by the displacement vector of the center point: $\boldsymbol{h} = \boldsymbol{p}_c(t) - \boldsymbol{p}_c(t - \Delta t)$. To avoid unreliable orientation estimates, we monitor the length of the displacement vector, which is proportional to the velocity of the animal, and skip the computation if the velocity is lower than a third of the animal length per second. We fill in the missing orientation values by interpolating linearly between the last known orientation and the moment when the animal is moving again.

Let us now look at what values the features take on during trajectory-related interactions. These observations are based on behavior labels given by a human annotator. We calculate the probability density function of the values along each feature dimension using a kernel density estimation with a Gaussian kernel [96]. Intuitively, the density function can be seen as a continuous histogram over the occurring values. In Figure 4.12 we show the density estimates for the distance d_{cc}, velocity v_c and change of distance d'_{cc}. The change of distance is the most discriminative feature of the three, showing that the distance typically decreases during an *approach*, increases when *moving away*, and remains reasonably unchanged during *following/chasing* as well as *circling*. It seems that neither distance



(a) RatSI



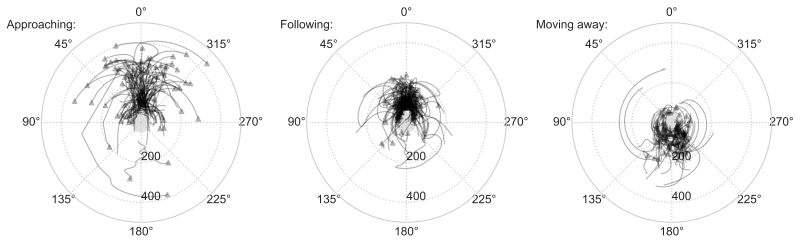
(b) CRIM13

Figure 4.12: Density estimation of distance and velocity features for each trajectory-related interaction, normalized to integrate to one.

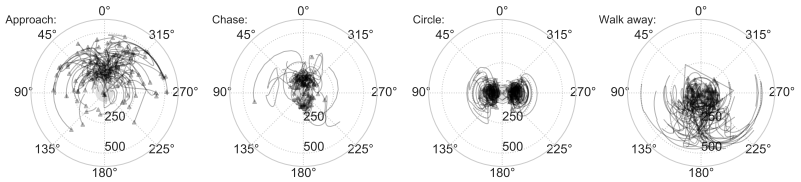
nor velocity convey much information to differentiate between the trajectory interactions. Nonetheless, both are likely to contribute to discriminating them from other actions, such as solitary activities which are often performed at a larger distance from each other. We further observe a distinct difference in scale between the datasets. Both, velocity and change of distance, are approximately twice as large in CRIM13 than in RatSI. The scale is largely determined by the video resolution and distance from the camera but also by the animal's size and natural velocity.

In Figure 4.13 we plot the angle γ and the distance d_{cc} along time of 100 random interactions per category in a polar coordinate system. The 0° angle corresponds to *toward* the other animal, while 180° is *away from* it. The outer circle diameter corresponds to the diagonal dimension of the observation cage (the maximum distance between the animals). The start of each interaction is marked by a triangle.

We observe several patterns emerge, matching our expectations of the nature of the interactions. The *approach* trajectories move from the outside to the inside, decreasing the distance between the animals over time. The orientation converges toward the other animal. In CRIM13 the variance among the orientation values seems larger. Similarly, *following* and *chasing* display a clear pattern toward the other animal maintaining a smaller distance than during an approach. *Moving* and *walking away* are the conceptual opposites of approaching and indeed show opposite trajectory patterns. They start at the center and move outside, increasing the animals' distance, while the orientation is mostly away from the other an-



(a) RatSI

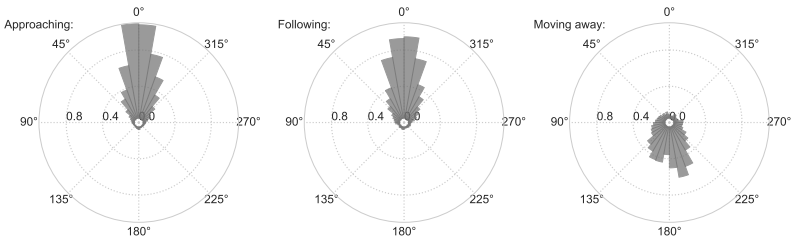


(b) CRIM13

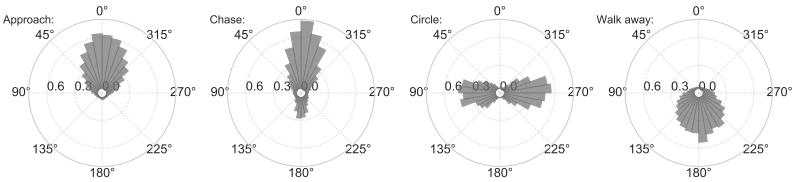
Figure 4.13: Trajectories of interactions (100 per category randomly selected) in polar space spanned by orientation feature γ (angle) and animal distance d_{cc} in pixels (radius). The approximate animal length (nose to tail-base) is in RatSI: 100 px, CRIM13: 150 px.

imal. Again the variance among distance and orientation values is larger in CRIM13 and the interactions appear generally longer. In RatSI, *moving away* interactions seem relatively short and end before a large distance between the animals has been established. The *circling* interaction is only observed in CRIM13 and is scored when a mouse is moving in circles around the other mouse. The orientation feature reflects the circles clearly by being constrained to either the left or the right of the other animal, depending on whether the mouse moves in clockwise or anti-clockwise direction. During circling it stays close to the other mouse. The histograms of the values of γ in Figure 4.14 support our observations of the stereotypical orientations during the interactions.

Besides the dominant patterns we notice a number of exceptions such as the few *approach* examples in RatSI that appear to be oriented away from the other animal and yet move closer. Such outliers are caused by tracking errors, for instance if the nose and tail-base points are confused and thus the orientations wrongly estimated, but also by annotation errors, if the animal that is being approached is erroneously labeled as the approaching animal. CRIM13 exhibits a generally larger variance in orientation features. This may be due to the fact that for CRIM13 the orientation of each mouse



(a) RatSI



(b) CRIM13

Figure 4.14: Histogram of the orientation feature γ per trajectory-related interaction, normalized to integrate to one.

has to be estimated from the direction of movement which is generally less robust than deriving it from the center and nose points.

Overall, the patterns we see emerging from the examples are intuitive. The relative orientation and the change in distance appear to capture the essence of the trajectory interactions. Nonetheless, there are still overlapping patterns, for instance between *approaching* and *following*, which will make it harder to distinguish between the two automatically. In addition to the overlap in feature values, the automated recognition needs to deal with the unknown temporal boundaries. In the shown examples we were given the exact start and end of the interactions by the human observer. During automated recognition the start and end points have to be determined jointly together with the interaction category.

4.3.2 *Contact-related Interactions*

We now turn to the representation of contact-related interactions such as *sniffing* and *nape attacking*. Such interactions have in common that they are performed at a close distance which makes them inherently different from the trajectory-related interactions. Rather than through whole body trajectories, the rodents interact using specific body parts including snout and paws.

RatSI provides the locations of three distinct body points: the nose, center and tail-base. We aim to express the relevant aspects of contact interactions in terms of the distances between those points and the change over time. In CRIM13, the lack of additional tracked body points prevents us from computing contact-related features.

We compute two body point distances, analogous to the center point distance in Section 4.3.1. As illustrated in Figure 4.15, we derive the Euclidean distance between the two nose points d_{nn} and the distance between the nose and the tail-base d_{nb} from the corresponding tracking points:

$$d_{nn} = \left\| \boldsymbol{p}_n^1 - \boldsymbol{p}_n^2 \right\| \quad \text{and} \tag{4.8}$$

$$d_{nb} = \left\| \boldsymbol{p}_n^1 - \boldsymbol{p}_b^2 \right\|. \tag{4.9}$$

The change of the distances over time is calculated as the difference in consecutive frames divided by the time between frames Δt:

$$d'_{nn}(t) = \left(d_{nn}(t) - d_{nn}(t - \Delta t)\right) / \Delta t \quad \text{and} \tag{4.10}$$

$$d'_{nb}(t) = \left(d_{nb}(t) - d_{nb}(t - \Delta t)\right) / \Delta t. \tag{4.11}$$

We further compute the velocity of the nose point to capture differences between interactions with slow head motions, such as *social nose contact*, and faster motions such as *nape attacking*. Similarly to the center point velocity (Equation 4.7), the nose velocity v_n is given by

$$v_n(t) = \left\| \boldsymbol{p}_n(t) - \boldsymbol{p}_n(t - \Delta t) \right\| / \Delta t. \tag{4.12}$$

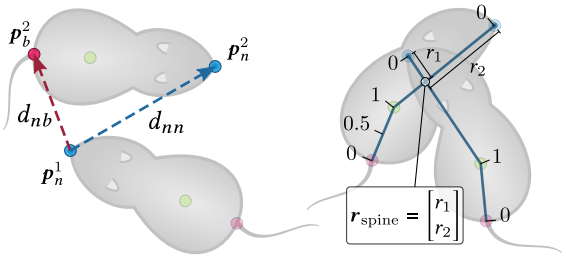


Figure 4.15: Body point distances d_{nn} and d_{nb}, and spine overlap ratio r_{spine}.

To be able to detect overlapping bodies, we extract a simplified spine structure for each rodent from the three body points. The spine consists of two line segments: the lines connecting the tail-base with the center and the center with the nose. In Figure 4.15 the spines of the two rats on the right are indicated by solid lines. To capture the extent of overlap as well as the contact location, we determine the point of intersection between the

two spines. We represent the intersection point as coordinates along each one-dimensional spine, which together form the two-dimensional vector $\boldsymbol{r}_{\text{spine}}$. The coordinates are assigned values in the range $[0, 1]$, where 1 corresponds to the center point and 0 to either end of the spine (nose and tail-base). The values between are interpolated linearly. If the spines do not intersect, $\boldsymbol{r}_{\text{spine}} = \left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]$. If the spines intersect more than once, the vector with the higher mean value is taken.

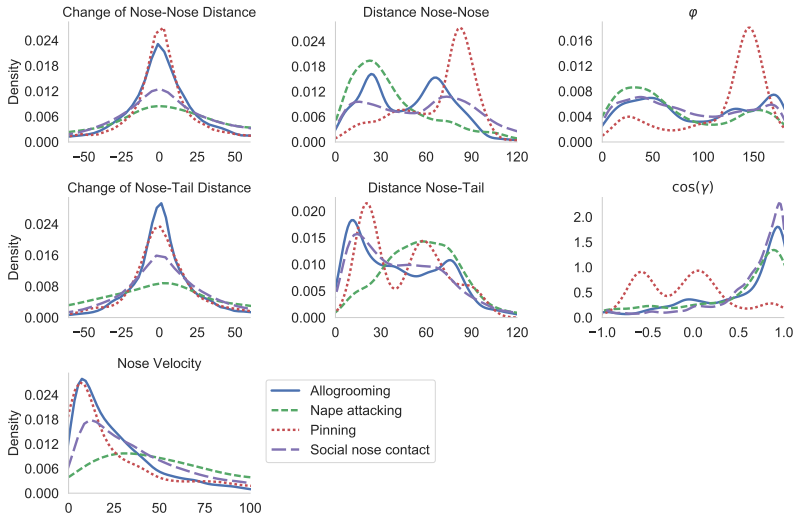


Figure 4.16: Density estimations of body point distances and their change in time, relative orientation and nose velocity computed for contact interactions in RatSI, normalized to integrate to one.

Let us look at the values of these features for the contact interactions in RatSI. To get an idea whether the features capture the differences between interactions, we compare the density estimates among the interactions in Figure 4.16. We find a few properties that seem to distinguish interactions from others. *Pinning* instances for example frequently exhibit high values in the relative orientation φ and the nose-nose distance. *Nape attacking* in contrast occurs more often with a small nose-nose distance and a larger nose-tail distance. That is in line with the fact that the attacker attempts to reach the neck area of the other rat with its snout. These attacks seem to happen at slightly higher nose velocities compared to the other interactions. The change of distance does not appear to capture any distinct differences.

We examine the values of the spine overlap vector in a joint histogram of both vector elements in Figure 4.17. The values are sorted into six

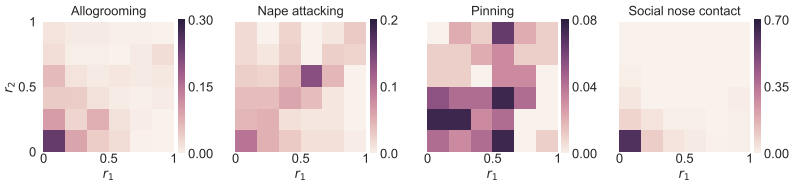


Figure 4.17: Histograms of two-dimensional spine intersection vector $\boldsymbol{r}_{\text{spine}}$ for contact interactions in RatSI.

equidistant intervals in each dimension. The darker the square, the more frames fall into the corresponding value range. The histograms include only frames with overlap, that is when $\boldsymbol{r}_{\text{spine}} \neq \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$. *Allogrooming* and *social nose contact* exhibit only minor overlap close to nose or tail. *Nape attacking* clearly shows more overlap, mostly concentrated half-way along either spine segment and with fewer frames in the lower right corner of the matrix. Frames located in this corner would show the attacking rat covering with its body center either the head or the tail of the other rat; neither of which would classify as a *nape attack*. Lastly, the overlap values for *pinning* are more spread out and suggest that a number of different overlap postures appear in the videos. Given that there are relatively few *pinning* interactions in RatSI, it seems to be challenging to identify a single, characteristics *pinning* posture using the spine overlap.

All things considered, we observe fewer clear patterns in the features that distinguish the contact interactions compared to the trajectory interactions. The reason is two-fold. First, the three body point locations are less suited to represent contact interactions. The paws for example are not tracked, yet play a central role in some of the interactions including *nape attacking*. Tracking only three body points may prove insufficient for reliably recognizing contact interactions automatically. Second, contact interactions occur per definition at close distances and with occlusions. These are challenging situations for the tracking algorithm which in turn is likely to be less accurate in locating the nose and tail-base points. These inaccuracies have a particularly strong effect on the orientation and spine features, potentially rendering them unreliable during contact situations.

Although we could not find features that exhibit as clear distinctions as for the trajectory interactions, we are limited by our capabilities to visualize high dimensional data. A classifier, which operates in the higher dimensional space of all features together, may still be able to model the interactions and facilitate automated recognition.

4.4 Conclusion

We have analyzed the RatSI and CRIM13 datasets regarding aspects related to learning, applying and evaluating rodent interaction classifiers. These aspects highlight some of the challenges that researchers face in developing novel classification methods. For instance, an unbalanced prior distribution of the behaviors is a common property of these datasets, not only in our data but also in related work [18, 48]. The main concern is the collection of sufficient training examples of the rare behaviors, and a fair evaluation that prevents the more frequent behaviors to dominate the error measure. Less attention is given to the fact that different datasets may have different prior distributions, that the priors may change over time, and that they are hard to estimate for new data for which no annotations are available. An investigation of the effects of varying prior distributions on the accuracy of automated annotation is left for future work.

The prior distribution is not the only variable property. The two examined datasets also showed substantial variations in the temporal and spatial extent of interactions, as well as the scale and variance of feature values. These variations are due to the different species, the cage, the acquisition setup, the camera resolution and the tracking algorithm. To be able to classify rodent interactions in arbitrary input videos, all combinations of these variations need to be incorporated into one common classification framework. The number of possible combinations alone limits the possibility to design such a framework manually. We are more likely to succeed with a general framework that automatically learns how to abstract from variations such as spatial scale. This approach appears to work well for other visual inference tasks such as object recognition [71] and human action recognition [131].

Lacking such a common framework yet, we have introduced meaningful numerical representations for movements and poses that are relevant for modeling rodent interactions. These representations form the basis for the automated annotation methods that we develop over the course of the following chapters. For trajectory-related interactions the change of distance and the relative orientation appear to be highly informative. The contact-related interactions seem more difficult to model from only a few body point locations. One of the challenges is the lower tracking quality in those contact situations which cause the little information that is available to be particularly inaccurate. We want to investigate the influence of the tracking quality on the classification accuracy in more detail and will therefore conduct a computational experiment involving varying degrees of tracking quality in the next chapter.

5

Tracking Quality & Feature Complexity

Movement and pose features derived from body point locations promise to be meaningful for automatic classification of rodent interactions. Tracking these body points throughout the videos is, as we discussed in Section 2.2.1.2, a challenging task. Occlusion and the similar appearance of the animals add to the complexity and can cause tracking errors in the form of misidentification of animals, confusion of body parts as well as inaccurate localization of those parts.

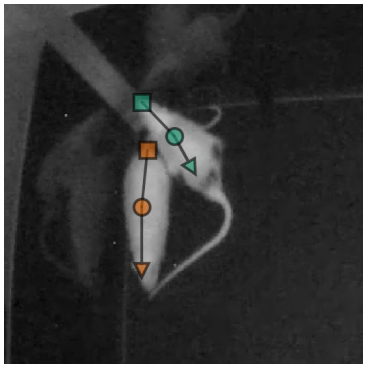
The quality of the tracking data is linked to the quality of the features. Tracking errors propagate through the feature computation where they potentially cause noisy and inaccurate estimations of, for example, the animal's pose, its orientation or velocity. Eventually, they may lead to misclassifications of the interactions with potential consequences for the behavior analysis.

In this chapter we aim at unraveling the effect of tracking quality on classification accuracy. We investigate the effect of common tracking errors, including misidentification and inaccurate localization, on the accuracy of the overall classification and on the recognition of specific interactions. To this end, we systematically vary the tracking quality in two orthogonal directions. We begin by incrementally correcting two types of tracking errors in Section 5.1. Then in Section 5.2, we derive three feature sets from the locations of a varying number of body points. Finally, we compare the classification performance using off-the-shelf classifiers in Section 5.3.

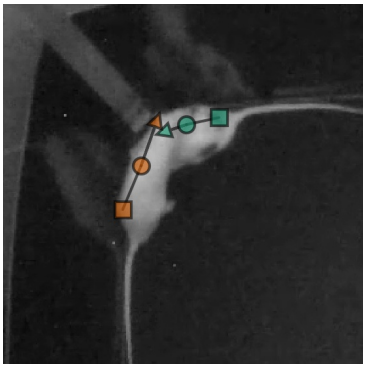
The result of this analysis separates the influence of tracking errors on the classification accuracy from other classification errors. This yields important information for future studies because it highlights the margin for improvement of the classification beyond solving the tracking task. It can help us to identify and prioritize the classification challenges to address in the future, regardless of the preferred tracking algorithm.

5.1 Eliminating Systematic Tracking Errors in YR

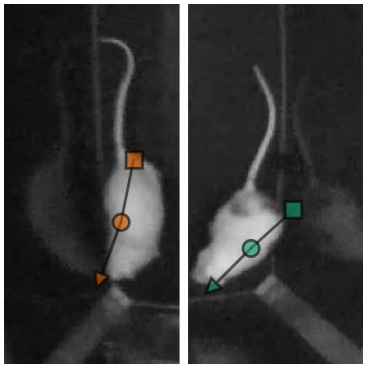
We investigate the effects of tracking errors and therefore need to isolate tracking from other factors that may influence the classification accuracy such as label noise. The YR dataset is better suited than RatSI for this task



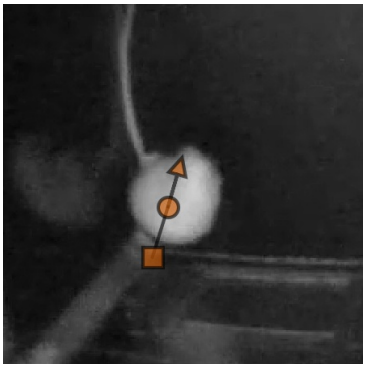
(a) Confusion of nose and tail-base points



(b) Occlusion during *allogrooming*

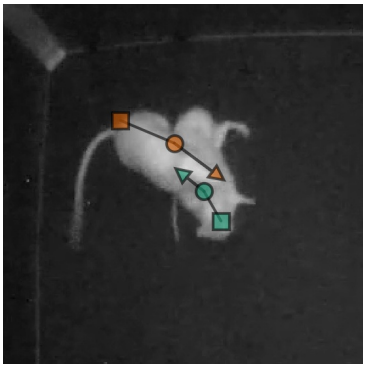


(c) Inaccurate pose due to reflection on cage wall



(d) Inaccurate pose due to self-occlusion

(e) Occlusion during *pinning*



(f) Occlusion during *pinning*

Figure 5.1: Examples of tracking errors. Body points: △ (nose), ◯ (center), □ (tail-base).

because YR consists of clean annotations and an equal number of selected clips of every behavior.

YR provides tracking data of three points on each animal body: the nose point, the center of mass of the body contour and the tail-base. We incrementally eliminate two types of tracking errors leading to three versions of the same dataset with varying tracking quality. We denote the initial, uncorrected version as `YR`. This version contains errors in the assignment of nose and tail as well as animal identities. While wrong identities may propagate through large portions of a video, a nose-tail swap is automatically corrected by the tracking algorithm as soon as the animal starts walking. This correction is based on the assumption that if a rat moves from one place to another, it does so by moving forward. Nose-tail swaps therefore occur predominantly in occlusion situations and last for only a short period of time. Occlusion can also lead to inaccurate localization of the body points. To give an overview of scenarios that may cause errors, we show a few uncorrected tracking examples in Figure 5.1.

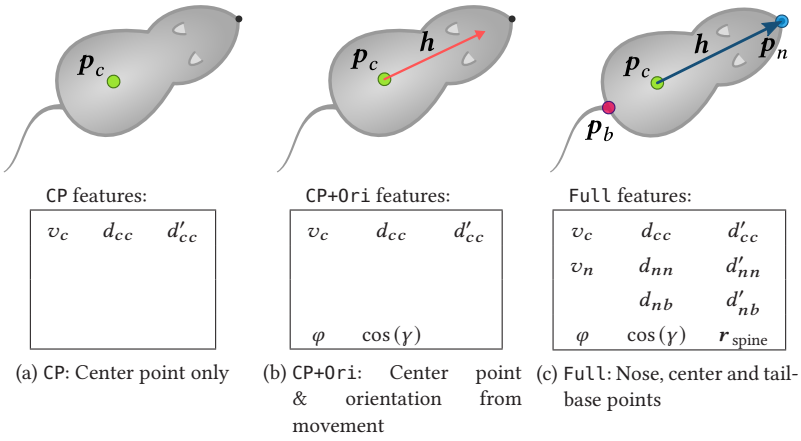
As a first correction step, we eliminate errors in the identity assignment by manually correcting identity swaps. Identities are not changed during fast, close-contact and occlusion situations. As the body point positions are often misplaced in those situations, it is impossible to assign the correct identities without correcting the positions first (e.g., see Figure 5.1e). We denote the dataset version with corrected identities as `YR-ID`.

In the second elimination step (`YR-ID+Loc`), we additionally correct the body point locations. As mentioned earlier most of the corrections are necessary in occlusion situations (e.g., Fig. 5.1b, 5.1e and 5.1f). The correction of body point locations eliminates nose-tail swaps, yields more reliable orientation values, and generally leads to smoother trajectories as noise in the positions is reduced.

Note that this correction process involves a substantial amount of manual work. The correction of the body point locations in particular takes multiple times longer than the duration of the videos as every video frame has to be inspected. Manual correction does not present a solution for increasing the tracking quality in practice.

5.2 Extracting Features from the Dataset Versions

We derive three feature sets with increasing level of detail as illustrated in Figure 5.2. In the first set, `CP`, we assume we are given only the center point position $\boldsymbol{p}_c$. This is the case in the CRIM13 dataset [18] but also other works take this approach [147]. Features we can derive from a single point



(a) `CP`: Center point only (b) `CP+Ori`: Center point & orientation from movement (c) `Full`: Nose, center and tail-base points

Figure 5.2: Tracking a rodent with increasing level of detail.

include velocity v_c, the distance between the animals d_{cc} and the change of that distance d'_{cc}.

For the second set, `CP+Ori`, we derive each animal's orientation from the direction of its movement. This allows us to calculate the relative orientation between the animals in terms of the absolute, relative angle φ and the relative heading direction $\cos(\gamma)$ as described in Section 4.3.1. Estimating the orientation from motion is sometimes unreliable as it requires robust tracking of the center point and is unavailable if the animal is not moving. Sidewards or backwards motion can also impair orientation estimates.

The third set, `Full`, solves the limitations of the `CP+Ori` set by exploiting the three body point locations $\boldsymbol{p}_b, \boldsymbol{p}_c$ and $\boldsymbol{p}_n$. Deriving the orientation from the nose and center points is more robust against small displacements due to noise and is possible even if the animal is immobile. Moreover, we can incorporate additional body point distances and estimate points of contact in occlusion situations. We include the nose-nose distance d_{nn}, the nose-tail distance d_{nb}, their derivatives d'_{nn} and d'_{nb}, the velocity of the nose point v_n and the spine intersection vector $\boldsymbol{r}_{\text{spine}}$. Refer to Section 4.3 for details on the feature computation.

5.3 Classification Experiments

To analyze the links between feature quality and recognition accuracy, we examine the effects of tracking errors alone (using the `Full` feature set) as well as in combination with the different feature sets. Let us first formulate the classification task.

5.3.1 *Classifying Interactions from Video Frames*

We formulate the task of recognizing rodent behavior in videos as a multiclass classification problem. The goal is to classify each video frame into one behavior category. Let $\mathcal{X}$ be the domain of possible feature vectors and $\mathcal{Y}$ a finite set of K behavior categories. We further denote the feature vector of frame i by $x_i \in \mathcal{X}$ and the corresponding label by $y_i \in \mathcal{Y}$. The task of the classifier is to predict for a given feature vector x the correct label y. Note that although the YR dataset would allow classifying the video clips directly instead of individual frames, we do not have this option in practice where videos are not segmented into separate clips per interaction.

To find a suitable classifier for our analysis, we compare seven off-the-shelf classifiers and then stick to one classifier for the remaining experiments in this chapter. We compare a large range of classifiers to be able to eliminate the specific classification model from the measured effect: Linear Discriminant Classifier (LDC); Quadratic Discriminant Classifier (QDC) (as in [145]); Support Vector Machines with Gaussian (SVM-RBF) and linear (SVM-Lin) kernels (as in [42]); LDA with k-Nearest-Neighbors (LDA+kNN); Random Forest (RF) (as in [48, 55]); and a Gaussian Mixture Model (GMM). This selection is based on related work complemented by basic classifiers with different approaches such as linear class separation (LDC), neighborhood sampling (kNN), and a generative model (GMM). The optimal parameters of the classifiers are selected by cross-validation, described below.

Classification accuracy can often be improved by scaling all feature dimensions to a common range [78]. To normalize scale differences, each feature is independently scaled to zero-mean and unit-variance based on the training data.

5.3.2 *Measuring the Classification Accuracy*

We assess the recognition accuracy in terms of the classification performance per interaction. We mainly look at the F1 score and, if appropriate, at precision, recall and confusions between specific classes. The F1 score is the harmonic mean of the precision (true positive predictions divided by total number of positive predictions) and recall scores (true positive predictions divided by the number of actual occurrences). The class scores range from 0, with no correct predictions, to 1 for the correct prediction of all frames. To obtain a single measure of performance for the classifier, we average the F1 scores over all interaction classes leading to a final score in the range from 0 to 1. Averaging over classes as opposed to the total number of

frames (equivalent to the ratio of correct frames) assigns equal importance to all interaction classes and prevents the score from being dominated by the most-occurring interactions [72]. Hence it is better suited for datasets with interactions that occur with different frequencies.

To obtain a robust performance measurement, we apply a 5-fold cross-validation scheme. The folds correspond to the five videos in YR from which the clips have originally been selected. Creating the folds in this way prevents the contamination of the evaluation data with data from animals that are also contained in the training set. Hence, the performance is always measured on unseen animals. We report the mean and standard deviation of the F1 score across the folds.

In each of the five validation iterations, we perform another cross-validation with the goal to find the optimal classifier parameters. Among the four training videos, we repeatedly leave one video out and select the parameters with the highest average performance. We then retrain the classifier on all four training videos using the selected parameters.

5.4 Results

The comparison of the classifiers (Table 5.1) shows that all seven classifiers perform comparably on each dataset version. The difference between the worst and best classifier among each version is approximately two standard deviations whereas the difference between the worst and best tracking is approximately 4-5 standard deviations for every classifier. Given the range of classifiers tested, this emphasizes that feature quality, rather than the classifier, largely determines the performance in this experiment setup. We further see in Table 5.1 that fewer tracking errors lead to higher average accuracy. With each additional error eliminated, the score increases on average by 0.08.

The remaining experiments are conducted with the classifier that performs best on `YR-ID+Loc`: SVM-Lin. Although RF achieves comparable accuracy, we choose SVM-Lin over RF because it has fewer parameters to tune and was less sensitive in the model selection. In all training folds, the same value was selected as the optimal parameter ($C = 10$).

5.4.1 *Effect of Tracking Errors*

Looking at the F1 scores per interaction in Table 5.2 and the confusions in Figure 5.3, we see that not all interactions are affected by tracking errors in the same way. The accuracies are generally high for *solitary* actions and *allogrooming*, the two classes with the highest number of frames. The other

Table 5.1: The average per-class F1 scores achieved by the six classifiers on the three dataset versions with increasing degree of tracking quality. Sorted by performance on YR-ID+Loc.

	YR		YR-ID		YR-ID+Loc	
Classifier	mean	std	mean	std	mean	std
SVM-Lin	0.55	0.03	0.61	0.05	0.73	0.04
RF	0.58	0.04	0.68	0.04	0.73	0.05
SVM-RBF	0.59	0.04	0.65	0.05	0.71	0.04
QDC	0.54	0.06	0.61	0.04	0.70	0.03
LDA+kNN	0.52	0.04	0.61	0.03	0.69	0.03
GMM	0.52	0.05	0.61	0.05	0.67	0.04
LDC	0.50	0.04	0.55	0.03	0.66	0.04

contact-related interactions, *nape attacking*, *pinning* and *social nose contact* are not recognized well in the YR version but improve gradually as errors are corrected. The largest confusions occur among the contact-related interactions. Trajectory-related interactions on the other hand achieve comparably good accuracy even with the lowest tracking quality. Let us look at the improvements made through each correction step separately.

The correction of identity swaps (YR → YR-ID) leads to a major improvement of the recall of *social nose contact* by eliminating more than half of the confusions with *allogrooming* (48.4% → 19.8%). A number of small improvements across all classes lead to a higher average F1 score at both frame level (+0.07) and class level (+0.06).

Correcting the body point locations (YR-ID → YR-ID+Loc), and thereby also the orientation estimations, leads to considerable improvements. They are largest for the contact-related interactions where we see an increase in F1 score of 0.21 for *nape attacking*, 0.23 for *pinning* and 0.16 for *social nose contact*. Confusions among those classes are reduced but yet remain substantial with values between 11.4% (*pinning* → *nape attacking*) and 22.6% (*pinning* → *allogrooming*).

On the other hand, confusions between the trajectory-related interactions are largely eliminated. A few mistakes remain between *approaching* and *following* (5.4% and 7.0%) as well as between *moving away* and *solitary* (2.5% and 9.4%). These have a similar cause. *Approaching* often evolves into *following* but the transition is not clearly defined and may vary among annotators and every occurrence. We see the same effect for *moving away* where the transition to *solitary* is not always obvious. Misclassifications of this type cannot be resolved entirely as they are linked to the inter-

	Prec.	Recall	F1	#
Allogrooming	0.75	0.86	0.80	7480
Approaching	0.60	0.72	0.65	699
Following	0.67	0.73	0.70	1252
Moving away	0.51	0.66	0.58	641
Nape attacking	0.22	0.25	0.23	621
Pinning	0.37	0.27	0.31	1780
Social nose contact	0.44	0.22	0.30	2083
Solitary	0.96	0.92	0.94	4026
Avg. frames	0.69	0.71	0.69	18582
Avg. classes	0.57	0.58	0.56	8

(a) Per-class results: `YR`

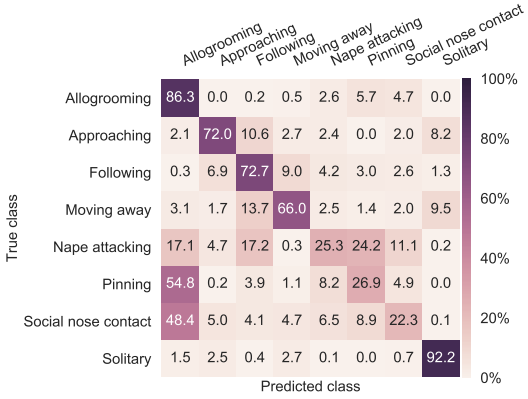
	Prec.	Recall	F1	#
Allogrooming	0.85	0.89	0.87	7523
Approaching	0.64	0.75	0.69	699
Following	0.69	0.66	0.67	1231
Moving away	0.53	0.75	0.63	641
Nape attacking	0.23	0.21	0.22	601
Pinning	0.49	0.31	0.38	1769
Social nose contact	0.55	0.62	0.58	1912
Solitary	0.97	0.92	0.94	4026
Avg. frames	0.76	0.77	0.76	18402
Avg. classes	0.62	0.64	0.62	8

(b) Per-class results: `YR-ID`

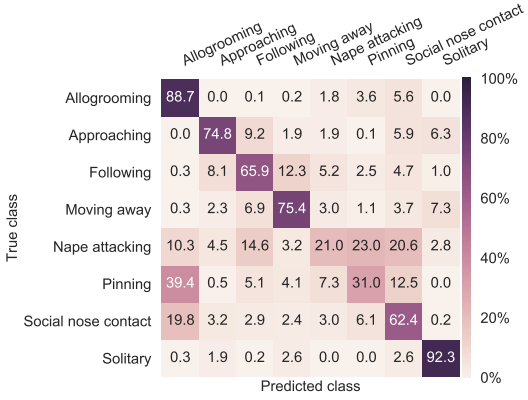
	Prec.	Recall	F1	#
Allogrooming	0.89	0.90	0.89	7560
Approaching	0.72	0.76	0.74	699
Following	0.81	0.80	0.81	1252
Moving away	0.72	0.80	0.76	641
Nape attacking	0.37	0.52	0.43	621
Pinning	0.64	0.59	0.61	1780
Social nose contact	0.75	0.72	0.74	2083
Solitary	0.97	0.92	0.94	4026
Avg. frames	0.83	0.83	0.83	18662
Avg. classes	0.73	0.75	0.74	8

(c) Per-class results: `YR-ID+Loc`

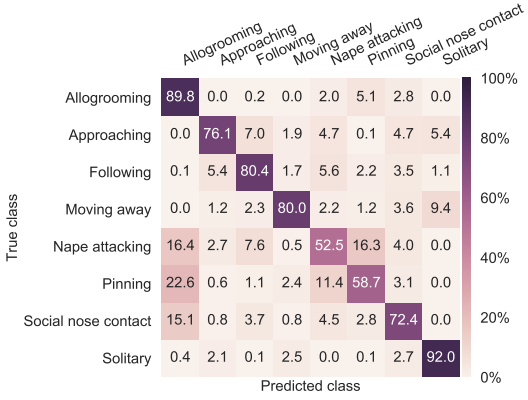
Table 5.2: Classification performance of SVM-Lin for different levels of tracking quality and `Full` feature set.



(a) Confusion matrix: `YR`



(b) Confusion matrix: `YR-ID`



(c) Confusion matrix: `YR-ID+Loc`

Figure 5.3: Confusion matrices for different levels of tracking quality.

annotator agreement: depending on the annotations with which the predictions are compared, the error may vary.

Overall, the F1 score increases by 0.07 when averaged over frames and by 0.12 when averaged over classes.

5.4.2 *Effect of Feature Set*

We now also include different feature sets in our analysis. We examine every combination between the three dataset versions (tracking quality) and the three feature sets. The F1 scores averaged over the evaluation folds are reported in Figure 5.4.

Considering the rows in the matrix, there is an upwards trend in performance with increasing tracking quality irrespective of the feature set. Again, this highlights the strong effect of tracking errors on the classification performance. Moreover, the performance also increases along the columns: richer feature sets improve the performance. The gain is negligible for the low quality tracking, but is substantial with the tracking errors eliminated. Clearly, adding more features only pays off in better performance if they are computed from reliable tracking data.

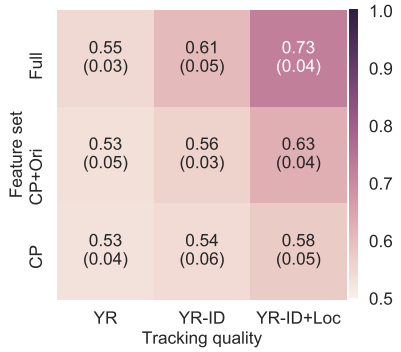


Figure 5.4: The average per-class F1 score achieved by SVM-Lin in all combinations of tracking quality and feature sets. In parentheses: standard deviation.

5.5 Cross-Quality Experiment

In the preceding experiments, we have trained the classifier and evaluated its performance using data with the same level of tracking quality. Good quality data has shown to be essential for accurate classification. Obtaining such high quality data currently still involves manual corrections and

is therefore time-consuming. Hence, an interesting question is whether a classifier trained with good quality data is also able to recognize interactions in data with lower tracking quality. This would allow us to train a classifier after a one-time correction effort that could then be used on novel data without the extra effort.

We train the SVM-Lin classifier with the corrected data from `YR-ID+Loc` and evaluate its performance on `YR` and `YR-ID`. In both cases we fail to achieve competitive performance. The average F1 score is 0.45 (0.06 std) when evaluated on `YR` and 0.59 (0.09 std) on `YR-ID`. Both scores are lower than in the corresponding same-quality experiments (`YR`: 0.55, `YR-ID`: 0.61). This suggests that we do not benefit in practice from corrected, clean features as long as we cannot guarantee that we can generate them without expensive, manual intervention.

5.6 Discussion

All in all, we have shown that eliminating tracking errors leads to better classification. This pattern occurred for all tested classifiers, which suggests that the effect is indeed inherent to the underlying data and not to the classifier.

In contrast, pose features that are derived from unreliable tracking data have no positive effect on the performance. Furthermore, tracking needs to be equally reliable both in training and for classification. In particular, manual correction of training data requires the same corrections to be performed for every experiment in practice. Only under the condition that reliable tracking is available can pose and orientation features enrich the representation of interactions and allow for more accurate classification.

Improvements from the corrections are most visible for contact-related interactions, which are characterized by distinct relative poses. Trajectory-related interactions, however, can be recognized using features derived from only a single body point. Nonetheless, the accuracy improves when the orientation is estimated from multiple body points instead of from the direction of motion as the former is more robust and also possible when the animal is immobile.

In our systematic analysis, we found that contact interactions are recognized less accurately than trajectory interactions. Despite the improved tracking and pose features, contact interactions remain partially ambiguous to the classifier. We conclude that the features that we derive from three body points are insufficient to facilitate the distinction. We are confident that it is possible to achieve better performance with a) more advanced tracking techniques addressing the challenging occlusion

situations and b) specialized features that capture the rat's body in more detail including the paws and fine-grained motion.

The results of this chapter motivate two choices that we make with respect to the investigations in the following chapters. First, given that the focus of this thesis is the learning of classification models with reduced manual effort, tracking and feature design lie beyond our scope. With this in mind, we will consider the classifier in the following chapters as a black box that may be replaced by a classification algorithm that uses any suitable set of features. Second, considering that we are currently not able to correctly distinguish many of the contact interactions, we will group these interactions in the following chapters into one common *contact* class. This allows us to concentrate on the learning challenges in absence of artifacts from insufficient features.

6

Variations in Rodent Social Behavior and the Implications for Cross-Dataset Classification

In the previous chapter we showed that the quality of the tracking data has a direct effect on the classification performance. We have also seen that tracking is not the only factor involved. The choice of features as well as ambiguity among different interactions both limit the classification accuracy.

Now we address another aspect that may influence the accuracy: the variation of how rodents perform interactions. These variations are partly natural, partly artificially induced, for example by stimulating certain behaviors of interest or by using only animals of a specific gender or genetic background. As illustrated in Figure 6.1, when we train a classifier, it learns a model for each interaction that reflects the variations in the training examples. But what if the classifier later encounters examples of a known interaction that do not fit the learned model? Is it still accurate, do we need to train a new classifier or can we adapt it to the new situation?

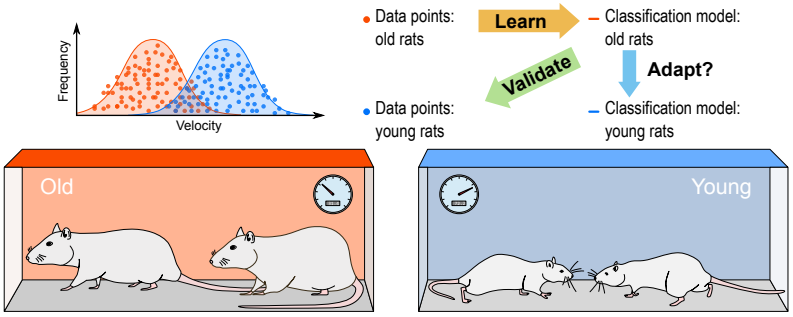


Figure 6.1: Example of a cross-dataset application with an old and a young rat population. Behavior variations may lead to different classification models for each population.

In this chapter, we examine the capability of two classifiers to generalize from the training examples to a) the inherent variations of the interactions and to b) examples from another dataset. We first discuss the different sources of variation and how they are typically handled by automated recognition methods in the literature (Section 6.1). In Section 6.2

we conduct cross-dataset classification experiments in which we focus on the difference between evaluating within and across datasets. To extent the classifier to the other dataset, we aim to remove the difference with a simple domain adaptation technique. In Section 6.3 we report the results which we then discuss in Section 6.4.

With the experiments in this chapter, we want to show that dataset variations are not only due to technical differences such as camera resolution but can also be induced by behavior itself. Those variations are potentially more difficult to incorporate into a classification method. Prior to the analysis, they may be hidden and thus prevent us from predicting their effects on the classification. This stands in contrast to external factors including camera resolution and cage size, which are known beforehand. Besides arguing for cross-dataset validation, we also suggest a potential approach to deal with variations, namely to adapt to the variations using techniques from *domain adaptation* [95].

6.1 Classification with Behavioral Variations

Generally, there are two sources of variation in the performance of an interaction. First, if two animals perform the same interaction multiple times in the course of a video, they will do so slightly differently every time. They may move at a different speed or start the interaction from a different location. We consider this the *natural variation* of an interaction.

Second, there is a *systematic bias* in the natural variation that depends on the tested animal population and the environment in which the animals are observed. The properties of the population and the environment can influence the way the interactions are performed. For example, genetic background, age and progress of a disease or its treatment are factors that can cause animals to move slower than animals from another population. Similarly, the environment, which is often created by the researcher to study and stimulate specific behaviors, is characterized by the available space and the presence of hiding places or novel objects. All of these may allow or prevent interactions to be performed in specific ways or be performed at all. Finally, the definition of the behavior categories could perhaps be considered a systematic bias as well. A different definition or its interpretation by the annotator can cause parts of interactions or entire occurrences to be labeled in one experiment and disregarded in another. In contrast to the other types, the category definition does not actually affect the animals' behavior but only its labeling. Hence we do not consider it as a systematic bias.

From an automated recognition method we expect that it can handle natural variations and that the classifier can generalize from the empirical training examples to those inherent variations. These are, in essence, the variations against which a classifier is evaluated. In the literature, the majority of automated recognition methods are trained and evaluated using data from one dataset, hence from one specific animal population observed in one specific environment [18, 42, 48, 55, 61, 73]. Consequently, training and evaluation data follow the same distribution with respect to the systematic bias of the behavior variations. The classifier learns only from variations that are due to natural, stochastic diversity.

In itself, having learned the natural variations is not a problem. In contrast, it is exactly what we want the classifier to do. If it succeeds, it will achieve a high accuracy and we would consider it a good classifier. The systematic bias only poses a problem if it differs compared to the training data, for instance, due to modifications to the animal population or the environment [122]. Let us illustrate the problem with an example.

In a longitudinal study conducted to investigate the social behavior of rats at different points in their life, the advanced age can cause the rats to move slower during *approaching* or *following*. This systematically biases the variations of these interactions toward lower velocities. If the classifier has been trained with data from younger, quicker rats, it may not recognize the slower interactions and misclassify them. Unlike in this example, a change in the population or the environment is not necessarily known. A change in systematic bias can be hidden from us, even when replicating a previous experiment. Precise replication is actually rather difficult [26, 146] and involves a range of environmental factors whose effects on the behavior are not well understood yet [133].

In the literature, the effect of systematic biases in rodent behavior datasets has received little attention. As stated above, the majority of the presented methods are developed, trained and eventually evaluated on a single dataset. This becomes critical when the trained classifier is applied in practice. Beyond the specific experiment setting that is included in the dataset, the evaluation is of limited value as it cannot predict the classifier's performance in another setting. Few works explicitly address different settings and include them in their datasets. van Dam *et al.* [145] use videos from different rat strains recorded in different cages and Jhuang *et al.* [59] vary the lighting conditions and mouse strains. Both use cross-validation to estimate the accuracy in an unseen setting by leaving one setting out of the training set and then evaluating on that held out data. The two works focus exclusively on *individual* rodent behavior. The social mouse behavior dataset CRIM13 includes mice of both genders and various treatments, but

the recognition method is not cross-validated with respect to the different populations [18].

It seems that including a range of variations in the dataset is the favorite approach to handle systematic bias. Creating such a dataset obviously comes at the cost of obtaining additional videos as well as manual annotation effort as the new data need to be labeled. In practice, this means there is a tradeoff between the effort required to create the dataset and the coverage of systematic variations. Besides this tradeoff, there is an increased interest in designing heterogeneous animal experiments that explicitly include environmental variations to enhance the robustness of behavior analysis [110, 111, 151]. As a result, we may see an even higher degree of diversity within and across experiments in the future. We therefore argue for an evaluation of automated recognition methods across settings and datasets. Only with cross-dataset evaluation can we be confident about the performance of the classifier in practice and judge to which settings we can apply it without retraining.

6.2 Cross-Dataset Experiments

We now move on to our cross-dataset experiments. Our goal is to quantify the effect of a systematic bias on the classification accuracy in a realistic scenario. We need two datasets that are sufficiently similar to allow for cross-classification but yet differ in one or more aspects that induce a systematic bias. The RatSI and YR datasets satisfy both premises. They contain the same set of interactions annotated based on the same definitions, are recorded in a similar environment with the same type of cage and lighting conditions, but differ in the age of the rats. The juvenile rats in YR are about five weeks old and engage frequently in fast-paced, playful interactions. The rats in RatSI are nine month old, larger, slower and their interactions are typically more gentle and mild.

We perform three experiments. First, we assess the baseline performance for each dataset separately by training and evaluating a classifier with data from the same dataset (*Within-dataset*). Second, we assess whether the classifier generalizes to other settings by evaluating its performance on the other dataset (*Cross-dataset*). Third, we aim to neutralize the age difference between the two datasets by scaling the distribution of the feature values (*Adaptation*).

In the following paragraphs, we explain the computation of the features and introduce two classification models that will be used in the experiments.

6.2.1 *Unifying Features Across Animals*

In the previous chapter we derived a set of twelve features for the YR dataset, which we denoted as `Full`. We will use this feature set with a few modifications.

Previously, we assumed we would know which animal plays the active role in the interaction, for example that rat A is *following* and rat B is *being followed*. It was hence straightforward to base the classification on the features of the active animal. Now we turn to the realistic scenario in which we do not have this knowledge. The classifier is then required to predict the correct interaction label irrespective of which of the two animals is the active one.

If both animals are potentially active, we need to consider the features of both animals for classification. The two-animal classification can be formulated in two ways. We can consider either animal as the active one and apply the classifier twice [48], thereby predicting two interaction labels for every video frame; or combine the two sets of features to one common feature vector and then classify a frame based on that vector. In the first approach, conflicts in the two predictions need to be resolved in post-processing, for example by prioritizing the interaction categories beforehand or based on the classifier's confidence in each label. Furthermore, it requires that during training of the classifier the active animal is known so as to omit the examples from the passive one. We choose the second approach and unify the features across animals because we consider an interaction being performed by two animals together and should therefore be inferred from their joint features. The drawback is that the classifier outputs only one label for both animals. If required, the role assignment needs to be computed in an extra post-processing step, for example, by assessing the relative positioning of the rats.

Given that the order of the animals is arbitrary, we cannot simply concatenate the two feature vectors. Instead, we need to aggregate them across animals to create a vector that is invariant to the order. For some features, such as the center-point distance, aggregation is not necessary because the values for both animals are the same. We call these *symmetric* features, as opposed to *asymmetric* features which take on different values for each animal, such as velocity. In order to aggregate the asymmetric features, we compute the mean and the absolute difference of the values of the two animals. In total, this yields 17 features computed for every video frame listed in Table 6.1.

To prevent that different tracking quality influences the cross-dataset performance, we use the `YR-ID` version of YR with corrected identities in

Table 6.1: Features used in classification experiments. Asymmetric features are aggregated over animals (superscript 1 and 2) by computing mean and absolute difference.

Symmetric	Asymmetric	
d_{cc}	mean(d^1_{nt}, d^2_{nt})	diff(d^1_{nt}, d^2_{nt})
d'_{cc}	mean(d'^1_{nt}, d'^2_{nt})	diff(d'^1_{nt}, d'^2_{nt})
d_{nn}	mean(v^1_c, v^2_c)	diff(v^1_c, v^2_c)
d'_{nn}	mean(v^1_n, v^2_n)	diff(v^1_n, v^2_n)
φ	mean(cos(γ^1), cos(γ^2))	diff(cos(γ^1), cos(γ^2))
	mean($\boldsymbol{r}_{\text{spine,0}}$, $\boldsymbol{r}_{\text{spine,1}}$)	diff($\boldsymbol{r}_{\text{spine,0}}$, $\boldsymbol{r}_{\text{spine,1}}$)

the tracking data. The tracking quality of this version is comparable to the quality in RatSI. Furthermore, as we have found in Chapter 5, we cannot reliably distinguish the contact-related interactions using trajectory-based features. Therefore, we merge the annotations of the four contact interactions *allogrooming*, *nape attacking*, *pinning* and *social nose contact* into one overall *contact* class.

6.2.2 *Measuring Cross-dataset Performance*

We conduct each of the three classification experiments (*within-dataset*, *cross-dataset* and *adaptation*) in two directions: training on RatSI with evaluation on YR, and training on YR with evaluation on RatSI. We further compare two classification methods and thus perform in total twelve experiments.

We use a Support Vector Machine with linear kernel (SVM-Lin) as we have done previously in Chapter 5 where it yielded the best classification performance on the YR dataset. For a deeper analysis and comparison of the models trained on different datasets, we additionally use a Gaussian Mixture Model (GMM). The GMM is a generative classification model that allows us to inspect and compare the mean values and covariances of the trained models.

Both classifiers have free parameters for which we need to determine the optimal values. We select the optimal model parameters by cross-validation. The free parameter of SVM-Lin is the cost function coefficient C, which regularizes the resulting model by indirectly controlling the number of support vectors. The GMM classifier consists of one mixture model per interaction. Every model is a mixture of multiple, multivariate Gaussian components. During training, the optimal mixture is determined by the Expectation Maximization algorithm [35] which iteratively

searches for the mixture that best explains the distribution of training examples of a given interaction. The parameters that need to be set beforehand are the number of Gaussian components per model, whether or not to restrict the shape of each component to be ellipsoidal (diagonal covariance matrices), and a regularization parameter that is added to the covariance matrices to artificially increase their variance. The latter can be beneficial for preventing overfitting in small training sets.

The performance is measured as in previous chapters by the F1 score of each interaction class. To obtain a single measure of performance for the classifier, we average the F1 scores over all classes leading to a final score between 0 and 1.

6.2.2.1 *Within-dataset Evaluation*

We do the within-dataset evaluation using a cross-validation scheme. That is, we split the dataset into k parts and then train the classifier on $k-1$ parts and measure its performance on the remaining part. This is repeated such that the performance is evaluated on every part once. For RatSI we set $k=3$ (three videos per part) and for YR $k=5$ (one video per part). Because YR is smaller than RatSI, we split it into more parts so as to maximize the amount of training data at the cost of more repetitions. In each repetition, the optimal model parameters are automatically determined by another cross-validation procedure applied to the videos of the training part. For the model selection we use four training videos of RatSI (two test videos and three repetitions), and three training videos of YR (one test video and four repetitions). Once the optimal parameters are found, the classifier is trained on all training parts.

6.2.2.2 *Cross-dataset Evaluation*

For the cross-dataset evaluation, we only need to measure the performance on the evaluation dataset. Hence we can use all videos of the training dataset for determining the optimal classifier parameters. On RatSI, the parameters are found among $k=3$ parts, on YR among $k=5$ parts. The best performing settings are then used to train the classifier on the entire dataset. Note that the classifiers evaluated in the cross-dataset setting are trained on more videos than in the within-dataset setting. This may give them an advantage as they encounter more natural behavior variations.

6.2.2.3 *Evaluation after Feature Scaling (Adaptation)*

To examine whether the age difference can be removed from the feature values, we employ a simple technique that scales the values of each fea-

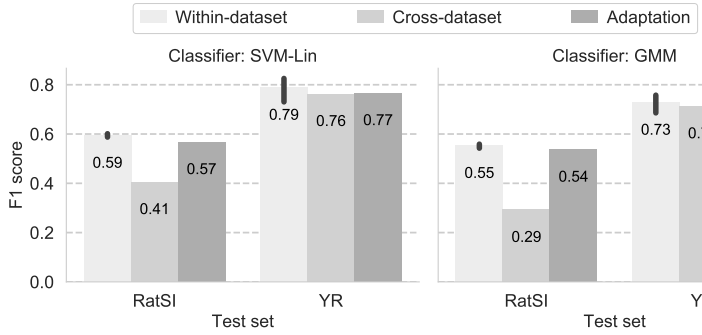


Figure 6.2: Recognition performance (F1 score averaged over classes) with standard error for cross-validated *within-dataset* evaluation.

ture such that the fifth-percentile value is -1 and the 95th-percentile value is 1. Using the percentiles instead of the minimum and maximum values increases the tolerance against outliers. We intentionally avoid scaling to unit-variance as the scaling can be influenced by the skewed class priors. After independently scaling the training and validation sets, we repeat the cross-dataset evaluation.

6.3 Results

The performance of the classifiers in the twelve experiments is reported in Figure 6.2. In the within-dataset experiment, SVM-Lin achieves a F1 score of 0.59 (0.01 sd.) on RatSI and 0.79 (0.06 sd.) on YR. GMM scores slightly lower but yet within the same order: 0.55 (0.01 sd.) on RatSI and 0.73 (0.04 sd.) on YR. We first notice that the performance on RatSI is substantially lower than on YR despite having more training examples which emphasizes the difficulty of annotating the RatSI videos. The overall performance on YR is higher than in Chapter 5 (SVM-Lin: 0.61, GMM: 0.61) as we have merged the contact interactions and thereby simplified the task.

When training on YR and evaluating on RatSI, the performance of both classifiers drops considerably by 31.7% to 0.41 for SVM-Lin and by 47.1% to 0.29 for GMM. In contrast, the performance is mostly maintained in reversed training direction (RatSI → YR). After adapting the features, the classifiers regain their performance with 0.57 and 0.54, respectively.

Let us delve deeper into the results to find the reasons for the performance decline in the cross-dataset setting. In the per-class results in Table 6.2, we see that most of the decline is caused by a failure to recognize *contact* and, in case of GMM, *approaching*. If we look at the models that

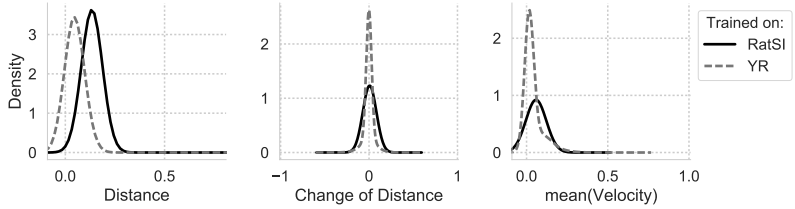
Table 6.2: Per interaction recognition performance for within-dataset (w), cross-dataset (c) and adaptation (a) experiments.

	SVM-Lin						GMM					
	RatSI			YR			RatSI			YR		
	w	c	a	w	c	a	w	c	a	w	c	a
Approaching	0.52	0.39	0.45	0.71	0.65	0.64	0.44	0.00	0.40	0.69	0.54	0.62
Contact	0.71	0.00	0.66	0.97	0.97	0.96	0.67	0.00	0.65	0.94	0.96	0.96
Following	0.58	0.53	0.56	0.77	0.75	0.72	0.51	0.52	0.54	0.68	0.67	0.58
Moving away	0.27	0.28	0.29	0.61	0.51	0.57	0.29	0.09	0.24	0.48	0.55	0.58
Solitary	0.89	0.83	0.88	0.88	0.93	0.94	0.85	0.84	0.85	0.85	0.85	0.91

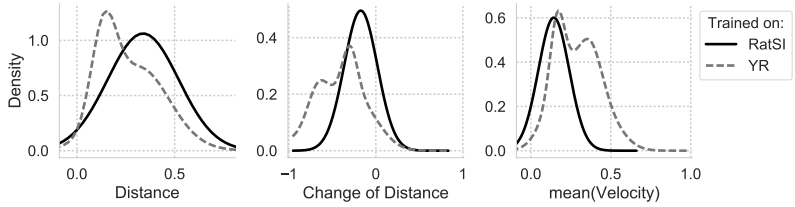
the GMM learned for these two interactions, we find distinct differences that presumably are responsible for many misclassifications. In Figure 6.3 we plot the marginal probability density functions of the *contact* and *approaching* models being trained on either RatSI or YR. We concentrate on three features that exhibit considerable differences between the datasets: the center point distance, the change of that distance over time, and the mean velocity.

The classifier has modeled the *contact* interaction with a smaller distance given samples from YR than from RatSI. Furthermore, the peak of the density in RatSI coincides with the peak in YR for *approaching*. Such a modeling conflict is likely to result in numerous misclassifications of *contact* frames. Besides the smaller distance values, the rats in YR also appear to run faster. The mean velocity of *approaching* extends further to higher velocities in the YR model compared to the RatSI model.

As for the reason why the performance is maintained if the classifier is trained on RatSI, it appears that the RatSI models have generally larger variances. The larger variance acts as regularization in favor for this cross-dataset classification. For example, the mean velocity of the *contact* interaction is modeled by a narrower distribution in YR causing samples just outside the captured values to be potentially misclassified. A larger variance can lead to a more gentle decision function. What causes the RatSI models to have larger variances? The main reason is that RatSI has more content than YR: 135 min compared to 12.6 min. RatSI comprises more examples of every interaction and therefore covers a larger variety of feature values.



(a) Interaction: *Contact*



(b) Interaction: *Approaching*

Figure 6.3: Marginal probability density functions of the Gaussian models learned from RatSI and YR.

6.4 Discussion

Despite that RatSI clearly poses the more challenging classification problem, we show that it is a suitable dataset for training interaction classifiers. While the classifiers trained on RatSI generalize well to YR, training on YR is not optimal as is evident from the declined performance on RatSI. This result demonstrates the necessity to validate classifiers on other datasets as their capabilities to generalize cannot easily predicted from a single-dataset evaluation.

It is a promising result that we were able to compensate for the age difference between the animals and restore the reduced accuracy with a simple scaling operation. It shows that classifiers are not necessarily bound to one experiment setting and highlights the potential for cross-dataset applications. Clearly, scaling features to a common range is not likely to perform well under more complicated behavior variations. More advanced adaptation methods need to disentangle the different influences of induced and natural variations as well as the video acquisition.

Because dealing with behavior variations by adaptation presents a novel approach for rodent behavior recognition, there are several open questions. For example, it is yet unclear whether novel examples need to be labeled for the adaptation and its validation. If so, when does adaptation become less

efficient than training a new classifier? On the positive side, developing adaptation methods may not only enable automated behavior recognition in longitudinal studies of diseases, it will also enhance our understanding of rodent interactions along the way. This will bring us a step closer to a unified recognition framework that is applicable to a broad range of experiments, animals and environments.

On a conceptual level, adaptation as performed in this chapter is restricted to scenarios in which the same set of behaviors is of interest, their definitions are identical and the video acquisition is similar. If we instead want to measure novel behaviors or acquire videos, for example, from the side view as opposed to the top view perspective, then previous classifiers may be of limited use. In this case, manual annotation is the only option. The topic of the next chapter is the reduction of that manual effort using the combined strengths of human and machine.

7

Interactive Annotation

In the previous chapter, we demonstrated that behavior variations caused by different experiment settings can decrease the accuracy of social interaction classifiers. Although the accuracy can perhaps be restored by adapting the classifier to the variations, there are scenarios in which no classifier is available or adaptation is not a possibility. For instance, a researcher may wish to analyze previous data in new light and modify behavior definitions or add new behaviors to the repertoire. Similarly, the behavior variations across experiments could be so complex that adaptation is no longer a viable, efficient option. The researcher is left with manual annotation.

Our goal in this chapter is to reduce the manual effort in these scenarios. While the researcher annotates the videos, a new classifier is trained from the examples she or he already labeled. As soon as enough examples have been labeled, the classifier can annotate the remaining, yet unlabeled videos automatically. This saves time because manual annotation can be stopped as soon as the classifier is sufficiently accurate.

Initially, it is unknown how many labeled examples are needed to achieve a satisfying performance. Consequently, we need to retrain the classifier repeatedly while more labeled examples are added. Alternating between labeling examples and training the classifier is an iterative learning process that can be implemented with a *human in the loop*. This active learning paradigm has the potential to significantly reduce the labeling effort [125].

It may be possible to reduce the effort even further by directing the human annotator to label those examples that carry the most information for training the classifier [84]. Examples that are similar to already labeled examples are less informative, thus less important than more distinctive or rare examples [58]. By considering only the most informative examples, we could train a classifier that is as accurate as a classifier trained on all examples, but with reduced labeling effort.

Such an interactive framework has three main components: selecting examples, labeling them, and training the classifier. These are the standard components in active learning applications [7, 152]. In this chapter we implement and apply them for the first time to annotate rodent social behavior. We experimentally analyze the performance and convergence properties of different selection algorithms and labeling strategies. To find

the parameters that allow us to optimally learn from rodent behavior, we first perform a series of *offline* experiments. In these offline experiments we replace the human annotator by a dataset oracle that labels examples from previously obtained annotations. Using the oracle instead of a human allows us to test a large number of parameter settings in a short amount of time. After we have determined suitable parameters, we validate the choice in a user study with human annotators. The study demonstrates the efficacy of our tool. As in Chapter 6 we cross-validate the results using the YR dataset to assure that the framework not only produces annotations for one dataset but also trains more general rodent behavior classifiers.

The chapter is structured as follows. We first introduce the annotation framework and its components in Section 7.1. We then apply the framework to rodent social behavior from the RatSI and CRIM13 datasets and experiment with different parameters in Section 7.2. In Section 7.3 we evaluate the framework in a user study. Possible extensions in learning and labeling geared toward larger datasets are tested in Section 7.4. We discuss the results in Section 7.5.

7.1 Interactive Behavior Annotation Framework

We now introduce our interactive annotation framework for videos. Recall that we denote the domain of the feature vectors by $\mathcal{X}$ and the set of the K behavior categories by $\mathcal{Y}$. The feature vector of a frame i is then $x_i \in \mathcal{X}$ and the corresponding label $y_i \in \mathcal{Y}$. We consider every frame a potential sample, although for labeling we include surrounding frames from the video to enrich the sample with context information. The dataset to be annotated has in total n video frames. RatSI for instance comprises about 202 500 frames and CRIM13 more than 2.5 million.

The framework consists of three main components as depicted in Figure 7.1. There is a pool of $n - m$ unlabeled samples, $\mathcal{U} = \{x_{m+1}, \ldots, x_n\}$. Initially, the entire dataset is unlabeled and $m = 0$. We do not impose any specific order on the samples in the pool but we keep references to the original video frames to be able to retrieve surrounding frames as context. From $\mathcal{U}$ a sample x_i, $m < i \leq n$, is selected using sampling strategy s. The sample is then presented to an oracle, usually a human expert, although for the offline experiments we use a data oracle instead. Depending on the labeling strategy q, the oracle provides label response y_i. The now labeled sample (x_i, y_i) is moved from $\mathcal{U}$ to the pool of labeled samples $\mathcal{L}$. From $\mathcal{L}$ the classification model $f(x)$ is learned. These steps are performed repeatedly either for a certain number of iterations or until the labeling is stopped

manually. We will now briefly describe the three main components of the framework.

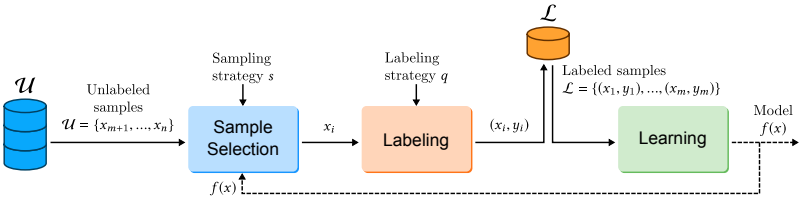


Figure 7.1: Framework components. See text for details.

7.1.1 *Sample Selection*

The sampling strategy $s(\mathcal{U})$ determines which sample from the unlabeled pool is selected. If the strategy does not consider any labeling information, it is prone to select samples that are similar to previous samples. Such samples might contain redundant information and are therefore less informative for the learning algorithm. Furthermore, behaviors that occur with a low frequency are less likely to be selected. Consequently, the trained classifier might be biased towards predicting the more frequent behaviors.

To make better informed decisions, the sampling strategy can be guided by the current classification model. The strategy then becomes a function of the unlabeled samples given the classifier: $s_f(\mathcal{U})$. The sampler can choose to exploit the classifier's predictions of the unlabeled samples as well as its confidence in these predictions [82, 83]. It can thus avoid selecting redundant examples and aim to balance the selection among all behavior categories. We experiment with both random and informed sampling in Section 7.2.4.

7.1.2 *Labeling*

Once a sample has been selected, we query its label. To minimize the overall annotation time, it is desirable to make the labeling efficient for the user. The response time, that is the time the user needs to label a sample, depends on how evident the behavior in the sample is, the number of labeling options to choose from as well as the interaction with the user interface. Also the amount of context that is shown, that is the frames preceding and succeeding the selected sample, may affect the response time.

Although the labeling strategy does not influence the content of the sample, it does determine the number of labeling options. For instance, it can

ask the oracle to select the true label for the presented sample out of all K possible labels. It can also suggest one category and ask whether or not the sample belongs to that category. With this binary query the number of options decreases from K categories to two (yes or no), which could potentially reduce the response time.

Furthermore, the response time could be reduced by making it easier to find the desired labeling option. This could be particularly useful if the number of options is large. A possible strategy is to sort the labels according to the classifiers certainty. Under the assumption that this moves the most likely answers to the top of the list, the user should be able to find the desired answer within the first few options. We evaluate the 1-of-K and binary labeling strategies using a data oracle in Section 7.2, and the fixed-order and ranked user interface strategies with human annotators in Section 7.4.2.

7.1.3 *Learning*

The task of the learning algorithm is to train the classification model with the samples that have been labeled so far. Although we treat both learning algorithm and classifier as black-boxes in this chapter, we briefly discuss some general implications of their properties. Given the iterative nature of the framework, the classification model needs to be retrained in regular intervals. To avoid long waiting times, it is desirable to use an efficient learning algorithm that scales favorably with the number of training samples. Besides being efficient, the classifier should be capable of providing an estimate of the confidence in its predictions to be able to inform the sample selection.

Depending on the complexity of the classification problem, we may avoid retraining the model with all samples in every iteration and instead use an incremental, online learning algorithm. An online learning algorithm is able to update a previous model using new training examples. Without making assumptions about the classification task or the data distribution, it is often impossible to guarantee that such an algorithm converges to the optimal solution. In Section 7.4 we examine whether the speed-up offered by Stochastic Gradient Descent [155], a popular algorithm for large-scale learning, involves a trade-off in accuracy.

Specifically for the evaluation of different labeling strategies, we pose an additional requirement on the learning algorithm. When labeling binary queries, a negative response only contains information about one class (the negatively labeled one) while for the other classes the sample remains unlabeled. The learning algorithm must be able to utilize the lim-

ited amount of information in such partially labeled samples. One way of achieving this is to split the multi-class classification problem into multiple one-versus-all problems. This allows for including a partially labeled sample in the corresponding sub-problem.

7.2 Active Learning for Rat Social Behavior

We now apply the proposed annotation framework to rat social behavior using the RatSI dataset. Our goal is to determine the optimal settings of our framework, in particular the following parameters: sample selection with respect to class balance and uncertainty level, and the presentation of the labeling options. We implement a range of sampling and labeling strategies and analyze their effect on the performance. We perform the analyses in offline experiments using a data oracle instead of a human annotator. The oracle responds with labels from the already annotated experiment dataset. This experimental setting enables us to conduct a large number of experiments in a short amount of time. Clearly, there is no guarantee that the determined settings are also effective in practice when a human annotator performs the labeling. Therefore, we will validate our choices in a user study in Section 7.3.

7.2.1 *Evaluating the Learning Performance*

We conduct a series of learning experiments in which we evaluate three sampling and two labeling strategies, and different parameter settings. To analyze the influence of each setting on the learning performance, we fix as many components of the experiment setup as possible.

First, we establish a validation set that is used to measure the classification performance. The validation set is separate from the experiment data set to ensure that the classifier has indeed generalized beyond the training examples. The validation set consists of two videos from the RatSI dataset that were chosen such that all behaviors occur sufficiently often to obtain a reliable performance measure. In practice, a validation set is typically not available. We only use it here to evaluate our framework. The experiment set contains the remaining seven RatSI videos and forms the initially unlabeled pool $\mathcal{U}$.

We compute the same feature set as in Section 6.2.1. The feature vector has 17 elements. As before we normalize each feature dimension such that the 5^{th} percentile has the value -1 and the 95^{th} percentile the value 1. Only the data from the experiment set are used for determining the normalization parameters.

We initialize the framework with one labeled sample per behavior and train the initial classifier. Although not strictly necessary, the initialization prevents several iterations of random sampling before a reasonably effective classifier can be trained. This effect is more pronounced when class distributions are unbalanced. In our experimental setup, this choice ensures that all experiments have the same starting point. The initialization samples are the mid-frames of randomly chosen interactions to avoid initialization with an ambiguous transition. We believe that finding one example per class by scrolling through the videos is a feasible task for the human annotator in practice.

We fix the number of queries per learning experiment to 400 as these yielded sufficient examples for convergence in previous experiments. To limit the duration per experiment, we issue the queries in batches. After a batch has been labeled, the model is retrained and its performance is measured. We experiment with different batch sizes in Section 7.2.3.1. Because the learning framework may include stochastic sampling decisions, we repeat every experiment ten times using the same settings. We report the means and standard deviations for all metrics.

As in previous chapters, we present the classification performance as the F1 score averaged over classes. We are also interested in how the performance evolves as more training examples become available. This gives us the option to detect when classifiers converge. We report the performance over time in learning curves that plot the averaged F1 score against the number of queried samples. To give an objective measure for comparing learning curves, we compute the area under the learning curve (AUC). The AUC combines the performance at the end of a learning experiment with the number of examples the classifier needed to reach that performance. Intuitively, we can interpret the AUC as a measure for the efficiency of the framework in terms of training examples. We report the area divided by the number of iterations to obtain a score in the range [0, 1]. A score of 1 indicates perfect learning performance where after only one iteration the classifier is able to label all test examples correctly.

7.2.2 *Querying the Oracle*

Because we are working with videos, a sample corresponds to a video frame. However, from a single frame the human annotator will not be able to reliably determine the performed action. Hence, we display a video clip surrounding the selected frame, which raises the question of a suitable clip duration. Short clips may not contain enough information while longer clips have a higher chance of containing more than one interaction which

cannot be annotated with a single label. In pilot experiments, we found that a duration of one second is a suitable trade-off. The optimal duration may vary for other types of behavior. A video clip is constructed such that the selected frame occurs halfway through the clip. The response of the user is then assigned to all frames of the clip.

In the offline experiments, the data oracle determines its response by the majority vote over all labels in the queried clip. To prevent that a clip which contains multiple interactions is labeled by only one label, we require that the majority vote covers at least 30% of the clip. Otherwise the data oracle rejects the clip and returns the label "Uncertain". A rejected clip accounts for one labeling iteration but the framework does not learn from rejected samples. A rejected clip cannot be queried again.

7.2.3 *Linear Classification Model*

To classify the rodent interactions we use a log-linear classification model. We achieve multi-class classification by training multiple binary classifiers in a one-versus-all scheme similar to related work [42, 61, 112]. Each classifier distinguishes one class from all other classes.

Each of the $K = |\mathcal{Y}|$ classifiers is determined by coefficients w_k and b_k with $k = 1...K$:

$$f_k(x) = w_k^\top x + b_k. \tag{7.1}$$

The result of $f_k(x_i)$ is positive if the sample x_i belongs to the positive class according to the classifier, and negative if it belongs to any other class. To assign the class label $\hat{y}_i \in \mathcal{Y}$, we evaluate all models and decide for the class with the highest positive output:

$$\hat{y}_i = \operatorname*{argmax}_k f_k(x_i). \tag{7.2}$$

7.2.3.1 *Training*

The K binary classifiers are trained independently of each other. Here we describe the general training procedure for one classifier $f(x)$ with binary labels $y \in \{-1, 1\}$. We determine the optimal model parameters w and b by minimizing the regularized training error

$$\min_{w,b} C \sum_{i=1}^{m} L(y_i f(x_i)) + \frac{1}{2} w^\top w \tag{7.3}$$

over the m examples in the training set $\mathcal{L} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. The loss function $L : z \mapsto [0, \infty) \in \mathbb{R}$ and the regularization term $w^\top w$ compete

for the conflicting goals of low classification error and low model complexity, respectively. This trade-off is controlled by the free parameter C. The loss function implies a cost for predicting $f(x_i)$ when the true label is y_i. We solve Equation 7.3 using the LIBLINEAR library which uses Coordinate Descent for finding the minimum [43].

For our classification task, training the model is an efficient operation that takes less than one second with a standard 2.4 GHz CPU. That allows us to simply learn a new model in every training iteration instead of updating the previous model. As learning more complex classification models may take substantially longer, it may be preferred to update the model in those scenarios. Performing an effective and efficient update is a challenging task that is the research topic of *online learning* [14, 21, 77].

In the case of skewed class distributions, the majority classes dominate the training error which can lead to inaccurate classifications of minority classes. To counter the imbalance, we use a class weight in the learning algorithm as introduced previously [154]. This weight assigns a relative higher importance to samples of smaller classes and is multiplied with the sample's loss. The weight is computed by $c_k = m/(K \cdot l(k))$, where $l(k)$ is the number of samples of class k in the labeled training set $\mathcal{L}$.

In order to enable the sample selection to utilize confidence information from the classifier, we require confidence scores for unlabeled samples. We can compute such scores from the classifier output if we use the logistic loss function

$$L\left(y_i f(x_i)\right) = \log\left(1 + \exp\left(-y_i f(x_i)\right)\right) \tag{7.4}$$

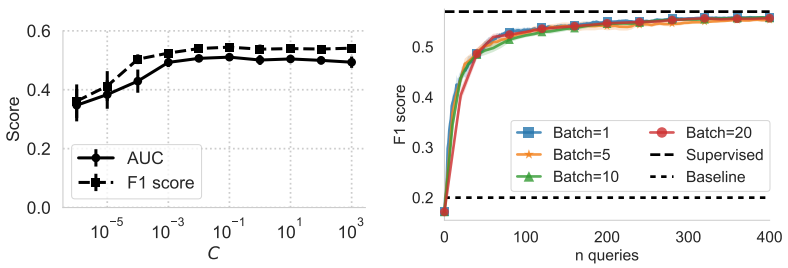
in the minimization problem in Equation 7.3. With this loss function we effectively train a logistic regression model that allows us to interpret the model outputs probabilistically. First, we estimate the posterior probability that the binary classifier correctly classifies sample x_i as positive ($y_i = 1$) using the sigmoid function:

$$p(y_i = 1|x_i) = \sigma(f(x_i)) = \frac{1}{1 + \exp\left(-f(x_i)\right)}. \tag{7.5}$$

Then, the posterior probabilities from all K classifiers are normalized to a confidence score $\tilde{p}(y|x)$ whose sum over all classes is one. We will later use this confidence score to find potentially informative examples that are queried for labeling.

7.2.3.2 *Parameter Search*

Before we move on to investigate different sampling and learning strategies, we first determine a suitable value for the regularization parameter



(a) Scores for regularization parameter C

(b) Learning curves for query batch size

Figure 7.2: Performance with respect to framework parameters.

C and the query batch size. We search for C in the range from 10^{-6} to 10^{3}. Figure 7.2a shows the area under the learning curves and the average F1 score after the last learning iteration. $C = 0.1$ achieves the highest performance with AUC = 0.51 (0.01) and F1 = 0.54 (0.01). Generally, high values of C and thus low regularization lead to better performance. We fix $C = 0.1$ for all remaining experiments.

Different query batch sizes have only a marginal effect on learning performance as observed in Figure 7.2b. The learning rate tends to be slightly lower in the beginning for larger batch sizes, but all classifiers eventually converge to the same classification performance. With a lower batch size, the classification model is updated more frequently which potentially leads to better sampling and learning decisions at the cost of higher computational demand. We choose a batch size of 10 as a trade-off between learning rate and the number of retraining operations.

For reference, we show the performance of a classifier trained with the entire labeled experiment dataset (supervised, F1 score = 0.57) and the baseline classifier that makes random classification decisions (baseline, F1 score = 0.2). The supervised F1 score is slightly lower than in the previous chapter (0.59) as we use two fixed validation videos instead of cross-validation, which influences the exact evaluation score.

7.2.4 *Sample Selection*

We now address the sampling from the unlabeled pool $\mathcal{U}$. We first assume that we have no knowledge about the properties of $\mathcal{U}$, such as the label distribution. A possible approach is then to simply select the first sample in $\mathcal{U}$, then the second, and so forth. However, we do know that $\mathcal{U}$ is created from videos and that the labels of consecutive samples are thus correlated.

To spread the selection across behaviors, events and videos, we can therefore better select *random* samples.

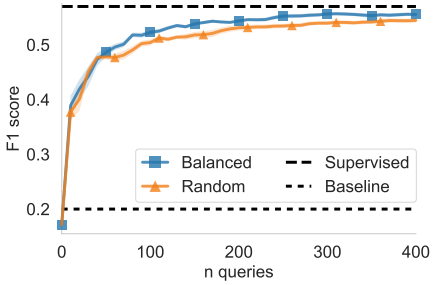
7.2.4.1 *Exploiting Class Priors*

The random strategy is unaware of the labels that the selected samples may have and also does not attempt to predict what it may be. Given the label imbalance in the dataset, it is thus prone to filling the labeled pool $\mathcal{L}$ with samples from the majority classes and neglecting smaller classes. While an unbalanced training set can lead to biased classifications, the underrepresentation of minority classes can also hinder learning accurate models of these classes especially in the first iterations. Intuitively, a better sampling strategy would aim to balance its selection among all classes. Using the current classification model, such a strategy predicts the labels of the unlabeled samples and makes its selection accordingly. We implement this balanced strategy such that it selects two samples of each class for every batch of ten queries.

To confirm our intuition, we compare the learning performance of the random and the balanced strategies in Figure 7.3. After 400 iterations the classifiers have converged to an accuracy that is close to the supervised classifier but using only 6.4% of the samples in the experiment set. This demonstrates the redundancy of many samples in the dataset. Comparing the performance of the two sampling strategies, we find that balanced leads to better classification accuracy after the 100^{th} iteration. This confirms our intuition that the random strategy does not sample sufficiently from the smaller classes. If we were to proceed with the labeling, the two strategies would eventually converge to the same performance as also the random strategy will encounter rare examples. Once the training set includes sufficient examples from all classes, the learner would not benefit from the more balanced set.

7.2.4.2 *Exploiting Classifier Confidence*

The balanced strategy randomly selects the samples within the targeted class. It is therefore still prone to select samples that are similar to previously labeled ones. If we want to select more informative samples, we first need to define what constitutes the expected information of a sample. As criterion for informativeness we utilize the classifiers' confidence in the prediction of a label. The intuition is that the sample that the classifier is least certain about is the most informative to learn from. Hence, a straight-



(a) Learning curves

	AUC		F1	
	mean	std	mean	std
Balanced	0.53	0.01	0.56	0.01
Random	0.51	0.01	0.54	0.01

(b) Statistics

Figure 7.3: Performance of random and balanced sample selection strategies.

forward approach is to select the sample x^* with the lowest confidence in the highest scoring label $\hat{y}$ [82]:

$$x^* = \underset{x}{\operatorname{argmin}}\, \tilde{p}(\hat{y}|x). \tag{7.6}$$

Recall that the confidence $\tilde{p}(\hat{y}|x)$ is a real-valued number in the range $[0, 1]$ with 0 being least confident and 1 being maximally confident. The sum of the confidence values for all possible labels of one sample is 1.

Selecting the least confident sample appears to be a good choice for classification tasks with clear and unambiguous data [124]. Rodent behavior however is often ambiguous [136]. In particular the transitions between interactions are inherently difficult to label consistently. Querying according to the lowest confidence is likely to result in many clips showing interaction transitions and ambiguous behavior. We therefore generalize the selection criteria in Equation 7.6 to select the sample that is closest to an arbitrary confidence level $\mathrm{CL} \in [0, 1]^{\mathbb{R}}$:

$$x^* = \underset{x}{\operatorname{argmin}} \left|\mathrm{CL} - \tilde{p}(\hat{y}|x)\right|. \tag{7.7}$$

We further extend the confidence-based strategy by reintroducing some explorative abilities. Given a classification model and a pool $\mathcal{U}$, the selection criteria in Equation 7.7 is deterministic. By converting the selection to a *probabilistic* sampling, we allow for more randomness. We assign each sample $x \in \mathcal{U}$ a weight v,

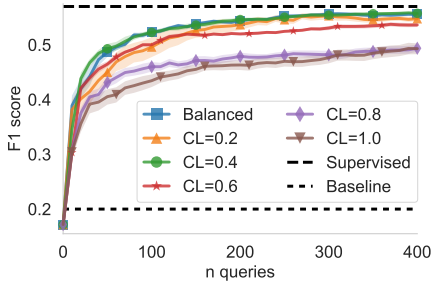
$$v(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\mathrm{CL} - \tilde{p}(\hat{y}|x))^2}{2\sigma^2}\right), \tag{7.8}$$

and then draw a sample $x^* \in \mathcal{U}$ with probability proportional to the assigned weight. For our experiments we set $\sigma = 0.025$ creating a narrow,

rather conservative window that allows for some randomization among samples but prefers confidence values close to the desired level.

7.2.4.3 *Results*

To find the optimal confidence level, we examine how the levels compare in terms of learning performance. The results in Figure 7.4 show that CL=0.4 leads to the best learning performance. A confidence level of 0.2, equivalent to highly uncertain samples, yields lower performance. The performance also decreases for higher confidence levels because they lead to typically less informative samples. Although these are plausible effects on the performance, the overall gain of using confidence-based sampling over the balanced strategy is limited.



(a) Learning curves

	AUC		F1	
	mean	std	mean	std
Balanced	0.53	0.01	0.56	0.01
CL=0.2	0.51	0.03	0.55	0.01
CL=0.4	0.53	0.01	0.56	0.01
CL=0.6	0.50	0.01	0.54	0.01
CL=0.8	0.46	0.02	0.49	0.02
CL=1.0	0.45	0.03	0.49	0.03

(b) Statistics

Figure 7.4: Sample selection with different confidence levels (CL).

Let us briefly investigate how the levels influence the sample selection. Our intuition is that a higher confidence level leads to less informative, more redundant samples. We measure the redundancy by the ratio of queries for which the desired target class matches the label given by the oracle. Figure 7.5 shows the measure for the different confidence levels as well as the balanced strategy for reference. The results support our intuition: samples with higher confidence values belong more often to the predicted class, while samples with lower confidence values are less often what the classifier predicts. Notably, below about 0.8 the confidence level is a reasonably accurate predictor as to how often the classifier is correct. For CL = 0.6, 65% of the queries match the prediction, for CL = 0.4 it is 49% and for CL = 0.2 29%. This confirms that the confidence score generated by the classifier is a reliable measure for its uncertainty. A high confidence however does not guarantee correct prediction.

Naturally, labeling a sample that is already predicted correctly will create less information than a sample that is currently misclassified. However,

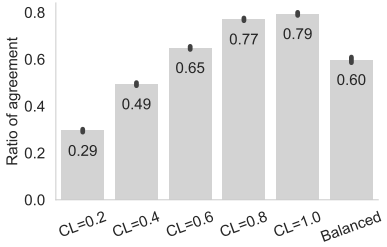


Figure 7.5: Agreement between target class and label response for different confidence levels (CL).

selecting low-confidence samples conflicts with the goal to balance the selection equally among all classes. Therefore, choosing a confidence level involves a tradeoff between a balanced but more redundant selection on one side and a random, more informative but potentially also more ambiguous selection on the other side.

7.2.5 *Labeling Strategy*

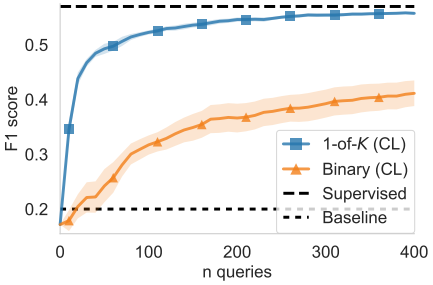
The labeling of queried video clips should be correct and quick. The labeling strategy influences both properties. We propose three labeling strategies:

1. 1-of-K, fixed-order labeling: the user selects the true label from a list of all options with some fixed ordering (e.g., alphabetical).

2. 1-of-K, ranked labeling: the user selects the true label from a list of all options that is sorted by the classifier's confidence. The confidence level is also indicated in some visual representation such as colored bars.

3. Binary labeling: the user selects whether or not the shown clip belongs to one specific target class.

Intuitively, we expect the labeling time to gradually decrease from strategy 1 to 3. In the ranked list, the user should find the true label among the first two items most of the time if we assume that the model is reasonably accurate after a few iterations. Otherwise the time should be comparable to the fixed-order case. Responding with only Yes or No should be even faster as the user only needs to confirm or reject instead of deciding between K classes.



(a) Learning curves

	AUC		F1	
	mean	std	mean	std
1-of-*K* (CL)	0.53	0.01	0.56	0.01
Binary (CL)	0.34	0.08	0.41	0.08

(b) Statistics

Figure 7.6: Comparison of 1-of-*K* (fixed-order) and binary labeling strategies.

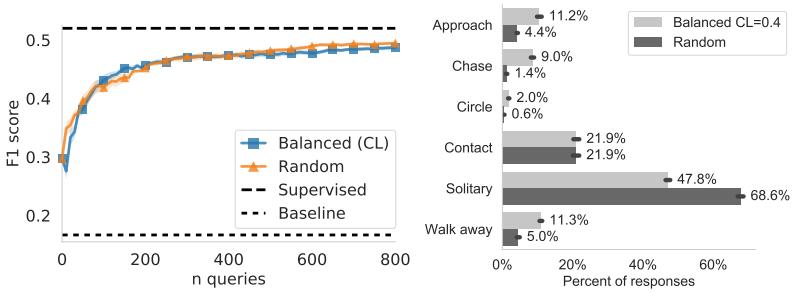
The 1-of-*K* and binary labeling strategies differ in the amount of information that the classifier gains from one user response. In the 1-of-*K* strategies, we always receive the true label irrespective of how the classifier predicted the sample. In the binary labeling strategy, the information depends on the response. With a positive response, we receive the true label. With a negative response we only learn that the proposed target class is incorrect but not which class would have been correct.

In the following experiment we focus on how the framework deals with the binary labeling strategy. Because the order of the labeling options is irrelevant for the data oracle, we disregard the ranked strategy for now and will return to it in the user evaluation in Section 7.4.2. We use CL = 0.4 for both 1-of-*K* and binary labeling as it is the optimal parameter for both strategies (data not shown for binary labeling).

As seen in Figure 7.6 the learning performance with binary labeling is substantially lower. Both learning rate and performance after 400 iterations show that the binary labeling responses carry less information and more labeled examples would be needed to reach the same accuracy. Binary labeling does not appear to be an adequate strategy for these data.

7.2.6 *Validation on CRIM13*

We validate the efficacy of the interactive framework on the public CRIM13 mouse social behavior dataset. The social interactions in CRIM13 are similar to RatSI and the mice also display *approach*, *chase*, and *walk away* behavior. In addition, there is *circling* behavior in which one mouse moves in circles around the other. *Circling* occurs only sporadically and is therefore a minority class in CRIM13. As for RatSI, we treat all close-contact interactions, which we cannot distinguish reliably with the current tracking and



(a) Learning curves

(b) Labeled examples after 800 iterations

Figure 7.7: Learning performance on CRIM13 dataset.

feature extraction algorithms, as one *contact* behavior. Similarly, we join actions performed by the individual animal such as *drink* and *eat* and create a *solitary* class. The label set then consists of six labels: *approach*, *circle*, *contact*, *following*, *moving away* and *solitary*.

For CRIM13 we use a smaller feature set with seven features as the location tracking provides only a single point instead of multiple body points. Detailed body point distances are thus not available and the orientation has to be estimated from motion, which we found to be less reliable (Section 4.3.1). We use the distance d_{cc}, relative orientation φ, relative heading $\cos(\gamma)$, the derivative of each of the former as well as the velocity v_c. The experiment set, which is the set of videos to be labeled, consists of the training videos as used by the dataset authors. We use the corresponding test videos for computing the validation scores below. We omit videos with anesthetized intruders as these lack the relevant social behavior.

We perform the same learning experiments as before using the data oracle and compare the balanced confidence-weighed sampling strategy with the random strategy. The same learning and classifier parameters are used ($C = 0.1$, $\mathrm{CL} = 0.4$). The results are shown in Figure 7.7.

The framework is able to train a classifier for the CRIM13 dataset with a different feature set and slightly different behavior categories. Compared to RatSI we need more training examples and the accuracy after 800 iterations (0.48) has not converged to the supervised reference performance yet (0.52). Looking at the ratio of examples that have been labeled per class in Figure 7.7b, we notice that the balanced strategy has indeed selected more minority examples than the random strategy. Despite this 2 to 7-fold increase in minority samples, we do not observe any substantial difference in learning performance between the two sampling strategies.

7.3 User Evaluation of the Annotation Framework

We now apply the annotation framework in practice using human annotators. In this user study we focus on the choices we made in Section 7.2 regarding the different framework settings: is a human annotator able to train an accurate classification model with the same settings?

7.3.1 *Experiment Setup*

7.3.1.1 *Framework Settings*

We use the framework settings that achieved the highest performance in the experiments in Section 7.2. We set the confidence level for the sample selection to CL = 0.4. Both learning algorithm and initial samples are the same as in the offline experiments.

7.3.1.2 *User Interface*

The user interface of the annotation framework as depicted in Figure 7.8 was specifically developed for this experiment. It shows the experiment progress, the video screen and the labeling buttons. During the experiment, the queries are automatically selected and presented by the framework. The video clips of one second are displayed in a continuous loop until the user responds by clicking one of the label buttons. Immediately after a response the next clip is played. Users have the option to reject a query if they are unsure about the correct label. As before, rejected queries count as one iteration but the corresponding samples are ignored for learning. Rejected samples cannot be queried again.

7.3.1.3 *Protocol*

All participants received the same written instructions about the labeling task. Next to a brief written description of the behaviors, participants were shown a short video with typical examples of each behavior. There was no time restriction for how long participants would familiarize themselves with the material (typically less than 5 minutes). From the five participants in total, one had no prior experience in labeling rodent interactions, and four had labeled rodent behavior before but were neither considered experts nor received professional training.

Given that the learning curves converge after approximately 300 iterations, we asked participants to label 300 clips per labeling strategy. After every 50 clips, the labeling process was interrupted so that participants could have a short break. Participants continued at their own pace.

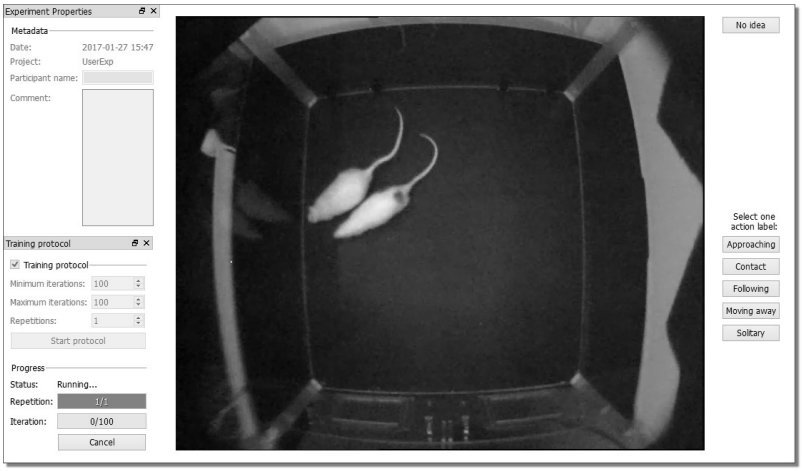


Figure 7.8: User interface of annotation framework used in user study.

7.3.1.4 *Measurements*

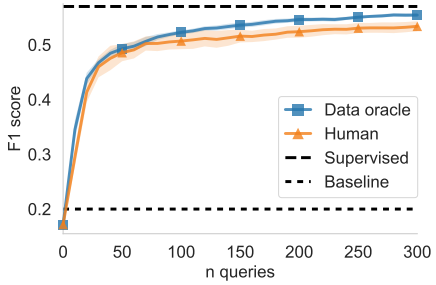
In addition to the learning curve and AUC as introduced in Section 7.2.1, we time the participants' responses. The response time is the time from the start of a video clip to the moment the user clicks on a label. We report the sum of all response times as the *total annotation time*. Scores are averaged across participants and mean and standard deviation are reported.

7.3.2 *Results*

The participants needed on average 18.7 minutes for labeling the 300 clips excluding breaks and they rejected on average 5.7% of the queries as *uncertain*. The average time needed for labeling one minute of video using the interactive framework was 3.7 min which is comparable to the time needed in traditional, sequential annotation tasks, usually between 3 and 10 min [3, 18, 145].

In contrast to traditional labeling, the interactive framework not only obtains annotations from the user, it also trains a behavior classifier at the same time. In Figure 7.9a we see that after 300 labeling iterations the classifiers are as accurate as classifiers trained by the data oracle. Also the learning rates (AUC) are comparable. The results validate the framework settings that we determined with the data oracle.

Note that our experiment setup causes small differences in performance figures because the data oracle uses the ground truth annotations from



(a) Learning curves

	AUC		F1	
	mean	std	mean	std
Data oracle	0.52	0.01	0.55	0.01
Human	0.50	0.03	0.53	0.02

(b) Statistics

Figure 7.9: Learning performance with human annotators on RatSI.

the expert who labeled our dataset in advance. This yields two advantages for the data oracle. First, the data oracle's score does not suffer from the inter-annotator disagreement between its labels and those in the validation set [49, 80]. Second, the expert was not restricted to labeling one second clips and could thus exploit contextual information. Therefore, the labeling of the data oracle can be assumed to be more informative.

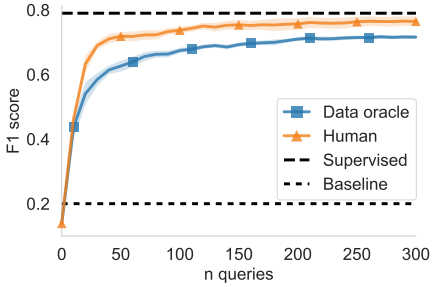
Let us take a look at the annotations that are automatically generated by the trained classifiers. We compare the annotations to each other directly by calculating the F1 score between each pair of annotations. Table 7.1 averages the results within and across groups, that is repetitions (data oracle) and study participants (human). Note that these are not accuracies in terms of the ground truth labels but agreements among the *predictions* of different classifiers. Within each group, oracle or human, the agreement is high with an average F1 score of 0.80 and 0.79, respectively. Between oracle and humans the agreement is lower with a score of 0.70. We find that the human-trained classifiers tend to predict more often *moving away* and less often *following* compared to the oracle-trained classifiers. This difference in the prior probabilities seems to be largely responsible for the lower agreement score.

Table 7.1: Average pairwise agreement (F1 score averaged across behaviors categories) among the annotations generated by the trained classifiers.

	Data oracle		Human	
	mean	std	mean	std
Data oracle	0.80	0.03	0.70	0.05
Human	-	-	0.79	0.04

7.3.3 *Cross-dataset Validation on YR*

To evaluate whether the classifiers trained by the users are applicable beyond the training videos, we validate them on the YR dataset. This cross-dataset validation is essentially a replication of our experiments in Chapter 6. In addition to the overall performance of the user-trained classifiers, we also compute the learning curves using the intermediate classifiers after every labeling iteration. For reference, we include the supervised performance of the classifier trained and tested on YR using a 5-fold cross-validation as in Section 6.2.2.



(a) Learning curves

	AUC		F1	
	mean	std	mean	std
Data oracle	0.66	0.02	0.72	0.02
Human	0.72	0.03	0.76	0.04

(b) Statistics

Figure 7.10: Learning performance with human annotators validated on YR.

The results are shown in Figure 7.10. The human annotators of our study outperform the data oracle in both learning rate and classification performance after 300 iterations. The classification performance (0.76) is only slightly lower than the supervised, within-dataset reference (0.79). This difference between within and cross-dataset evaluation (0.03) is in line with our previous observations in Section 6.3 (0.03 and 0.02 using SVM-Lin and GMM, respectively).

Remarkably the data oracle performs substantially worse. A look in the per-class accuracies reveals that the classification performance on the *moving away* category is worse than for human annotators. As mentioned before, the human annotators have labeled more *moving away* samples than the data oracle. This difference in prior probabilities appears to result in an accuracy difference on the YR dataset, where on RatSI it did not affect the average performance. A possible explanation is that *moving away* interactions in YR typically include clear and sudden changes in velocity. As the rats in RatSI are slower, the boundary between walking and running are more vague. The participant labels seem to align better with the characteristics of the young rats' behavior.

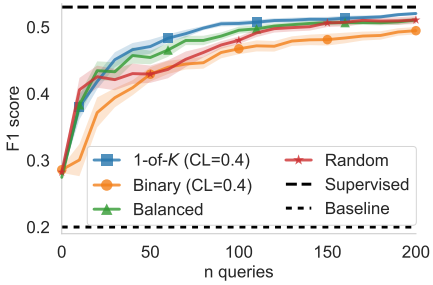
Overall, the cross-dataset result demonstrates that our annotation framework allows training rodent behavior classifiers that are not necessarily limited to one setting. In practice, this can decrease the manual effort even further as the trained classifier may be used to annotate also new videos beyond the initially considered experiment. Naturally, the concerns regarding cross-dataset validation that we raised in Chapter 6 also apply to the interactive annotation setting and some form of evaluation is recommended.

7.4 Scaling Toward Learning in Larger Datasets

So far we considered efficiency only in terms of annotation time and manual effort. We thereby disregarded the computational complexity of training the classification model. We have chosen a linear classification model that not only compared favorably against other classifiers with respect to accuracy but also efficiency. Together with the relatively low number of samples, our specific classification task has proven to be computationally inexpensive. We now want to make a first step toward scaling this learning framework to deal with larger, computationally more demanding datasets. Specifically, we replace the Coordinate Descent (CD) learning algorithm we have used until now with a potentially more efficient algorithm: Stochastic Gradient Descent.

Stochastic Gradient Descent (SGD) [155] is a popular choice for large-scale learning tasks such as training convolutional neural networks [14, 132]. SGD is a gradient-based minimization algorithm that aims to find the minimum by following the steepest gradient of the objective function. In large training sets with high-dimensional features, computing the gradient is an expensive operation. SGD reduces the computation time by approximating the gradient using only a random subset of the samples. Although the gradient computation is quicker, because it is an approximation the algorithm may require more iterations to converge to the minimum than an exact algorithm. A learning schedule may reduce the number of iterations by, for example, altering the step size with which the gradient is followed. Learning schedules introduce additional parameters that need to be tuned in order to optimize the accuracy of the final classification model. We use it to minimize the training error with the same objective function as before (Equation 7.3 in Section 7.2.3.1).

In this section, we examine how using SGD affects learning rate, classification performance and computation time by performing the same experiments as in Section 7.2. We further conduct another user study including an additional labeling strategy which ranks the labeling options and presents them in the order of their likelihood (Section 7.4.2).



(a) Learning curves

	AUC		F1	
	mean	std	mean	std
1-of-*K*	0.48	0.01	0.52	0.01
Binary	0.44	0.02	0.49	0.02
Balanced	0.48	0.02	0.51	0.01
Random	0.47	0.02	0.51	0.02

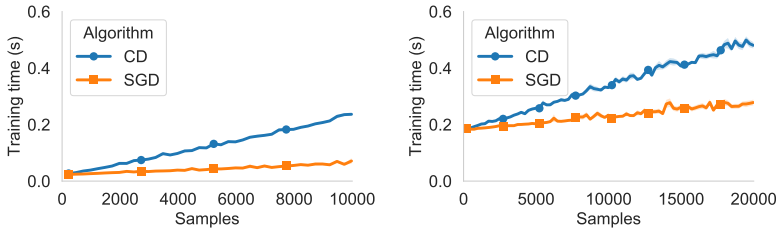
(b) Statistics

Figure 7.11: Learning performance using SGD and data oracle on RatSI.

7.4.1 *Results: Data Oracle*

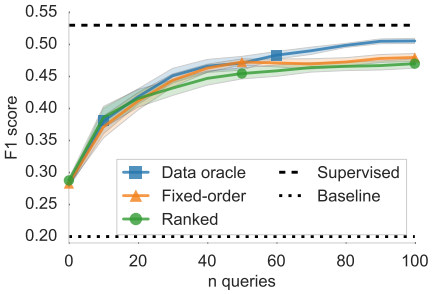
We first look at learning rate and classification performance achieved with the data oracle. Figure 7.11 summarizes the four learning settings with which we have experimented before: random sample selection (random), balanced selection (balanced), and the two labeling strategies (1-of-*K* and binary) applied to balanced selection with confidence level CL = 0.4. The supervised reference performance is given by the model trained with SGD using all potential training samples. As with the other learning algorithm, we find that the differences among the selection settings are modest. Only the binary labeling strategy cannot compete due to the lower amount of information carried by the binary labels. Overall, SGD-trained classifiers appear to converge in fewer iterations but yield a lower absolute performance (0.52) than the CD-trained classifiers (0.56).

Turning to the computational complexity, we measure the time the learning algorithm needs to train the classification model. Figure 7.12 shows how the training time evolves as the labeled training set becomes larger. Both algorithms are fast in absolute terms, but SGD scales more favorably toward larger datasets. Comparing between the two datasets, CRIM13 requires an additional 0.2 seconds irrespective of the number of training samples. This is due to a caching operation in our framework which takes longer for the much larger CRIM13 dataset. When we consider both time and learning performance, we recommend using an approximating learning algorithm such as SGD only when the training time otherwise significantly increases the total annotation time for the user.



(a) RatSI (b) CRIM13

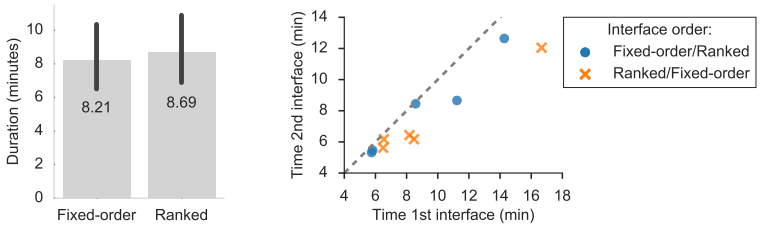
Figure 7.12: Time required for model training by Coordinate Descent (CD) and Stochastic Gradient Descent (SGD) for increasing training set size.



	AUC		F1	
	mean	std	mean	std
Data oracle	0.46	0.01	0.51	0.01
Fixed-order	0.44	0.02	0.48	0.02
Ranked	0.44	0.03	0.47	0.02

(a) Learning curves (b) Statistics

Figure 7.13: Learning performance using SGD with human annotators on RatSI.



(a) Labeling strategies (b) Participants represented by circles and crosses

Figure 7.14: Total annotation time needed by human annotators for 100 clips.

7.4.2 *Results: Human Annotators*

In this second user study we move our focus toward different labeling strategies: if we let the user know about the classifier's confidence in a query, does it decrease the response time? Or does this information bias his decisions and thus harm the learning performance? We address these questions by comparing the 1-of-K labeling strategies (fixed-order and ranked).

We asked ten participants to label 100 clips per labeling strategy. Each participant performed the annotation task subsequently using both strategies, hence labeling 200 video clips in total. To be able to reveal any learning effects between the first and the second trial, we alternated the order of the strategies between participants. After every 20 clips, the labeling process was interrupted so that participants could have a short break. Participants continued at their own pace.

Comparing the fixed-order and ranked strategies in Figure 7.13, there seems to be no clear winner in terms of performance. Although the fixed-order strategy reaches a slightly higher mean AUC, it shows no substantial gain given the variance among participants. Turning to the annotation time in Figure 7.14a, there is again no clear advantage in using either strategy. The average annotation time is similar, 8.2 min for fixed-order and 8.7 min for ranked, but varies from approximately six to eighteen minutes across participants. In Figure 7.14b we consider the annotation time of the participants individually. All participants were faster in the second trial irrespective of which interface they used first. This indicates a learning effect in the participants leading to shorter durations in the second trial.

As a final aspect we address the bias of users with respect to specific labeling responses. Recall that the sampling strategy attempts to balance the selection across classes. The algorithm uses the predictions of the unlabeled samples to select a sample of the desired target class. Because the selection is based on a prediction, the sample may belong to a different

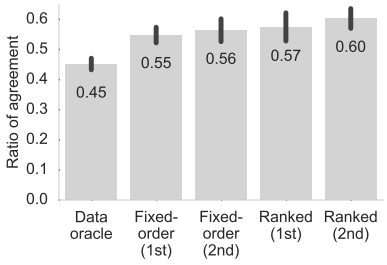


Figure 7.15: Agreement between target class and response by human annotators.

class than anticipated. In Figure 7.15 we display the ratio of agreement between the target class and the actual labeling response. We first notice that the data oracle agrees less often with the target class than the human annotators. This effect may suggest that human annotators tend to label a specific interaction even if it is only partly visible in the clip. The majority vote applied by the data oracle forbids such flexibility. This explanation is in line with the increase of agreement for users who used the ranked strategy in their second trial. The ranked strategy openly displays the classifier's confidence in each label. In the second trial, the users were already familiar with the interface and the behaviors. It is possible that the familiarity with the task caused these users to more often accept the proposed target class if there was at least some agreement with the shown behavior. Despite this slight bias, all users eventually trained classifiers with very similar performance, indicating how modest the effect of the interface on the annotation performance is.

7.5 Discussion

The presented interactive framework enables researchers to quickly annotate rodent behavior videos with a strongly reduced amount of work. The key to the reduced effort is to put the human in the annotation loop. With this active learning setup, we can avoid sequential and redundant labeling of similar samples and decide to stop labeling once the classification is sufficiently accurate. From our offline experiments with a dataset oracle, we conclude that the dominant factor for reducing labeling effort is the non-sequential labeling with a stopping criterion. Although balancing samples across rodent interaction classes and selecting more uncertain samples slightly improve the learning rate, the impact on the reduction is modest. We confirmed these results with another set of learning experiments on the mouse behavior dataset CRIM13 using a smaller feature set and slightly different interaction categories.

That the number of labeled samples rather than the information they carry appears to be more important for learning, indicates that our sampling strategies are not yet fully effective. On the one hand, the limited effectiveness could be caused by an inadequate criterion for the expected information. We based this criterion on the uncertainty of the classifier. On the other hand, a qualitative assessment of selected clips of different levels of uncertainty gives clear and intuitive results. Clips that are predicted with high certainty show indeed clear and prototypical interactions. Clips with low certainty often show ambiguous behavior or transitions between interactions. The disagreement between the qualitative assessment

and the quantitative effect on learning demonstrates the discrepancy between what seems informative to a human and what is actually informative for a learning algorithm. Similar effects are found for fine-grained object recognition where ambiguity in the most uncertain samples prohibits humans to choose a label [70]. A better criterion for selecting samples could be the expected increase of classification accuracy [24, 116] or the expected change of the classification model [47, 127].

Aside from the annotations, the framework also outputs a trained behavior classifier. This classifier can be used to annotate even more videos from similar experimental settings without additional manual labeling as we have validated in the cross-dataset application with the YR dataset. As such, the speed-up in annotation can even be larger. However, an interactively trained classifier is unlikely to be more general than a classifier trained on a large training set. Therefore the same concerns as in Chapter 6 can be raised regarding an adequate cross-dataset validation.

Considering the final classification model, we found the accuracy of the user generated annotations to be slightly lower than when the data oracle was used. The difference is presumably due to the inter-annotator disagreement between the user who labeled examples and the expert who labeled the validation videos. The data oracle uses examples that are labeled by the same annotator as the validation set and thus the accuracy does not suffer from the disagreement. We found no such advantage in the cross-dataset evaluation where the validation videos were labeled by another annotator.

Although the users in our study were novice users with little or even without any experience in rat behavior, they were all able to train an accurate classifier that is on par with the supervised classifier. Although the annotated interactions are rather intuitive and demand little experience from the annotators, this result encourages the application of the presented framework in practice. There are several practical questions to be addressed in the future in order to expand the functionality of the framework. For example, a common issue of the active learning approach is the quantitative evaluation of the classification performance. As training samples are selected automatically, they are not independent and can not easily be used for cross-validation. We may also address sample selection and learning in face of uncertain or wrong labeling decisions by the human annotator [124].

We have validated the framework mainly on a dataset with five rat social behaviors. If a dataset contains more behaviors, we will naturally need to label more video clips to capture examples of all behaviors. If the behaviors are more difficult to distinguish, for example because of temporal dependencies, we may need to provide more flexibility to deal with temporal

variability. A potential extension of the framework is to give the user the option to continue watching the video after the clip. Behavior transitions can be addressed by allowing the user to divide a clip and label each part separately. These extensions to the framework are solely in the user interface and do not require any modifications in the learning algorithm.

In conclusion, we have shown that interactive labeling can be used to annotate rodent behavior with strongly reduced manual effort. We are confident that extensions to the framework will allow for the annotation of an even larger range of relevant rodent behaviors in video.

8

Discussion

This chapter concludes the thesis. We summarize the main contributions and insights from our findings in Section 8.1, followed by a discussion of our work in Section 8.2 where we highlight achievements and limitations. We will give directions for future work in Section 8.3 and close with final remarks in Section 8.4.

8.1 Summary of Contributions

We investigated the automated annotation of rat social interaction videos in a systematic manner. We examined several aspects of automated annotation including tracking quality, feature representation, classification models, cross-dataset validation and adaptation to age.

- To enable these investigations and allow other researchers to build on our work, we have recorded and annotated a rat social interaction dataset (RatSI) and made the data publicly available (see `http://www.noldus.com/innovationworks/datasets/ratsi`).
- In our analysis of tracking and feature quality, we identified that inaccurate rodent location tracking and pose reconstruction limit the classification accuracy of close-contact interactions. These insights highlight the need for better tracking algorithms for multiple rodents or alternatively for research into feature representations that do not rely as much on tracking. With this limitation in mind, we focus on the classification-related challenges in absence of tracking artifacts.
- We performed cross-dataset classification experiments that revealed the importance of an adequate validation of rodent behavior classifiers. We demonstrated that environmental and experimental factors such as the animal age can induce behavioral variations that, if not properly addressed, can cause misclassifications. A possible approach to avoid misclassifications is to adapt the classifier to behavior variations before applying it to new data. We demonstrated this in an example case in which we adapted the feature representation to different animal ages.

- We introduced an interactive annotation framework that leverages the above insights. We investigated the properties of different sampling and labeling strategies to predict the optimal settings for our user study with human annotators. The framework allows users to annotate rodent behavior videos and simultaneously train a classifier with significantly reduced efforts. Study participants manually annotated in less than half an hour enough examples to train an accurate classifier that propagated the annotations throughout the remaining two hours of the videos. Interactive annotation enables neuroscientists and biologists to analyze behavioral data faster than before and allows them to study previous data in new light with limited manual work.

8.2 Discussion of our Findings

In this section we place our contributions in the context of related work and discuss the limitations of our approach. We group the discussion into four topics: observation, classification, cross-dataset application and interactive annotation.

8.2.1 *Observation*

We analyzed the role of tracking and feature quality in classifying rat social interactions. We found that both accurate tracking and a rich pose description are necessary for accurate interaction classification (Section 5.4). In particular, tracking errors cause misclassifications and can render pose features unreliable and hence less useful for classification. Although trajectory features such as velocity and distance are informative for many social interactions, they do not enable the classification of fine-grained, close-contact interactions such as nape attacking. For these, highly accurate tracking of body parts such as paws is needed, which is not yet possible with current tracking techniques. This stresses the importance of improving identification of body parts in the future.

Our analyses sketch the scope of interactions that can be annotated automatically and reliably with our current tracking method. Although the different contact interactions may not yet be annotated automatically, it is still possible to separate them from other non-contact interactions. This separation enables the human annotator to provide the detailed annotations manually while focusing on only the relevant subset of the videos. This reduces both annotation time and the risk of missing relevant interactions.

Clearly, the goal of future work on tracking and pose reconstruction is to enable the automated annotation of contact interactions. Recent work has shown the potential of 3D pose reconstruction from multiple viewpoints [128] or by using depth cameras [55, 87] (Section 2.2.1.4). These advances promise not only to make tracking of multiple rodents more robust to occlusion but also to allow for a richer description of the posture, perhaps even including the location of the paws. The main limitations for applying these systems in practice concern the high storage and processing demands, as well as the increased complexity of the setup which often requires technical expertise.

8.2.2 *Classification*

Besides tracking quality, there are other reasons for classification errors and confusion of behaviors. For instance, a large distance is a valuable cue for solitary activities, but the opposite is not always true. Animals that are in close proximity do not necessarily engage in interactions. They may individually explore the same area of the cage. Similarly, incidental movement towards or away from each other can confuse the classifier. In such cases, the animals indeed come closer or move away but they do not actively and seemingly intentionally approach or avoid each other. A human annotator is able to interpret the intention in such examples and labels them accordingly. On the one hand, the learned classification models are precise and objective: when an animal moves closer to another with a certain speed, it visually appears to *approach* the other. On the other hand, the classifier fails to reason about the intention of the animal which is not moving closer to engage the other animal in some interaction. Such reasoning capabilities are desired but difficult to facilitate in a frame-based classification model. Temporal context is not taken into account in a variable manner and over an extended period of time, and start and end states cannot be incorporated explicitly. We will suggest a possible approach to higher level reasoning in Section 8.3.

8.2.3 *Cross-dataset Application*

Regarding automatically annotating rodent social behavior, one of two scenarios can be encountered in practice, namely: there is a pre-trained classifier available for the behaviors of interest, or there is no such classifier. If there is a classifier and it is applicable to the present experiment setting, no manual effort is required for annotating the videos. Whether a classifier is applicable in a certain setting depends on a range of factors including the

original training set, the behavior variations therein, as well as the cage size, the animal's age or genetic background.

Because of the variety of these factors, it is difficult to predict whether newly acquired recordings are suitable to be annotated by an existing classifier. We have demonstrated in Chapter 6 that even varying only a single aspect, namely the age of the rats, can cause a significant decline in annotation quality. This insight has a direct impact on longitudinal behavior studies in which the animals are repeatedly observed at different points in their life.

We therefore argue for validating classifiers by appropriate cross-dataset experiments. This has rarely been done in a systematic way in previous work. Besides avoiding biased annotations in practice, identifying a bias allows us to attend to it properly. We may improve the annotation method so as to deal with the variations systematically or adapt either classification model or feature representation to the present settings in a one-time effort. For instance, in our experiment in Section 6.2 we were able to compensate for the age difference by normalizing feature values to a common range across ages. This is a promising result that demonstrates the potential to extend trained classifiers to new settings and behavior variations. Moreover, being able to join data from various settings increases the amount of available training data, which in turn makes the classification models more robust.

The challenge for adapting classifiers to different experimental settings is that it is based on low-level features such as velocity and distance. These depend to a large degree on the acquisition setup, the viewpoint, the size of the animals and the cage. Given the large number of possible combinations, it is not clear how adaptation should be approached in general. There are many open questions, such as whether it is better to adapt the classifier to the data or vice versa, and whether labeled examples from the new recordings are needed to inform and validate the adaptation. It is also uncertain what the limits of adaptation are and how many factors may change simultaneously before it becomes infeasible. Studying these questions in future research will improve adaptation for rodent behavior recognition and related pattern recognition problems. Furthermore, it will advance our understanding of the factors that are responsible for natural and induced behavior variations in rodents.

8.2.4 *Interactive Annotation*

In the case that no classifier is available or existing classifiers prove unsuitable, we are left with obtaining annotations manually. For this scenario we

have presented an interactive annotation framework that allows annotating large sets of behavior videos and simultaneously learn a classification model with significantly reduced manual effort. Our framework trains the classification model with examples that the user labels one-by-one. Being included in the training procedure, the user can continue labeling examples until the classifier has reached a sufficient accuracy. Manual labeling is then stopped and the remaining videos are annotated automatically. As a result, all videos are annotated and a new classifier has been trained that can be used to annotate even more rodent data. In user experiments, we have demonstrated that users are able to train an accurate rat interaction classifier by labeling 300 clips of one second, which took on average less than thirty minutes (Section 7.3).

We evaluated several sample selection algorithms that decide which yet unlabeled video clips may be worth labeling. We compared algorithms that attempt to balance the selection among behaviors and prefer clips about which the classifier is yet uncertain. These selection strategies improve learning performance slightly compared to random selection but are not able to reduce the labeling effort substantially. Other sample selection strategies could be more successful [47, 116]. The largest reduction of manual effort that we achieved stems from the ability to select samples from anywhere in the videos and to stop labeling once the accuracy is sufficient (Section 7.2).

One detail that our framework still misses is the monitoring of the convergence of the classification performance. In our experiments we retained a set of two already annotated videos for the monitoring task. These are usually not available in practice. Hence, we need a different solution to enable the user to assess the current performance. One option would be to assign a few of the user-labeled clips as test samples and evaluate the performance on those. However, this test set would not be independent as it is selected by the algorithm based on some information criterion. The resulting performance measure would be biased. The same problem arises when the monitoring is based on the convergence of the training error. Alternatively, the framework could regularly select random clips that the user labels purely for evaluation purposes. This increases the labeling effort for the human. At last, evaluation could simply be left to the user who makes a qualitative assessment of the annotations (as in [61]).

The proposed framework addresses one specific aspect of the annotation task, namely the classification. We have therefore assumed that the features needed for classification are readily provided. A logical extension is to involve both feature design and classification in the interactive framework. A simple extension, as implemented in the JAABA framework [61],

would provide the user a list of possible features to choose from. Making a suitable choice requires a certain amount of experience with computer vision and presumably involves some trial and error. Additionally, if classification fails, it is unclear to the user whether the number of labeled examples is insufficient or the choice of features inadequate.

Selecting suitable features automatically could circumvent this issue. For example, Crispim-Junior *et al.* [29] select features based on the distribution of their values to ensure they differ significantly among the four considered solitary behaviors. The method is applied in a supervised learning setting and it is uncertain whether it works in an active learning framework where the number of labeled training examples is limited. A semi-supervised approach may be better suited as the large set of unlabeled data can be enriched with a few labeled examples. Inspiration could come from work on text categorization [107, 126] where users label not only examples but also relevant features.

In the bigger picture, our approach to measure behavior using interactive labeling does not excuse us from meticulously defining the relevant behavior categories. As long as the annotator labels the behaviors consistently, the classifier will learn to consistently annotate accordingly. In other words, if two annotators use our framework to annotate the same set of videos but use different interpretations of the behavior definitions, the resulting annotations may still exhibit disagreement. We have observed this effect in our user study (Section 7.3.3): the participants appeared to agree more with the annotator of the YR dataset than of the RatSI dataset, leading to a higher accuracy of the respective classifiers for YR. Our approach can improve the intra-annotation agreement, as the classifier produces consistent annotations based on the user input, but it has no effect on the inter-annotator agreement. This restriction is a general concern of measuring behavior based on human-defined categories and therefore constrains any automated method that learns from user-labeled data.

Finally, we want to briefly reflect on our long-term goal of a general, unified framework for rodent interaction recognition across settings and behaviors. Since it is difficult to prove the generality of our approach, we cannot provide hard evidence on how much our work contributes to this goal. Although our intention is to draw general conclusions from our experiments, our work is not free of assumptions. Assumptions are sometimes made explicitly by limiting the scope and complexity of experiments, and sometimes implicitly by deciding to work with one of the few available datasets. We sought to show that our insights indeed generalize beyond the considered scope, for instance, by validating the interactive framework on other rodent behavior datasets such as CRIM13 and YR.

8.3 Future Directions

We have presented our work on rodent interaction recognition with which we have gained valuable insights into the aspects that play a role in understanding rodent social behavior. Our interactive annotation framework facilitates the research of neuroscientists and biologists who seek to advance our knowledge of neurological disorders and animal behavior in general. In this section, we address future challenges and discuss extensions to our approach that could make it available to a wider range of applications. We specifically address feature representation and reasoning about behavior.

The requirement of our framework that suitable features are provided limits its applicability to scenarios in which we know what suitable features are. The aforementioned approach to automatically select features implies that there is a list of features to choose from. For rodent behavior recognition, it seems infeasible to design such a list that captures every possible aspect of appearance and motion for multiple animals. Hence feature *selection* may not be the right direction to pursue. Instead we rather want to learn suitable feature representations from behavior data. This leads us to end-to-end frameworks in which both features and classification are learned jointly using a complex, non-linear model.

Since they have proven successful in other visual inference tasks such as object detection and human action recognition, end-to-end frameworks have also received attention in the domain of rodent behavior recognition [69, 108]. The work focuses on analyzing individual rodent behavior and seeks to circumvent the difficult tracking problem by directly feeding the video images to a convolutional neural network (CNN). The networks are pre-trained using training images for an object detection task and then fine-tuned with rodent video images. It is yet unclear what these networks learn about rodent behavior and whether they extend to more fine-grained behaviors and interactions. The results, for example, show that a network based on single images outperforms a network that encodes motion [108]. This seems unintuitive as static appearance extracted from a single image certainly lacks relevant information about the rodent's motion.

A critical question to address in the future is whether the network architectures used in work on object detection are suitable for rodent behavior. Most popular architectures are developed for human made objects and natural scenes and therefore encode visual features ranging from edges and texture to object parts such as car wheels and human eyes. Whether the same architectures are equally adequate for laboratory environments in which the most relevant objects are the rodents, is an open question. To answer this question, and to be able to train these networks from scratch,

we certainly need to devote more effort to creating larger databases of annotated rodent behavior videos from various settings than are available at the moment.

An intermediate solution could be to first work with networks that are more constrained to a specific task before focusing on complex, general models. Eyjolfsdottir *et al.* [41] for example propose a recurrent network that hierarchically decomposes fruit fly behavior into *motion* and *action*. The input to the network are not raw video images but previously tracked animal locations. This allows concentrating on the behavior representation and avoids blending the influences of tracking, representation and classification. The proposed network learns in a semi-supervised setting, that is, only a few annotations are provided by the user. It would hence connect seamlessly with our interactive annotation framework.

Regarding reasoning and interpretation of social behavior that would allow us to distinguish for example between an incidental and intentional approach, we may need to abandon frame-based classification and move toward a top-down classification approach. A top-down approach seeks to optimally segment a video into labeled behavior events.

To give an example, an *approach* is not only described by a decreasing distance with some minimum velocity, but also by the distance and the pose at the start and the proximity at the end. It is easier to define *approach* in terms of a starting and ending state than the transition between them. An approach begins when an animal decides to approach another animal which causes a distinct change in velocity, posture or orientation towards the other animal. Eventually, the consequence of an approach is that the animal is close to the other. Eyjolfsdottir *et al.* [42] apply a similar concept to annotate the behavior of fruit flies and mice, where features are designed manually to capture how behavior events start, end and progress over time.

The challenging task then is to learn these intermediate behavioral states automatically from trajectory and pose data. That such a representation may exist has been demonstrated by Drai *et al.* [39] who showed that rats have at least two intrinsic states of locomotion that are manifested by different velocity modes (roughly: lingering and intentionally changing position). In their work, these states are learned from trajectory data without supervision. Similar ideas are followed by Berman *et al.* [12] and Wiltschko *et al.* [149] who decompose trajectory and pose data from fruit flies and mice, respectively, into smaller recurring behavioral units. They show that these units, as they evolve over time, follow a specific grammar which we eventually observe as behavior. An open question is whether these units can also model *interactions* between animals.

Suppose we find such an intermediate representation of interactions. Reasoning about interactions in that representation could not only improve classification, it would also decouple classification from low-level features and therefore from the specific acquisition environment. This can make classification independent of how features are generated and from what type of input data. The input data could be the locations of the animals but also the raw video images from which features are extracted using a CNN. This independence would allow us to transfer or adapt classifiers more easily to other environments, genetic strains and perhaps even species.

8.4 Conclusion

Automated annotation of rodent social behavior videos involves various tasks including locating the animals, estimating pose and motion, classifying actions and eventually validating the annotations. In this thesis, we have related each of these tasks to the quality of the automated annotations. We analyzed previously known challenges such as the difficult tracking in contact situations and discovered new challenges such as behavior variations in cross-dataset applications. We thereby broadened our understanding of the prevailing issues of rodent interaction recognition, which helps to shape future research efforts.

Perhaps the strongest motivation to automate the annotation process is to reduce the manual effort for the human observer. We have achieved a significant reduction of the labeling effort with our interactive annotation framework. The framework enables neuroscientists and biologists to annotate behavioral data quicker than ever before, to analyze previous data in new light and eventually to advance our knowledge of rodent behavior in general. Our work therefore contributes to the goals of the three Rs [118] – Replacement, Reduction and Refinement – in that it refines behavior measurements to be more consistent and more efficient. The latter can reduce the number of animals because previous experiments can be reanalyzed when a new hypothesis emerges with a modified behavior repertoire.

An important challenge to address in the future is the extension of annotation methods to a wider range of behaviors and experiment settings. In order to reduce dependence on external factors such as the acquisition setup, the specific animal population and the experiment protocol, we need to progress toward a unified framework that reasons about behavior on a more abstract level than low-level velocity and distance features. A unified framework would be more versatile and more reliable than current methods and can lift automated behavior analysis to the next level.

Samenvatting in het Nederlands

Het meten van sociaal gedrag van knaagdieren is van belang binnen verschillende onderzoeksvelden. In de neurowetenschap bijvoorbeeld wordt knaagdiergedrag bestudeerd om de pathologie en de ontwikkeling van neurologische aandoeningen, zoals de ziekte van Huntington, beter te begrijpen. Daarnaast is het belangrijk om een oog te houden op sociaal gedrag voor het dierenwelzijn. Een tekort aan sociaal contact kan duiden op een ongezonde leefomgeving.

Het voornaamste doel om gedrag te meten is om het te kunnen vergelijken. Dit kan een vergelijking zijn met een wenselijk toestand (bv. een gezonde leefomgeving) of tussen verschillende dierpopulaties (bv. gezonde versus zieke dieren). Hiervoor moet het gedrag objectief gekwantificeerd worden, meestal door aantekeningen in een gedragsprotocol te maken. Hierin wordt elke actie of interactie van de dieren genoteerd. Vaak wordt ook bijgehouden hoelang de acties duren. Dit soort aantekeningen wordt tijdens observaties in de natuur of op basis van video-opnames uit laboratoria gemaakt.

In het algemeen is het handmatig bijhouden van een gedragsprotocol een langdurige en soms subjectieve taak. Het duurt gewoonlijk drie tot twaalf keer zo lang als de lengte van de video [3, 18, 145]. Daarom is het wenselijk om ten minste een deel van dit werk te automatiseren. Naast de tijdbesparing heeft een automatische meetmethode nog andere voordelen. Ze is objectief, levert reproduceerbare resultaten en kan doorwerken zonder vermoeid te raken. De grootste uitdaging is de ontwikkeling van een methode die, net als de mens, flexibel kan omgaan met variabele omgevingsfactoren. De mens heeft een waarnemings- en interpretatievermogen die nog steeds niet volledig te vervangen zijn door een machine.

De eerste automatische meetmethoden voor knaagdiergedrag concentreren zich op het meten van bewegingen van individuele dieren. Door de locatie van het dier gedurende de video te volgen, wordt informatie verzameld over hun voorkeur voor bepaalde plekken, hun snelheid, de afgelegde afstand of hoeveel tijd ze doorbrengen buiten hun schuilplaats [121, 135]. Door verbeterde beeldverwerkingstechnieken is het mogelijk geworden om specifieke acties zoals lopen en opstaan automatisch te herkennen en ook de duur van deze acties te bepalen. Deze technieken werken goed voor het meten van individueel gedrag maar voldoen nog niet om interacties tussen dieren te herkennen.

Het herkennen van sociale interacties brengt nieuwe uitdagingen met zich mee. Ten eerste zijn er nu meerdere dieren wiens locaties gevolgd moeten worden. Vaak zijn de dieren van dezelfde genetische afkomst, waardoor ze erg op elkaar lijken en snel verwisseld worden. Hoewel dit probleem nog niet helemaal opgelost is, hebben onderzoekers het aantal fouten door verbeterde technieken wel verminderd [113]. Dat geeft ons de mogelijkheid om een nieuwe stap te maken: het automatisch herkennen van sociale interacties.

Voor die herkenning is een algoritme nodig dat voor elk videobeeld berekent welk gedrag de dieren op dat moment vertonen. Voor dit classificatieprobleem wordt doorgaans een computatief model ontwikkeld dat de verschillende interacties onderscheidt aan de hand van numerieke kenmerken. Deze kenmerken bevatten vaak informatie die is afgeleid van de locaties van de dieren, zoals snelheden en bewegingen ten opzichte van elkaar. Ze kunnen echter ook direct uit het beeld worden opgemaakt, zoals de lichaamsvorm en -houding. Voordat het model gedrag kan onderscheiden, moet het aan de hand van voorbeelden leren hoe de verschillende interacties eruitzien.

In dit proefschrift presenteren we ons onderzoek naar automatische herkenning van sociaal gedrag tussen knaagdieren in video's. Ons doel is een methode te ontwikkelen die interactief, samen met de gebruiker, aantekeningen maakt van interacties in video's en tegelijkertijd een classificatiemodel leert. Zodra het model genoeg heeft geleerd, kan het de resterende video's verwerken en alle interacties markeren. Daarvoor richten we onze aandacht eerst op een aantal aspecten van automatische gedragsherkenning.

Om te beginnen introduceren we een nieuwe dataset (RatSI) die ons in staat stelt om sociaal gedrag van ratten te bestuderen (Hoofdstuk 3). Deze dataset bestaat uit negen video's met een totale lengte van 135 minuten. Alle video's zijn volledig geannoteerd door een bioloog. Daardoor zijn ze geschikt om classificatiemodellen te trainen en te evalueren. We hebben RatSI vrij beschikbaar gemaakt voor andere onderzoekers op `http://www.noldus.com/innovationworks/datasets/ratsi`.

Automatische herkenning is een complex probleem met meerdere facetten die onze aandacht vragen. In Hoofdstuk 4 analyseren we deze en identificeren we een aantal uitdagingen. Het classificatiemodel moet rekening houden met het feit dat de verschillende interacties met afwijkende frequenties optreden. Dit kan negatieve gevolgen hebben voor het resultaat, omdat zeldzaam gedrag minder voorbeelden oplevert om van te leren en om aan te toetsen. Daarnaast laten de gedragscategorieën soms ruimte voor interpretatie waardoor zelfs mensen het niet eens zijn over welke in-

teractie hoelang te zien is. In Hoofdstuk 5 stellen we uiteindelijk vast dat de moeilijke taak om meerdere, bijna identieke dieren te volgen directe consequenties heeft voor de kwaliteit van de herkenning. In het bijzonder als de dieren dicht bij elkaar en in contact met elkaar zijn, leidt occlusie tot fouten in het bepalen van locatie en lichaamshouding. Deze fouten beperken de mogelijkheden om interacties te herkennen die tijdens direct contact optreden (bv. paringsgedrag).

Nadat we een classificatiemodel hebben getraind, kunnen we het toepassen op video's die opgenomen zijn in een vergelijkbare omgeving. In Hoofdstuk 6 laten we zien dat het onverwacht moeilijk kan zijn om een omgeving constant en onveranderd te houden. Het probleem ligt in het feit dat de omgeving niet alleen door controleerbare factoren zoals verlichting en kooimaat beïnvloed wordt, maar ook door variaties bij de dieren zelf. Wij experimenteren bijvoorbeeld met video's van zowel jonge als oude ratten. Door het verschil in leeftijd ontstaan kleine veranderingen in het gedrag, met name in snelheid, die de nauwkeurigheid van de herkenning verminderen. We adviseren daarom om aandacht te besteden aan een geschikte cross-dataset validatie en om verder onderzoek te doen naar manieren om met zulke variaties systematisch om te gaan.

In de praktijk is niet altijd een reeds getraind classificatiemodel beschikbaar. Als bijvoorbeeld de gedragscategorieën zijn veranderd of uitgebreid, dan moet het gedragsprotocol toch handmatig worden gemaakt. In dit geval willen we proberen om het handmatige werk zo gering mogelijk te houden. We pakken dit probleem aan in Hoofdstuk 7 met een interactieve methode waarbij mens en machine samenwerken en elkaar aanvullen. De gebruiker begint door voorbeelden van elke interactie te annoteren. Een algoritme gebruikt deze voorbeelden om tegelijkertijd het classificatiemodel te trainen. Zodra er genoeg is geleerd, kan de herkenningsmethode het invullen van het gedragsprotocol overnemen. De resterende video's worden automatisch verwerkt. Om de tijd mogelijk verder te verkorten experimenteren we met verschillende strategieën om de aandacht van de gebruiker te richten op de meest nuttige voorbeelden.

Door de gebruiker actief bij de herkenning en annotatie van het gedrag te betrekken, kunnen we de tijd om een gedragsprotocol te maken wezenlijk verkorten. De deelnemers van onze gebruikersstudie trainden een nauwkeurig classificatiemodel binnen een half uur. Deze was vervolgens in staat om het gedragsprotocol automatisch in te vullen voor de resterende twee uur van de video-opnames. Deze interactieve aanpak stelt neurowetenschappers in staat om gedrag sneller te meten dan voorheen en om bestaande data in een nieuw licht te bekijken met beperkte inspanning.

Bibliography

[1] N. Adams and R. Boice, "A longitudinal study of dominance in an outdoor colony of domestic rats", *Journal of Comparative Psychology*, vol. 97, no. 1, pp. 24–33, 1983.

[2] J. Altmann, "Observational study of behavior: Sampling methods", *Behaviour*, vol. 49, no. 3, pp. 227–266, 1974.

[3] D. Anderson and P. Perona, "Toward a Science of Computational Ethology", *Neuron*, vol. 84, no. 1, pp. 18–31, 2014.

[4] T. Arakawa, A. Tanave, S. Ikeuchi, A. Takahashi, S. Kakihara, S. Kimura, H. Sugimoto, N. Asada, T. Shiroishi, K. Tomihara, T. Tsuchiya, and T. Koide, "A male-specific QTL for social interaction behavior in mice mapped with automated pattern detection by a hidden Markov model incorporated into newly developed freeware", *Journal of Neuroscience Methods*, Measuring Behavior, vol. 234, pp. 127–134, 2014.

[5] R. Bakeman and J. M. Gottman, *Observing Interaction: An Introduction to Sequential Analysis*, 2nd ed. Cambridge University Press, 1997.

[6] F. Balci, S. Oakeshott, J. L. Shamy, B. F. El-Khodor, I. Filippov, R. Mushlin, R. Port, D. Connor, A. Paintdakhi, L. Menalled, S. Ramboz, D. Howland, S. Kwak, and D. Brunner, "High-Throughput Automated Phenotyping of Two Genetic Mouse Models of Huntington's Disease", *PLoS Currents*, vol. 5, 2013.

[7] S. Bandla and K. Grauman, "Active learning of an action detector from untrimmed videos", in *Proc. Conf. Computer Vision (ICCV)*, 2013, pp. 1833–1840.

[8] S. A. Barnett, "An Analysis of Social Behaviour in Wild Rats", *Proceedings of the Zoological Society of London*, vol. 130, no. 1, pp. 107–152, 1958.

[9] R. Bellman, *Dynamic Programming*. Courier Corporation, 2013, 388 pp.

[10] Y. Bengio, "Learning Deep Architectures for AI", *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[11] Y. Benjamini, D. Lipkind, G. Horev, E. Fonio, N. Kafkafi, and I. Golani, "Ten ways to improve the quality of descriptions of whole-animal movement", *Neuroscience & Biobehavioral Reviews*, vol. 34, no. 8, pp. 1351–1365, 2010.

[12] G. J. Berman, D. M. Choi, W. Bialek, and J. W. Shaevitz, "Mapping the stereotyped behaviour of freely moving fruit flies", *Journal of the Royal Society, Interface*, vol. 11, no. 99, 2014.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007, 738 pp.

[14] L. Bottou, "Stochastic Gradient Descent Tricks", in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2012, pp. 421–436.

[15] K. Branson and S. Belongie, "Tracking Multiple Mouse Contours (Without Too Many Samples)", in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 1039–1046.

[16] K. Branson, V. Rabaud, and S. Belongie, "Three brown mice: See how they run", in *Proc. Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2003.

[17] J. Brodkin, D. Frank, R. Grippo, M. Hausfater, M. Gulinello, N. Achterholt, and C. Gutzen, "Validation and implementation of a novel high-throughput behavioral phenotyping instrument for mice", *Journal of Neuroscience Methods*, vol. 224, pp. 48–57, 2014.

[18] X. P. Burgos-Artizzu, P. Dollár, D. Lin, D. J. Anderson, and P. Perona, "Social behavior recognition in continuous video", in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1322–1329.

[19] T. M. Caro, R. Roper, M. Young, and G. R. Dank, "Inter-Observer Reliability", *Behaviour*, vol. 69, no. 3, pp. 303–315, 1979.

[20] M. Casarrubea, F. Sorbera, and G. Crescimanno, "Multivariate data handling in the study of rat behavior: An integrated approach", *Behavior Research Methods*, vol. 41, no. 3, pp. 772–781, 2009.

[21] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning", in *Advances in Neural Information Processing Systems (NIPS)*, vol. 13, 2001, p. 409.

[22] L. E. Clemens, E. K. H. Jansson, E. Portal, O. Riess, and H. P. Nguyen, "A behavioral comparison of the common laboratory rat strains Lister Hooded, Lewis, Fischer 344 and Wistar in an automated homecage system: Automated homecage observation of wild type rat behavior", *Genes, Brain and Behavior*, vol. 13, no. 3, pp. 305–321, 2014.

[23] J. Cohen, "A Coefficient of Agreement for Nominal Scales", *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[24] D. A. Cohn, Z. Ghahramani, M. I. Jordan, and D. Cohn, "Active learning with statistical models", *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[25] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models-Their Training and Application", *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[26] J. C. Crabbe, D. Wahlsten, and B. C. Dudek, "Genetics of mouse behavior: Interactions with laboratory environment", *Science*, vol. 284, no. 5420, pp. 1670–1672, 1999.

[27] J. N. Crawley, S. Szara, G. T. Pryor, C. R. Creveling, and B. K. Bernard, "Development and evaluation of a computer-automated color tv tracking system for automatic recording of the social and exploratory behavior of small animals", *Journal of Neuroscience Methods*, vol. 5, no. 3, pp. 235–247, 1982.

[28] C. F. Crispim Junior, C. N. Pederiva, R. C. Bose, V. A. Garcia, C. Lino-de-Oliveira, and J. Marino-Neto, "ETHOWATCHER: Validation of a tool for behavioral and video-tracking analysis in laboratory animals", *Computers in Biology and Medicine*, vol. 42, no. 2, pp. 257–264, 2012.

[29] C. F. Crispim-Junior, F. M. de Azevedo, and J. Marino-Neto, "What is my rat doing? Behavior understanding of laboratory animals", *Pattern Recognition Letters*, vol. 94, pp. 134–143, 2017.

[30] L. De Visser, R. Van Den Bos, W. W. Kuurman, M. J. H. Kas, and B. M. Spruijt, "Novel approach to the behavioural characterization of inbred mice: Automated home cage observations", *Genes, Brain and Behavior*, vol. 5, no. 6, pp. 458–466, 2006.

[31] F. de Chaumont, R. D.-S. Coura, P. Serreau, A. Cressant, J. Chabout, S. Granon, and J.-C. Olivo-Marin, "Computerized video analysis of social interactions in mice", *Nature Methods*, vol. 9, no. 4, pp. 410–417, 2012.

[32] R. de Haas, A. Nijdam, T. A. Westra, M. J. Kas, and H. G. Westenberg, "Behavioral pattern analysis and dopamine release in quinpirole-induced repetitive behavior in rats", *Journal of Psychopharmacology*, vol. 25, no. 12, pp. 1712–1719, 2011.

[33] C. Decker and F. A. Hamprecht, "Detecting individual body parts improves mouse behavior classification", in *Workshop on Visual Observation and Analysis of Vertebrate And Insect Behavior (VAIB)*, 2014.

[34] A. I. Dell, J. A. Bender, K. Branson, I. D. Couzin, G. G. de Polavieja, L. P. J. J. Noldus, A. Pérez-Escudero, P. Perona, A. D. Straw, M. Wikelski, and U. Brose, "Automated image-based tracking and its application in ecology", *Trends in Ecology & Evolution*, vol. 29, no. 7, pp. 417–428, 2014.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[37] R. A. Dielenberg, P. Halasz, and T. A. Day, "A method for tracking rats in a complex and completely dark environment using computerized video analysis", *Journal of Neuroscience Methods*, vol. 158, no. 2, pp. 279–286, 2006.

[38] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features", in *Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2005, pp. 65–72.

[39] D. Drai, Y. Benjamini, and I. Golani, "Statistical discrimination of natural modes of motion in rat exploratory behavior", *Journal of neuroscience methods*, vol. 96, no. 2, pp. 119–131, 2000.

[40] S. E. R. Egnor and K. Branson, "Computational Analysis of Behavior", *Annual Review of Neuroscience*, vol. 39, no. 1, pp. 217–236, 2016.

[41] E. Eyjolfsdottir, K. Branson, Y. Yue, and P. Perona, "Learning recurrent representations for hierarchical behavior modeling", in *Proc. Conf. Learning Representations (ICLR)*, 2017.

[42] E. Eyjolfsdottir, S. Branson, X. P. Burgos-Artizzu, E. D. Hoopfer, J. Schor, D. J. Anderson, and P. Perona, "Detecting Social Actions of Fruit Flies", in *Proc. Conf. Computer Vision (ECCV)*, 2014, pp. 772–787.

[43] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification", *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, Aug 2008.

[44] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional Two-Stream Network Fusion for Video Action Recognition", in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1933–1941.

[45] S. E. File and P. Seth, "A review of 25 years of the social interaction test", *European Journal of Pharmacology*, Animal Models of Anxiety Disorders, vol. 463, pp. 35–53, 1-3 2003.

[46] J. Freund, A. M. Brandmaier, L. Lewejohann, I. Kirste, M. Kritzler, A. Krüger, N. Sachser, U. Lindenberger, and G. Kempermann, "Emergence of Individuality in Genetically Identical Mice", *Science*, vol. 340, no. 6133, pp. 756–759, 2013.

[47] A. Freytag, E. Rodner, and J. Denzler, "Selecting Influential Examples: Active Learning with Expected Model Output Changes", in *Proc. Conf. Computer Vision (ECCV)*, 2014, pp. 562–577.

[48] L. Giancardo, D. Sona, H. Huang, S. Sannino, F. Managò, D. Scheggia, F. Papaleo, and V. Murino, "Automatic Visual Tracking and Social Behaviour Analysis with Multiple Mice", *PLoS ONE*, vol. 8, no. 9, E74557, 2013.

[49] J. M. Girard and J. F. Cohn, "A Primer on Observational Measurement", *Assessment*, vol. 23, no. 4, pp. 404–413, 2016.

[50] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[51] E. H. Goulding, A. K. Schenk, P. Juneja, A. W. MacKay, J. M. Wade, and L. H. Tecott, "A robust automated system elucidates mouse home cage behavioral structure", *Proceedings of the National Academy of Sciences*, vol. 105, no. 52, pp. 20 575–20 582, 2008.

[52] S. L. Handley and S. Mithani, "Effects of alpha-adrenoceptor agonists and antagonists in a maze-exploration model of 'fear'-motivated behaviour", *Naunyn-Schmiedeberg's Archives of Pharmacology*, vol. 327, no. 1, pp. 1–5, 1984.

[53] D. J. Heeren and A. R. Cools, "Classifying postures of freely moving rodents with the help of fourier descriptors and a neural network", *Behavior Research Methods, Instruments, & Computers*, vol. 32, no. 1, pp. 56–62, 2000.

[54] A. R. Hollenbeck, "Problems of reliability in observational research", *Observing behavior*, vol. 2, pp. 79–98, 1978.

[55] W. Hong, A. Kennedy, X. P. Burgos-Artizzu, M. Zelikowsky, S. G. Navonne, P. Perona, and D. J. Anderson, "Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning", *Proc. National Academy of Sciences*, vol. 112, no. 38, E5351–E5360, 2015.

[56] C. L. Howerton, J. P. Garner, and J. A. Mench, "A system utilizing radio frequency identification (RFID) technology to monitor individual rodent behavior in complex social settings", *Journal of Neuroscience Methods*, vol. 209, no. 1, pp. 74–78, 2012.

[57] J. B. Hoy, P. G. Koehler, and R. S. Patterson, "A microcomputer-based system for real-time analysis of animal movement", *Journal of Neuroscience Methods*, vol. 64, no. 2, pp. 157–161, 1996.

[58] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples", in *Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 892–900.

[59] H. Jhuang, E. Garrote, X. Yu, V. Khilnani, T. Poggio, A. D. Steele, and T. Serre, "Automated home-cage behavioural phenotyping of mice", *Nature Communications*, vol. 1, no. 6, pp. 1–9, 2010.

[60] J. A. Johansen, L. G. Clemens, and A. A. Nunez, "Characterization of Copulatory behavior in female mice: Evidence for paced mating", *Physiology & behavior*, vol. 95, no. 3, pp. 425–429, 2008.

[61] M. Kabra, A. A. Robie, M. Rivera-Alba, S. Branson, and K. Branson, "JAABA: Interactive machine learning for automatic annotation of animal behavior", *Nature Methods*, vol. 10, no. 1, pp. 64–67, 2012.

[62] N. Kafkafi, C. Mayo, D. Drai, I. Golani, and G. Elmer, "Natural segmentation of the locomotor behavior of drug-induced rats in a photobeam cage", *Journal of Neuroscience Methods*, vol. 109, no. 2, pp. 111–121, 2001.

[63] Z. Kalafatic, S. Ribaric, and V. Stanisavljevic, "A system for tracking laboratory animals based on optical flow and active contours", in *Proc. Conf. Image Analysis and Processing (ICIAP)*, 2001, pp. 334–339.

[64] A. V. Kalueff, A. M. Stewart, C. Song, K. C. Berridge, A. M. Graybiel, and J. C. Fentress, "Neurobiology of rodent self-grooming and its value for translational neuroscience", *Nature Reviews Neuroscience*, vol. 17, no. 1, pp. 45–59, 2016.

[65] A. Kelp, A. H. Koeppen, E. Petrasch-Parwez, C. Calaminus, C. Bauer, E. Portal, L. Yu-Taeger, B. Pichler, P. Bauer, O. Riess, and H. P. Nguyen, "A Novel Transgenic Rat Model for Spinocerebellar Ataxia Type 17 Recapitulates Neuropathological Changes and Supplies In Vivo Imaging Biomarkers", *Journal of Neuroscience*, vol. 33, no. 21, pp. 9068–9081, 2013.

[66] W. J. Kernan Jr., P. J. Mullenix, and D. L. Hopper, "Pattern recognition of rat behavior", *Pharmacology Biochemistry and Behavior*, vol. 27, no. 3, pp. 559–564, 1987.

[67] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets", *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 27, no. 11, pp. 1805–1819, 2005.

[68] S. Krackow, E. Vannoni, A. Codita, A. H. Mohammed, F. Cirulli, I. Branchi, E. Alleva, A. Reichelt, A. Willuweit, V. Voikar, G. Colacicco, D. P. Wolfer, J.-U. F. Buschmann, K. Safi, and H.-P. Lipp, "Consistent behavioral phenotype differences between inbred mouse strains in the IntelliCage", *Genes, Brain and Behavior*, vol. 9, no. 7, pp. 722–731, 2010.

[69] G. Kramida, N. A. Francis, C. M. Parameshwara, C. Fermüller, P. Kanold, and Y. Aloimonos, "Automated Mouse Behavior Recognition using VGG Features and LSTM Networks", in *Visual Observation and Analysis of Vertebrate And Insect Behavior Workshop (VAIB)*, 2016.

[70] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei, "The Unreasonable Effectiveness of Noisy Data for Fine-Grained Recognition", in *Proc. Conf. Computer Vision (ECCV)*, 2016, pp. 301–320.

[71] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[72] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection", in *Proc. Conf. Machine Learning (ICML)*, vol. 97, 1997, pp. 179–186.

[73] H. Kuehne, J. Gall, and T. Serre, "An end-to-end generative framework for video segmentation and recognition", in *Proc. Conf. Applications of Computer Vision (WACV)*, 2016, pp. 1–8.

[74] E. I. Kyriakou, G. Manfré, J. A. Spadaro, H. P. Nguyen, J. E. V. der Harst, and J. R. Homberg, "Anxiety and risk assessment-related traits in a rat model of Spinocerebellar ataxia type 17", *Behavioural Brain Research*, vol. 321, pp. 106–112, 2017.

[75] E. I. Kyriakou, H. P. Nguyen, J. R. Homberg, and J. E. Van der Harst, "Home-cage anxiety levels in a transgenic rat model for Spinocerebellar Ataxia type 17 measured by an approach-avoidance task: The light spot test", *Journal of Neuroscience Methods*, 2017.

[76] E. I. Kyriakou, J. G. van der Kieft, R. C. de Heer, A. Spink, H. P. Nguyen, J. R. Homberg, and J. E. van der Harst, "Automated quantitative analysis to assess motor function in different rat models of impaired coordination and ataxia", *Journal of Neuroscience Methods*, vol. 268, pp. 171–181, 2016.

[77] P. Laskov, C. Gehl, S. Krüger, and K.-R. Müller, "Incremental Support Vector Learning: Analysis, Implementation and Applications", *Journal of Machine Learning Research*, vol. 7, pp. 1909–1936, Sep 2006.

[78] Y. Lecun, L. Bottou, G. Orr, and K. Müller, "Efficient BackProp", in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, vol. 1524, Springer Verlag, 1998, pp. 9–50.

[79] P. N. Lehner, *Handbook of Ethological Methods*. Cambridge University Press, 1998, 672 pp.

[80] D. A. Levitis, W. Z. Lidicker Jr., and G. Freund, "Behavioural biologists do not agree on what constitutes behaviour", *Animal Behaviour*, vol. 78, no. 1, pp. 103–110, 2009.

[81] L. Lewejohann, A. M. Hoppmann, P. Kegel, M. Kritzler, A. Krüger, and N. Sachser, "Behavioral phenotyping of a murine model of Alzheimer's disease in a seminaturalistic environment using RFID tracking", *Behavior Research Methods*, vol. 41, no. 3, pp. 850–856, 2009.

[82] D. D. Lewis and J. Catlett, "Heterogeneous Uncertainty Sampling for Supervised Learning", in *Proc. Conf. Machine Learning (ICML)*, 1994, pp. 148–156.

[83] D. D. Lewis and W. A. Gale, "A Sequential Algorithm for Training Text Classifiers", in *Proc. Conf. Research and Development in Information Retrieval*, 1994, pp. 3–12.

[84] D. J. C. MacKay, "Information-Based Objective Functions for Active Data Selection", *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.

[85] G. Manfré, V. Doyère, S. Bossi, O. Riess, H. P. Nguyen, and N. El Massioui, "Impulsivity trait in the early symptomatic BACHD transgenic rat model of Huntington disease", *Behavioural Brain Research*, vol. 299, pp. 6–10, 2016.

[86] P. Martin and P. Bateson, *Measuring Behaviour: An Introductory Guide*, 2nd ed. Cambridge University Press, 1993.

[87] J. Matsumoto, H. Nishimaru, T. Ono, and H. Nishijo, "3D-Video-Based Computerized Behavioral Analysis for In Vivo Neuropharmacology and Neurophysiology in Rodents", in *In Vivo Neuropharmacology and Neurophysiology*, ser. Neuromethods, vol. 121, Springer New York, 2017, pp. 89–105.

[88] J. Matsumoto, S. Urakawa, Y. Takamura, R. Malcher-Lopes, E. Hori, C. Tomaz, T. Ono, and H. Nishijo, "A 3D-Video-Based Computerized Analysis of Social and Sexual Interactions in Rats", *PLoS ONE*, vol. 8, no. 10, e78460, 2013.

[89] M. Mayya and C. Doignon, "Visual tracking of small animals based on real-time Level Set Method with fast infra-red thermographic imaging", in *Proc. Conf. Robotic and Sensors Environments (ROSE)*, 2011, pp. 60–64.

[90] J. Monteiro, H. Oliveira, P. Aguiar, and J. Cardoso, "A depth-map approach for automatic mice behavior recognition", in *Proc. Conf. Image Processing (ICIP)*, 2014, pp. 2261–2265.

[91] J. Nadler, S. Moy, G. Dold, D. Trang, N. Simmons, A. Perez, N. Young, R. Barbaro, J. Piven, T. Magnuson, and J. Crawley, "Automated apparatus for quantitation of social approach behaviors in mice", *Genes, Brain and Behavior*, vol. 3, no. 5, pp. 303–314, 2004.

[92] L. P. J. J. Noldus, A. J. Spink, and R. A. J. Tegelenbosch, "EthoVision: A versatile video tracking system for automation of behavioral experiments", *Behavior Research Methods, Instruments, & Computers*, vol. 33, no. 3, pp. 398–414, 2001.

[93] S. Ohayon, O. Avni, A. L. Taylor, P. Perona, and S. E. R. Egnor, "Automated multi-day tracking of marked mice for the analysis of social behaviour", *Journal of Neuroscience Methods*, vol. 219, no. 1, pp. 10–19, 2013.

[94] D. Page, O. Kuti, and M. Sur, "Computerized assessment of social approach behavior in mouse", *Frontiers in Behavioral Neuroscience*, vol. 3, p. 48, 2009.

[95] S. J. Pan and Q. Yang, "A Survey on Transfer Learning", *Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[96] E. Parzen, "On Estimation of a Probability Density Function and Mode", *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[97] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2014, 552 pp.

[98] S. M. Pellis and T. J. Pasztor, "The developmental onset of a rudimentary form of play fighting in C57 mice", *Developmental Psychobiology*, vol. 34, no. 3, pp. 175–182, 1999.

[99] S. M. Pellis and V. C. Pellis, "Play fighting of rats in comparative perspective: A schema for neurobehavioral analyses", *Neuroscience & Biobehavioral Reviews*, vol. 23, no. 1, pp. 87–101, 1998.

[100] S. M. Pellis, V. C. Pellis, and D. A. Dewsbury, "Different levels of complexity in the play-fighting by muroid rodents appear to result from different levels of intensity of attack and defense", *Aggressive Behavior*, vol. 15, no. 4, pp. 297–310, 1989.

[101] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. de Polavieja, "idTracker: Tracking individuals in a group by automatic identification of unmarked animals", *Nature Methods*, vol. 11, no. 7, pp. 743–748, 2014.

[102] S. M. Peters, I. J. Pinter, H. H. J. Pothuizen, R. C. de Heer, J. E. van der Harst, and B. M. Spruijt, "Novel approach to automatically classify rat social behavior using a video tracking system", *Journal of Neuroscience Methods*, vol. 268, pp. 163–170, 2016.

[103] H. Pistori, V. V. Viana Aguiar Odakura, J. B. Oliveira Monteiro, W. N. Gonçalves, A. R. Roel, J. de Andrade Silva, and B. B. Machado, "Mice and larvae tracking using a particle filter with an auto-adjustable observation model", *Pattern Recognition Letters*, vol. 31, no. 4, pp. 337–346, 2010.

[104] M. Pratte and M. Jamon, "Detection of social approach in inbred mice", *Behavioural Brain Research*, vol. 203, no. 1, pp. 54–64, 2009.

[105] V. Quera, R. Bakeman, and A. Gnisci, "Observer agreement for event sequences: Methods and software for sequence alignment and reliability estimates", *Behavior Research Methods*, vol. 39, no. 1, pp. 39–49, 2007.

[106] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[107] H. Raghavan, O. Madani, and R. Jones, "Active Learning with Feedback on Features and Instances", *Journal of Machine Learning Research*, vol. 7, pp. 1655–1686, Aug 2006.

[108] Z. Ren, A. Noronha, A. V. Ciernia, and Y. J. Lee, "Who Moved My Cheese? Automatic Annotation of Rodent Behaviors with Convolutional Neural Networks", in *Proc. Winter Conf. Applications of Computer Vision (WACV)*, 2017.

[109] C. A. Richardson, "The power of automated behavioural homecage technologies in characterizing disease progression in laboratory mice: A review", *Applied Animal Behaviour Science*, vol. 163, pp. 19–27, 2015.

[110] S. H. Richter, J. P. Garner, and H. Würbel, "Environmental standardization: Cure or cause of poor reproducibility in animal experiments?", *Nature Methods*, vol. 6, no. 4, pp. 257–261, 2009.

[111] S. H. Richter, J. P. Garner, B. Zipser, *et al.*, "Effect of Population Heterogenization on the Reproducibility of Mouse Behavior: A Multi-Laboratory Study", *PLOS ONE*, vol. 6, no. 1, e16461, 2011.

[112] R. Rifkin and A. Klautau, "In Defense of One-Vs-All Classification", *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[113] A. A. Robie, K. M. Seagraves, S. E. R. Egnor, and K. Branson, "Machine vision methods for analyzing social interactions", *Journal of Experimental Biology*, vol. 220, no. 1, pp. 25–34, 2017.

[114] R. Rosenthal and D. B. Rubin, "Interpersonal expectancy effects: The first 345 studies", *Behavioral and Brain Sciences*, vol. 1, no. 3, pp. 377–386, 1978.

[115] J. B. I. Rousseau, P. B. A. Van Lochem, W. H. Gispen, and B. M. Spruijt, "Classification of rat behavior with an image-processing method and a neural network", *Behavior Research Methods, Instruments, & Computers*, vol. 32, no. 1, pp. 63–71, 2000.

[116] N. Roy and A. McCallum, "Toward Optimal Active Learning Through Sampling Estimation of Error Reduction", in *Proc. Conf. Machine Learning (ICML)*, 2001, pp. 441–448.

[117] O. Rudenko, V. Tkach, V. Berezin, and E. Bock, "Detection of early behavioral markers of Huntington's disease in R6/2 mice employing an automated social home cage", *Behavioural Brain Research*, vol. 203, no. 2, pp. 188–199, 2009.

[118] W. M. S. Russell and R. L. Burch, *The Principles of Humane Experimental Technique*. Methuen, 1959.

[119] G. H. Salem, J. U. Dennis, J. Krynitsky, M. Garmendia-Cedillos, K. Swaroop, J. D. Malley, S. Pajevic, L. Abuhatzira, M. Bustin, J.-P. Gillet, M. M. Gottesman, J. B. Mitchell, and T. J. Pohida, "SCORHE: A novel and practical approach to video monitoring of laboratory mice housed in vivarium cage racks", *Behavior Research Methods*, vol. 47, no. 1, pp. 235–250, 2015.

[120] G. Salem, J. Krynitsky, T. Pohida, M. Hayes, and X. P. Burgos-Artizzu, "Three Dimensional Pose Estimation of Mouse from Monocular Images in Compact Systems", in *Proc. Conf. Pattern Recognition (ICPR)*, 2016, pp. 1751–1756.

[121] F. Sams-Dodd, "Automation of the social interaction test by a video-tracking system: Behavioural effects of repeated phencyclidine treatment", *Journal of Neuroscience Methods*, vol. 59, no. 2, pp. 157–167, 1995.

[122] J. Schneider and J. D. Levine, "Automated identification of social interaction criteria in Drosophila melanogaster", *Biology Letters*, vol. 10, no. 10, E20140749, 2014.

[123] I. K. Sethi and R. Jain, "Finding Trajectories of Feature Points in a Monocular Image Sequence", *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 9, no. 1, pp. 56–73, 1987.

[124] B. Settles, "From theories to queries: Active learning in practice", in *Proc. Workshop on Active Learning and Experimental Design*, 2011, pp. 1–18.

[125] B. Settles, "Active learning literature survey", University of Wisconsin-Madison, Technical Report 1648, 2010, p. 11.

[126] ——, "Closing the Loop: Fast, Interactive Semi-supervised Annotation with Queries on Features and Instances", in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2011, pp. 1467–1478.

[127] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning", in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 1289–1296.

[128] A. L. Sheets, P.-L. Lai, L. C. Fisher, and D. M. Basso, "Quantitative Evaluation of 3D Mouse Behaviors and Motor Function in the Open-Field after Spinal Cord Injury Using Markerless Motion Tracking", *PLOS ONE*, vol. 8, no. 9, e74536, 2013.

[129] Y. Shemesh, Y. Sztainberg, O. Forkosh, T. Shlapobersky, A. Chen, and E. Schneidman, "High-order social interactions in groups of mice", *eLife*, vol. 2, E00759, 2013.

[130] J. L. Silverman, M. Yang, C. Lord, and J. N. Crawley, "Behavioural phenotyping assays for mouse models of autism", *Nature Reviews Neuroscience*, vol. 11, no. 7, pp. 490–502, 2010.

[131] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos", in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 568–576.

[132] ——, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *CoRR*, vol. abs/1409.1556, 2014.

[133] M. B. Sokolowski and D. Wahlsten, "Gene-environment interaction and complex behavior", in *Methods in Genomic Neuroscience*, ser. Frontiers in Neuroscience, CRC Press, 2001, pp. 3–27.

[134] B. M. Spruijt and W. H. Gispen, "Prolonged animal observation by use of digitized videodisplays", *Pharmacology Biochemistry and Behavior*, vol. 19, no. 5, pp. 765–769, 1983.

[135] B. M. Spruijt, T. Hol, and J. Rousseau, "Approach, avoidance, and contact behavior of individually recognized animals automatically quantified with an imaging technique", *Physiology & Behavior*, vol. 51, no. 4, pp. 747–752, 1992.

[136] B. M. Spruijt, S. M. Peters, R. C. de Heer, H. H. Pothuizen, and J. E. van der Harst, "Reproducibility and relevance of future behavioral sciences should benefit from a cross fertilization of past recommendations and today's technology: "Back to the future"", *Journal of Neuroscience Methods*, vol. 234, pp. 2–12, 2014.

[137] M. L. Terranova and G. Laviola, "Scoring of social interactions and play in mice during adolescence", *Current Protocols in Toxicology*, pp. 1–11, 13.10 2005.

[138] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks", in *Proc. Conf. Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[139] V. Trezza, P. J. Baarendse, and L. J. Vanderschuren, "The pleasures of play: Pharmacological insights into social reward mechanisms", *Trends in Pharmacological Sciences*, vol. 31, no. 10, pp. 463–469, 2010.

[140] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large Margin Methods for Structured and Interdependent Output Variables", *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[141] C. J. Twining, C. J. Taylor, and P. Courtney, "Robust tracking and posture description for laboratory rodents using active shape models", *Behavior Research Methods, Instruments, & Computers*, vol. 33, no. 3, pp. 381–391, 2001.

[142] J. Uhrich, "The social hierarchy in albino mice", *Journal of Comparative Psychology*, vol. 25, no. 2, pp. 373–413, 1938.

[143] J. Unger, M. Mansour, M. Kopaczka, N. Gronloh, M. Spehr, and D. Merhof, "An unsupervised learning approach for tracking mice in an enclosed area", *BMC Bioinformatics*, vol. 18, p. 272, 2017.

[144] C. L. Van Den Berg, J. M. Van Ree, and B. M. Spruijt, "Sequential analysis of juvenile isolation-induced decreased social behavior in the adult rat", *Physiology & Behavior*, vol. 67, no. 4, pp. 483–488, 1999.

[145] E. A. van Dam, J. E. van der Harst, C. J. F. ter Braak, R. A. J. Tegelenbosch, B. M. Spruijt, and L. P. J. J. Noldus, "An automated system for the recognition of various specific rat behaviours", *Journal of Neuroscience Methods*, vol. 218, no. 2, pp. 214–224, 2013.

[146] D. Wahlsten, P. Metten, T. J. Phillips, *et al.*, "Different data from different labs: Lessons from studies of gene-environment interaction", *Journal of Neurobiology*, vol. 54, no. 1, pp. 283–311, 2003.

[147] A. Weissbrod, A. Shapiro, G. Vasserman, L. Edry, M. Dayan, A. Yitzhaky, L. Hertzberg, O. Feinerman, and T. Kimchi, "Automated long-term tracking and social behavioural phenotyping of animal colonies within a semi-natural environment", *Nature Communications*, vol. 4, E2018, 2013.

[148] I. Q. Whishaw, F. Haun, and B. Kolb, "Analysis of Behavior in Laboratory Rodents", in *Modern Techniques in Neuroscience Research*, Springer, 1999, pp. 1243–1275.

[149] A. B. Wiltschko, M. J. Johnson, G. Iurilli, R. E. Peterson, J. M. Katon, S. L. Pashkovski, V. E. Abraira, R. P. Adams, and S. R. Datta, "Mapping Sub-Second Structure in Mouse Behavior", *Neuron*, vol. 88, no. 6, pp. 1121–1135, 2015.

[150] Y. Winter and A. T. U. Schaefers, "A sorting system with automated gates permits individual operant experiments with mice from a social home cage", *Journal of Neuroscience Methods*, vol. 196, no. 2, pp. 276–280, 2011.

[151] H. Würbel, "Behavioral phenotyping enhanced – beyond (environmental) standardization", *Genes, Brain and Behavior*, vol. 1, no. 1, pp. 3–8, 2002.

[152] R. Yan, J. Yang, and A. Hauptmann, "Automatically Labeling Video Data Using Multi-class Active Learning", in *Proc. Conf. Computer Vision (ICCV)*, 2003, pp. 516–523.

[153] T.-H. Ou-Yang, M.-L. Tsai, C.-T. Yen, and T.-T. Lin, "An infrared range camera-based approach for three-dimensional locomotion tracking and pose reconstruction in a rodent", *Journal of Neuroscience Methods*, vol. 201, no. 1, pp. 116–123, 2011.

[154] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting", in *Proc. Conf. Data Mining (ICDM)*, 2003, pp. 435–442.

[155] T. Zhang, "Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms", in *Proc. Conf. Machine Learning (ICML)*, 2004, pp. 919–926.

[156] J. B. Zurn, X. Jiang, and Y. Motai, "Video-based tracking and incremental learning applied to rodent behavioral activity under near-infrared illumination", *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 6, pp. 2804–2813, 2007.

List of Publications

Parts of this thesis appear in the following publications:

M. Lorbach, R. Poppe, E. A. van Dam, L. P. J. J. Noldus, and R. C. Veltkamp, "Automated Recognition of Social Behavior in Rats: The Role of Feature Quality", in *Proc. Conf. Image Analysis and Processing*, 2015, pp. 565–574.

——, "Clustering-based Active Learning in Unbalanced Rodent Behavior Data", in *Proc. Visual Observation and Analysis of Vertebrate And Insect Behavior Workshop*, 2016.

——, "Transfer Learning for Rodent Behavior Recognition", in *Proc. Conf. Measuring Behavior*, 2016, pp. 461–469.

M. Lorbach, E. I. Kyriakou, R. Poppe, E. A. van Dam, L. P. J. J. Noldus, and R. C. Veltkamp, "Learning to Recognize Rat Social Behavior: Novel Dataset and Cross-Dataset Application", *Journal of Neuroscience Methods*, 2017, in press.

Other publications not related to this thesis:

M. Lorbach, S. Höfer, and O. Brock, "Prior-assisted propagation of spatial information for object search", in *Proc. Conf. Intelligent Robots and Systems (IROS)*, 2014, pp. 2904–2909.

R. Martin-Martin, M. Lorbach, and O. Brock, "Deterioration of depth measurements due to interference of multiple RGB-D sensors", in *Proc. Conf. Intelligent Robots and Systems (IROS)*, 2014, pp. 4205–4212.

Acknowledgments

About three and a half years ago, in April 2014, I started wandering along a trajectory scattered with interactions, solitary stretches, twists and turns, side tracks and dead ends. Now and then, other people would come and join me. Some left again later on; others stuck around until the end. The following is dedicated to all those people.

Let us start at the beginning of the journey. Lucas and Hoa paved the first meters and brought the PhenoRat project to life. Not only did they make this research possible, they also managed to avoid treacherous abysses so we could keep going. Thanks, Lucas in particular, for the freedom I have been given in finding my way in an entangled subject; for supporting me even beyond the PhenoRat track. Thanks for all the valuable feedback given and concerns shared. The same goes for Remco, thank you for being there whenever needed, along with pulling the strings in the background. I am very grateful that the path beyond PhenoRat continued smoothly.

One of the first persons who joined my path was Elsbeth. I deeply appreciate her supervision, her insights and guidance, and our countless discussions. Thank you for bringing me back on track when I wandered off with one of my unfathomable, vague ideas. Similarly, thanks, Rob, for sharing decades of experience in rodent tracking and for letting me try even unpolished algorithms.

Although our paths were not always as neatly aligned as they may have been in an ideal world, it would have been a much longer and more difficult road without the other PhenoRat fellows. Thank you, Elisavet, Giuseppe, David and Oscar, for being great comrades in arms. I have learned a lot from working in an interdisciplinary project. Despite the many pitfalls and hurdles, what prevails are the positive experiences I gained by thinking outside the box, or perhaps rather outside the cage.

Suzanne and Johanneke were invaluable resources for all my questions about rats, what they do and why. Thank you both for introducing me to the big issues and the small riddles in ethology and behavioral neuroscience. Special thanks go to Suzanne who shared hours of video recordings and annotations with me and helped me get started in the first place.

He missed the start but stayed until the very end. I have to thank Ronald for uncountable things within university walls and far beyond. I am lucky that I was able to enjoy his daily supervision and I am grateful for the relentless comments in writing and in person. Thanks for thinking along; for

making me stick to deadlines even if we set them ourselves. For letting us use the boat and for liberating it from the firm grip of the canal vegetation; and finally for the vigorous attempts to expand the quaint corners of my Dutch vocabulary. Ronald, zonder jouw puike inzet en voortdurende gezanik was het mettertijd broddelwerk geworden. Hartelijk bedankt!

I would also like to thank everyone else who offered help, support, advice or wisdom at any point in time. Most did so willingly and some without even realizing. Thanks, Carola for standing on our toes whenever it was really necessary and otherwise protecting us from EU bureaucracy. Thanks, Nina, for all the cakes and muffins that were neither virtual nor augmented but truly delicious in reality. Thanks to the unofficial Noldus table football team for the most entertaining lunch breaks. Thanks to the Moving Beyond fellows for making training weeks so much more enjoyable. Thanks to former and current co-workers and colleagues. Thanks to all participants of the user study and finally, thanks to all members of the dissertation committee for taking the time to read this thesis and providing very valuable and constructive feedback.

Some people have been there for me for much longer than three and a half years. I am very thankful to my family, and my parents in particular, for supporting me, for giving me the freedom to make my own decisions and for helping me if it turned out to be a bad one. Luckily, they taught me how to avoid bad ones, so most of them were pretty okay.

In all places I lived, studied or worked, I made friends who I had to leave behind eventually. Even if we don't see each other very often or even lost contact altogether, every one of them means a great deal to me. I know you are there when I need you to be. This is a very comforting thought.

Speaking of comfort, I thank all my Ultimate Frisbee companions for providing lots of positive spirit and keeping my life in balance. They provided me with perspective and just the right amount of distraction.

Finally, there is one person I can't thank enough. Freke, I could call on you at any time to share thoughts, ideas and despair. In return I got understanding, inspiration, and love. If I had any considerable knowledge of Greek mythology, I might call you my muse. But I haven't, so I won't. Thank you for walking at my side the whole time.

I gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for parts of this research.

Curriculum Vitae

Malte Lorbach was born on June 2, 1986 in Osterode am Harz, Germany. After school he moved to Bremen to study Technische Informatik (Computer Engineering) at Hochschule Bremen. In an exchange program with London South Bank University, he spent one year in the United Kingdom, where he continued the study in Electrical and Electronic Engineering. In 2010 he graduated with a dual Bachelor degree, B.Sc. & B.Eng. (Hons), from both universities. Malte then followed a Master's course in Computer Science at Technische Universität Berlin, in which he focused on Computer Vision and Robotics. He completed his Master's thesis in 2014 in the Robotics & Biology Laboratory dealing with spatial inference for object search. Working on an inference problem raised his interest in Pattern Recognition with the result that, after graduation, he started his doctoral research on video-based behavior recognition. He conducted this research at Noldus Information Technology in Wageningen, the Netherlands and Utrecht University. The results are presented in this thesis.

A

Appendix

Table A.1: Behavior definitions used in RatSI and inter-annotator agreement study.

Behavior	Description
Following	A following animal attempts to maintain a close distance to another animal while the latter is moving. This behavior includes interactions in which the following animal is (partly) on top of the other animal. Chasing bouts are also scored as following. During chasing, the two animals run after each other in close contact and their path often describes an S- or 8-like shape. A following bout starts when both animals are in motion. The bout ends if either of the animals stops moving, the following animal leaves the path of the other, or its velocity gets much smaller than the velocity of the other. An approaching animal tries to get into proximity of another animal. The two animals are not in proximity yet and one animal moves in a direct way towards the other until (near-) contact is established.
Approaching	The approached animal might be stationary or moving. However, once both animals are in proximity and keep moving in the same direction, this is scored as following. The approach bout starts with a clear change of direction and/or increase of velocity towards the other animal and ends with establishing (near-)contact or another change of direction/velocity away from the other (aborting the approach).
Moving Away	One animal is actively and quickly moving away from another animal after being in close proximity. Its movement is directed away from the other animal, i.e., their distance would increase, and is performed faster than with walking speed. A moving away bout is initiated by a high acceleration with a movement direction away from the other animal. It ends once the velocity decreases again or the direction is not away from the other anymore.
Allogrooming	The grooming animal has one or both front paws on the other animal and pulls repeatedly at its fur. The head of the grooming animal often makes nodding-like movements. Both animals remain stationary, and the front part of the active animal can be on top of the other animal during grooming. The more "aggressive" kind of allogrooming in which the groomed animal attempts to run away but fails (the other keeps up) is not considered within this category. It will be scored as following. The start of an allogrooming bout is marked by the first clear nodding of the head. The bout ends with the separation of the animals or a clear repositioning and the engagement in another action.

Continued on next page...

Table A.1: Behavior definitions used in RatSI and inter-annotator agreement study.

Behavior	Description
Nape Attacking	One animal approaches or attacks the neck area of another with its front part of the body, i.e., its paws and head. The attack can occur while both animals are moving (e.g., from within a following bout or subsequent to an approach bout). Most of the time a nape attack is short. It starts with a clear approach of the neck by a short movement of the head towards the neck of the other. This movement can be a head-turn, a leap, or a translational motion. The bout does not start earlier than a few frames before contact (approx. 120-150 ms) and therefore does not include approaching the other animal from a distance (this part would be scored as approaching). The bout ends when the contact between head and attack point (i.e., the neck area) is lost or when the attacking animal engages in another interaction like pinning, allogrooming, or following.
Pinning	Pinning is one possible follow-up interaction after a nape attack. The attacked animal may turn on its back. The attacker then pins the other animal to the ground by pushing it down with its forepaws or its whole body. It actively tries to keep the other on its back. Pinning sometimes evolves into allogrooming. Since there is no clear transition point, we give the higher priority to allogrooming, i.e., in doubt we score it as allogrooming. The pinning bout starts when the attacked animal starts rotating on its back and the attacker is on top of the other. The bout ends if the attacker moves down from the other animal or the latter rotates back on its feet.
Social Nose Contact	One individual establishes contact or near-contact with its nose to another's body parts. Both animals remain mainly stationary during this interaction although the active animal might move around the partner's body. Social nose contact comprises two forms: social sniffing and anogenital inspection. Scoring of this interaction starts not earlier than 2-3 frames before the contact is established. That is, the approach, if occurring, is not scored as social nose contact. Scoring ends when the contact with the nose is lost (e.g., when the animal turns away).
Solitary	The animals are not actively interacting or perform individual actions such as self-grooming, rearing or exploration. The animals are allowed to be in close proximity as long as their actions are not oriented towards the other animal.
Other	An animal actively interacts with the other animal but the interaction is not defined. Examples for not defined interactions (that may occur in the videos shown) are: boxing, kicking, crawling over/under each other or wrestling.
Unknown	An animal actively interacts with the other animal but the interaction is not distinguishable. You may score "unknown" if you are uncertain which of two defined behaviors you see. You may leave a comment about what you are uncertain about in the comments field.