



**CZECH TECHNICAL
UNIVERSITY
IN PRAGUE**

F3

**Faculty of Electrical Engineering
Department of Circuit Theory**

Master's Thesis

Automatic Classification of Social Interactions of Rats from Video

Bc. Fadi Kanout

Medical Electronics and Bioinformatics

January 2024

Supervisor: RNDr. David Levčík Ph.D., prof. Dr. Ing. Jan Kybic

I. Personal and study details

Student's name: **Kanout Fádi** Personal ID number: **474585**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Circuit Theory**
Study program: **Medical Electronics and Bioinformatics**
Specialisation: **Signal processing**

II. Master's thesis details

Master's thesis title in English:

Automatic Classification of Social Interactions of Rats from Video

Master's thesis title in Czech:

Automatické rozpoznání sociálních interakcí potkan z videosekvencí

Guidelines:

1. Assemble a multi-camera hardware and software system for recording the activities of laboratory rats.
2. Get familiar with:
 - a. existing approaches for activity recognition from video (via research),
 - b. existing relevant software (such as DeepLabCut, Argus, DANNCE, OpenCV, etc.), and
 - c. existing annotated datasets of animal video sequences (AcinoSet, PAIR-R24M, etc.).
3. Define key points and types of behavior of individual animals and relevant social interactions for particular datasets. Check the availability of annotations for the proposed key points and types of behavior in the datasets and supplement any missing annotations.
4. Assemble a software system for automatic recognition of social interactions using existing software modules for key point detection, correspondence finding in time and space, 3D reconstruction, feature evaluation and classification using features. Develop missing modules.
5. Experimentally verify the functionality of the created system and individual modules both qualitatively and quantitatively, compare the functionality of different methods and compare with results from literature.
6. Identify the main weaknesses of the created system and propose, implement and experimentally verify suitable improvements.
7. Optionally: Use the developed system to recognize social interactions in transgenic TgF344-AD rats, an animal model of Alzheimer's disease, and try to identify differences compared to the control group.

Bibliography / sources:

- [1] Lauer J, Zhou M, Ye S, Menegas W, Schneider S, Nath T, Rahman MM, Di Santo V, Soberanes D, Feng G, Murthy VN, Lauder G, Dulac C, Mathis MW, Mathis A. Multi-animal pose estimation, identification and tracking with DeepLabCut. Nat Methods. 2022 Apr;19(4):496-504. doi: 10.1038/s41592-022-01443-0. Epub 2022 Apr 12. PMID: 35414125; PMCID: PMC9007739.
- [2] Mathis A, Mamidanna P, Cury KM, Abe T, Murthy VN, Mathis MW, Bethge M. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. Nat Neurosci. 2018 Sep;21(9):1281-1289. doi: 10.1038/s41593-018-0209-y. Epub 2018 Aug 20. PMID: 30127430.
- [3] Joska D, Clark L, Muramatsu N, Jericevich R, Nicolls F, Mathis A, W. Mathis M, Patel A. AcinoSet: A 3D Pose Estimation Dataset and Baseline Models for Cheetahs in the Wild," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 13901-13908, doi: 10.1109/ICRA48506.2021.9561338.
- [4] Shams S, Amlani S, Scicluna M, Gerlai R. Argus: An open-source and flexible software application for automated quantification of behavior during social interaction in adult zebrafish. Behav Res Methods. 2019 Apr;51(2):727-746. doi: 10.3758/s13428-018-1083-y. PMID: 30105442.
- [5] Marshall JD, Klibaite U, Gellis A, Aldarondo DE, Ölveczky BP, Dunn T. The PAIR-R24M Dataset for Multi-animal 3D Pose Estimation. NeurIPS. 2021. doi: 10.6084/m9.figshare.17032895.v1

Name and workplace of master's thesis supervisor:

RNDr. David Lev ík, Ph.D. Institute of Physiology CAS, Prague

Name and workplace of second master's thesis supervisor or consultant:

prof. Dr. Ing. Jan Kybic Biomedical imaging algorithms FEE

Date of master's thesis assignment: **01.02.2023** Deadline for master's thesis submission: **09.01.2024**

Assignment valid until: **22.09.2024**

RNDr. David Lev ík, Ph.D.
Supervisor's signature

doc. Ing. Radoslav Bortel, Ph.D.
Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

Acknowledgement / Declaration

I would like to express my deepest appreciation to my supervisors, RNDr. David Levčík, Ph.D., and Prof. Dr. Ing. Jan Kybic. Their invaluable guidance, patience, and ability to provide insightful solutions to any question or challenge have significantly shaped my work, thinking and experience.

I am also profoundly grateful to my family and friends for their unwavering support and encouragement throughout my studies.

I hereby declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university thesis.

In Prague, January 9, 2024.

.....

Abstrakt / Abstract

Alzheimerova choroba (ACH) je nejčastější příčinou demence. Narušení sociálního chování je často časným příznakem ACH, který se typicky objevuje před nástupem poruch v oblasti kognitivních funkcí.

Hlavním cílem této práce je vytvoření algoritmů pro automatickou detekci a klasifikaci různých typů specifických sociálních interakcí u transgenních potkanů TgF344-AD - animálního modelu Alzheimerovy choroby. Potkani byli pozorováni pomocí kamer z více úhlů, což umožnilo sledovat pohyb potkanů v trojrozměrném souřadnicovém systému. Potkani byli pozorováni během testu sociální interakce, při němž se neznámí jedinci primárně zkoumají očicháváním anogenitální oblasti, hlavy nebo zbytku těla v uzavřené aréně.

Práce prezentuje implementaci vhodných metod rozpoznávání akcí, vytvoření datových sad obsahujících akce potkanů, využití softwaru jako je DeepLabCut pro získání polohy potkanů a využití více kamerových pohledů pro zlepšení odhadu polohy a sledování identity potkanů. Naše metody spolehlivě klasifikovaly akce potkanů a, pokud to bylo platné, iniciátora akce. Práce také zahrnuje statistickou analýzu pro porovnání sociálních interakcí u potkanů TgF344-AD a F344 (kontrolní skupina), s využitím klasifikace z vyvinutých metod na získaných videozáznamech. Nenašli jsme žádné významné rozdíly mezi TgF344-AD a F344 ve vybraných akcích.

Klíčová slova: Rozpoznávání akcí, Alzheimerova choroba, počítačové vidění, hluboké učení, odhad polohy.

Alzheimer's disease (AD) is the most common cause of dementia. Disruption in social behavior is often an early symptom of AD, typically appearing before the onset of cognitive domain symptoms.

The main focus of this work is the creation of algorithms for the automatic detection and classification of various types of specific social interactions in transgenic TgF344-AD rats - an animal model of Alzheimer's disease. The rats were observed using cameras from multiple views, which allowed the assessment of the rat's motion in a 3D coordinate frame. The rats were observed during a social interaction test, in which unfamiliar individuals primarily examine each other by sniffing the anogenital area, head, or the rest of the body in a closed arena.

The work presents the implementation of suitable action recognition methods, the creation of datasets comprehending the rat's actions, utilizing software such as DeepLabCut to obtain the pose of the rats, and utilizing the multiple camera views to improve the rat's estimated pose and identity tracking. Our methods managed to classify the rat's actions reliably and, if applicable, the initiator of the action. The work also includes a statistical analysis to compare the social interactions in TgF344-AD and F344 rats (control group), using the predictions from the developed methods on the obtained video footage. We did not find any significant differences between the TgF344-AD and F344 in denoted actions.

Keywords: Action recognition, Alzheimer's disease, Computer vision, Deep learning, Pose estimation.

Contents /


1 Introduction	1		
1.1 Alzheimer’s Disease	1		
1.1.1 AD and Social behavior	2		
1.1.2 Forms of AD	2		
1.1.3 Animal Models	2		
1.1.4 Transgenic Model TgF344-AD	2		
1.2 Pose Estimation	3		
1.3 Tracking	4		
1.4 Action Recognition	5		
1.5 3D Reconstruction	6		
2 Data	7		
2.1 Hardware Setup	7		
2.2 Camera Setup	7		
2.3 Animals	8		
2.4 Experiment Setup	8		
2.5 Camera Calibration	9		
2.6 Data Acquisition	10		
2.7 Video analysis	10		
2.8 Data Pre-processing	11		
2.9 3D Coordinate Reconstruction from Multi-View 2D Projections	11		
2.10 3D Coordinate Spatial Imputation	12		
2.10.1 Point Distribution Model for 3D coordinate Imputation	12		
2.10.2 Geometric Rule-Based Imputation	14		
2.11 3D Coordinate Temporal Imputation	16		
2.12 Outlier Detection	17		
2.13 Datasets	18		
2.13.1 Custom Datasets	18		
2.13.2 PAIR-R24M Dataset	20		
2.13.3 DLC Dataset	21		
3 Methods	22		
3.1 Pose Estimation and Tracking in 3D	23		
3.2 Action Recognition	24		
3.2.1 Rule-Based Model	24		
3.2.2 Rule-Based Model Label Adaptation	28		
3.3 Action recognition based on Deep Learning Models	28		
3.3.1 ST GCN Model	29		
3.3.2 Coordinate Data Augmentation	31		
3.3.3 TSRJI-CNN Model	32		
3.3.4 Pivot TSRJI-CNN Model	34		
3.3.5 CNN-LSTM Model	34		
3.3.6 View Image Data Augmentation	35		
3.3.7 3D CNN Model	35		
3.3.8 1D CNN Model	35		
3.3.9 Multi-Modal Model	36		
3.3.10 Action Initiator classification	36		
3.3.11 Human Predictions	36		
3.4 Data Analysis	37		
3.5 Training protocol	38		
4 Experiments	39		
4.1 Evaluation of DLC Pose Estimation	39		
4.2 Point Distribution Model (PDM)	40		
4.3 Comparison of PDM Imputation with PDM and Geometric Rule-Based Imputation	44		
4.4 Temporal Imputation method	46		
4.5 Identity Swap correction	46		
4.6 Rule-Based Model Evaluation	47		
4.7 Rule-Based Model – TgF344AD Pair 18 adaptation Evaluation	49		
4.8 Experiments: Deep Models Evaluation	50		
4.9 ST GCN Model Training and Evaluation	51		
4.10 TSRJI-CNN Model Training and Evaluation	52		
4.11 Pivot TSRJI-CNN Model Training and Evaluation	54		
4.12 CNN-LSTM Model Training and Validation	55		
4.13 3D CNN Model Training and Validation	57		

4.14	1D CNN Model Training and Validation	58
4.15	MultiModal Model Training and Validation	59
4.16	MultiModal Model Configuration c_3 Training and Validation	60
4.17	Model Comparison	61
4.18	Action Initiator Classification .	62
4.19	Comparison of human expert classification with an automatic method	63
4.19.1	Visual inspection of MMc_3 model predictions .	65
5	Data Analysis Results	66
5.1	Dataset TgF344 AD Pair 18 MMc_3 model action predictions Analysis Results	66
5.2	Dataset TgF344 AD Pair 18 Rule-Based Model action predictions Analysis Results . .	69
6	Conclusion	71
6.1	Future Work	71
	References	72
A	DeepLabCut (DLC) Project configuration	77
B	Contents of attachment, Shortcuts	78
B.1	Contents of attachment	78
B.2	Shortcuts	78

Tables / Figures

2.1	Habituation Rat Pairings Table	9	2.1	D435 Quad View Camera Setup	8
2.2	Interaction Rat Pairings Table ..	9	2.2	PDM Alignment.....	14
2.3	Keypoint Descriptions	10	2.4	Rule-Based Imputation Depiction	15
4.1	RMSE on Training Dataset....	40	2.3	Rule-Based Imputation Systematic Search	16
4.2	RMSE on Testing Dataset.....	40	2.6	View removal correction	17
4.3	Rule-Based Model Class Performance Metrics	47	2.5	DLC Wrongly assigned markers	18
4.4	Rule-Based Model Thresholds .	49	2.7	Social Action sequences examples 1	20
4.5	Rule-Based Model Adaptation Class Performance Metrics	49	2.8	Social Action sequences examples 2	20
4.6	Rule-Based Model Adaptation Thresholds.....	50	3.1	Corrected Sequence acquisition diagram.....	23
4.7	ST GCN Class Performance Metrics on TgF344AD Pair 18	51	3.2	Example of the approach rule in Rule-Based Model.....	27
4.8	ST GCN Class Performance Metrics on Pair 24-M Dataset .	52	3.3	GCN Graph Topologies	30
4.9	TSRJI-CNN Class Performance Metrics on TgF344AD Pair 18.....	53	3.4	ST GCN Architecture.....	31
4.10	Pivot TSRJI-CNN Class Performance Metrics on TgF344AD Pair 18	54	3.5	TSRJI RGB representation....	33
4.11	CNN-LSTM Class Performance Metrics on TgF344AD Pair 18.....	56	3.6	TSRJI-CNN Architecture.....	33
4.12	3D CNN Model Class Performance Metrics on TgF344AD Pair 18.....	57	3.7	CNN Image representation	35
4.13	1D CNN Model Class Performance Metrics on TgF344AD Pair 18.....	58	4.1	DLC RMSE and likelihood Comparisson.....	39
4.14	MultiModal Models Class Performance Metrics on TgF344AD Pair 18	59	4.2	PDM Estimation Depiction (3 and 4 missing markers)	41
4.15	MultiModal Model configuration c_3 Class Performance Metrics on TgF344AD Pair 18	60	4.3	PDM Estimation Depiction (5 and 6 missing markers)	42
4.16	Model Comparison Validation F_1 Score and Accuracy ...	62	4.4	MSE Evaluation of PDM	43
4.17	TSRJI-CNN Initiator Models Performance Metrics	63	4.5	PDM Estimation compared with Geometric Imputation....	44
			4.6	PDM Estimation compared with Geometric Imputation Scatter Plot	45
			4.7	PDM Estimation compared with Geometric Imputation Visualisation.....	45
			4.8	3D coordinate Temporal Imputation methods MSE	46
			4.9	Identity Swap correction Experiment	47
			4.10	Rule-Based Model evaluation..	48
			4.11	Rule-Based Model Confusion Matrix	48

4.18	Human and Model Predictions Accuracy Comparison....	64
4.19	Human and Model Predictions Class Count Comparison	64
5.1	Table: P-values from Kruskal-Wallis Test	67
5.2	Table: P-values from Mann-Whitney U Test	69
4.12	Rule-Based Model Adaptation evaluation	50
4.13	ST GCN Models evaluation on TgF344AD Pair 18	52
4.14	ST GCN Models evaluation on Pair R24-M	52
4.15	TSRJI-CNN Model evaluation on TgF344AD Pair 18	53
4.16	TSRJI-CNN Model Metrics evaluation on TgF344AD Pair 18.....	54
4.17	Pivot TSRJI-CNN Model Metrics evaluation on TgF344AD Pair 18	55
4.18	CNN-LSTM Model evaluation on TgF344AD Pair 18	56
4.19	CNN-LSTM Model Metrics evaluation on TgF344AD Pair 18.....	57
4.20	3D CNN Model Metrics evaluation on TgF344AD Pair 18	58
4.21	1D CNN Model Metrics evaluation on TgF344AD Pair 18	59
4.22	MultiModal Models evaluation on TgF344AD Pair 18	60
4.23	MultiModal Model configuration c_3 Metrics evaluation on TgF344AD Pair 18	61
4.24	MultiModal Model configuration c_3 Validation Confusion Matrix on TgF344AD Pair 18.....	61
4.25	Model Comparison Validation F_1 Score	62
4.26	TSRJI-CNN Initiator Models evaluation	63
5.1	Rat Pairings Age Group 6 Months boxplot Analysis	67
5.2	Rat Pairings Age Group 10 Months boxplot Analysis	67
5.3	Rat Type Initiator Age Group 6 Months boxplot Analysis	68



5.4	Rat Type Initiator Age Group 10 Months boxplot Analysis	68
5.5	Rat Pairings Action [HH] boxplot Analysis	69
5.6	Rat Type Initiator Action [SB] boxplot Analysis	70

Chapter 1

Introduction

The Alzheimer's disease (AD) is the most common cause of dementia and represents a significant public health problem. Disruption of social behavior is often an early symptom of AD, which frequently appears before the onset of cognitive impairments. The main objective of this project is to develop algorithms for the automatic detection and classification of various types of specific social interactions in rodents, both transgenic TgF344-AD rats - an animal model of Alzheimer's disease and healthy control rats.

By using multiple cameras and pose estimation software such as DeepLabCut, we will be able to track the points of interest on the rodent's bodies, and we will be able to preprocess the tracked points and work in a 3D coordinate system to obtain a more detailed representation of the animal skeleton. We will be able to observe rodent behavior during a social interaction test, in which unfamiliar individuals investigate each other primarily by sniffing the anogenital area, head, or body by developing and employing action recognition algorithms and deep learning. The action recognition methods will be evaluated on pre-labeled datasets such as Pair R24-M or self-made datasets obtained for the TgF344-AD and control rats.

By comparing the behavior of the transgenic rat's AD model to that of healthy control rats, we aim to identify specific changes in social behavior that may serve as early diagnostic markers for AD. Furthermore, a better understanding of the social behavior of TgF344-AD rats may lead to the development of targeted treatment options and slowing the disease's progression.

1.1 Alzheimer's Disease

AD is the most common type of dementia, described as a fatal degenerative dementing disorder with initial mild memory impairment that can progress to a total loss of cognitive and physical abilities. AD affects 10-15% of people over 65 years. Put in numbers, around 50 million patients (dating year 2020) worldwide are affected by AD, with an estimated 152 million patients by the year 2050. There is no cure for AD, but there are available treatments that can improve the symptoms. [1]

The pathological hallmarks in the brain of AD patients can be categorized into two classes of abnormal structures on a cellular level. The extracellular deposits of insoluble amyloid- β protein (amyloid- β plaques) and intracellular aggregates or tangles of hyperphosphorylated tau protein which is accompanied by neuronal loss and gradual atrophy of the parts of the brain that mediate memory and cognition. The behavioral symptoms of AD correlate with the accumulation of plaques and tangles. [2]

The symptoms of AD depend on the stage of the disease, which progresses with age. AD is classified into preclinical, mild, and dementia stages depending on the degree of cognitive impairment. In the early stages and most commonly, the initial symptom is episodic short-term memory loss with relative sparing of long-term memory. That is followed by deterioration in problem-solving, judgment, executive functioning, lack of motivation, disorganization, abstract thinking, and problems with multitasking,

leading to several behavioral dysfunctions [3]. Because AD is currently not treatable, it is important to focus on early diagnostics, studying behavioral changes (that are present with AD), and could be an indication to treat the patient before manifestation of dementia.

■ 1.1.1 AD and Social behavior

Social withdrawal is one of the earliest noticeable non-cognitive symptoms, occurring up to almost three years before the diagnosis in 40% of AD patients. By degenerating cells in the brain, AD impacts the patient's memory, including social memory [4]. The patients with AD show deficits in social cognition, emotion recognition, and empathy – the patients have difficulties recognizing known faces, reconstructing memories, or interpreting social signals, which also leads to anxiety and isolation of the affected patient [5].

■ 1.1.2 Forms of AD

AD can be categorized into two primary forms – familial and sporadic. The sporadic form is common (95% of AD cases) and can affect people older than 65 years without genetic predisposition [6]. The familial form represents 5 % of AD and can affect people much younger (starting at the age of twenty) [7]. Emerging due to gene mutations (in APP, PSEN1, and PSEN2 genes), individuals suffering from familial form have a high probability of fully developing AD. Because the familial form is hereditary with a known genetic background, it can be imitated in animal models by using genetic engineering tools [8].

■ 1.1.3 Animal Models

To study and understand a disease, we can model a non-human organism to mimic the aspects of biological processes, symptoms, or diseases found in humans. These models are often genetically engineered. In the context of AD, the most commonly used experimental animal models are transgenic mice or rats, which overexpress human genes associated with familial AD, leading to the formation of amyloid plaques [9]. While these models are essential for understanding the pathology of AD and finding potential treatments for humans, they might be limited due to disparities with humans, such as lifespan or brain structure [10].

■ 1.1.4 Transgenic Model TgF344-AD

The transgenic rat model TgF344-AD expresses the mutation of APP and PSEN1 genes [11]. Starting at the age of six months, the model shows an age-dependent accumulation of amyloid plaques in the hippocampus (major structure for learning and memory) and cortex [12]. While at that age, cognitive impairments are usually not observed [13], at the age of eight months, the model starts to indicate cognitive impairments [14]. The social behavior of the TgF344-AD model has not been fully described yet, although impairments of social behavior are common neuropsychiatric symptoms in the preclinical stage of AD.

For a thorough understanding of the social behavior of the rats, it is necessary to develop sensitive instruments that can detect specific changes in social interactions. For that, we aim our project to develop action recognition models that target rats' social behavior and work in a 3D reconstructed scene depicting the detailed rat behavior. That allows us to look at the tracked animals and their interactions in a much more detailed view while also working under conditions that are more natural for the animals

(e.g., an open arena, where the animals can freely interact without the use of distinctive markers on their bodies, which could attract the attention of another individual, etc.). Understanding social behavior in the early stages of AD can be essential for better and early diagnosis of the disease.

1.2 Pose Estimation

To successfully classify actions, the individual's pose needs to be determined. Pose estimation refers to a process of determining the spatial orientations and positions of specific body parts of an individual (human or animal), standardly from image or video data [15]. The pose can be expressed through joint position – keypoints or angles between body parts (for example, ankle, hip, etc.). These keypoints are typically defined in a 2D or 3D coordinate frame. Pose estimation is pivotal in action recognition as it enables the extraction of meaningful information regarding the subject's movements and interactions. Such information can be used in sports [16], healthcare and rehabilitation – patient monitoring or physical therapy, or behavioral studies, providing insights into behavioral patterns, social interaction and cognitive processes – whether human or animal.

Historically, the field was constrained by data availability and computational resources, leading researchers to focus on handcrafted features. However, a growing number of pose datasets (considering human pose estimation – HPE) and the advancements in deep learning have dramatically increased the accuracy and efficiency of pose estimation algorithms [17].

Advancements in deep learning, specifically regression methods, have catalyzed a methodological shift in pose estimation techniques. Toshev and Szegedy proposed a cascaded deep neural network regressor named DeepPose to learn keypoints from images [18], making extensive use of Convolutional Neural Networks (CNNs) in Pose Estimation [19].

CNNs operate through convolutional layers that employ filters to capture spatial features from input images, starting with simple edges and textures, and building up to complex structures that can represent parts of the body such as limbs. In the context of Pose Estimation, CNNs solve a regression problem of pointing the coordinates of a keypoint on the individual's body [18].

Multiple algorithms have been published over the past decade for pose estimation (e.g., OpenPose, DeepLabCut, SLEAP or DeeperCut). These algorithms allow for utilizing pre-trained networks or training new networks by a rather small number of labeled examples [20]. In this work, the DeepLabCut (DLC) algorithm will be employed [21].

DLC is based on multi-task convolutional neural networks (CNNs) by predicting score maps that encode the probability of a keypoint occurring at a particular location and location refinement fields (predicting offsets). To solve a problem of multiple individuals (ergo the CNN predicts the location of a keypoint for more individuals), DLC employs a network to predict Part Affinity Fields (PAFs) to assemble keypoints into shapes that define an animal (or a human). Several networks can be used for the pose estimations in the DLC framework, such as adapted ImageNet, pretrained ResNets, EfficientNets, or a multiscale DLCRNet-ms5 [21].

PAFs are set of flow fields encoding unstructured pairwise relationships between different keypoints (essentially, a 2D vector field for each limb, e.g., right shoulder and right elbow, with each vector in the field pointing along the direction of the limb) [22].

PAFs provide a structured, spatially coherent representation of the detected keypoints into full-body poses, especially in scenarios where multiple subjects are present in an image [22].

1.3 Tracking

In the context of pose estimation, tracking involves following the identified keypoints across sequential frames in a video and determining a constant identity through time for the given individual. Such a task can be difficult, especially with multiple individuals present in a video, where occlusions of the individuals occur. A common approach involves algorithms that predict the position of a keypoint in the subsequent frame based on its previous positions or state, such as the Kalman Filter [23], up to deep learning methods such as DeepSort or Detection Embeddings for Tracking (DEFT) [24].

The Kalman Filter is a predictive model that estimates the future state $x \in \mathfrak{R}^n$ (2D or 3D points in the case of pose estimation) based on its previous states, adjusting predictions with new observations, defined as follows:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (1.3.1)$$

With a measurement $z \in \mathfrak{R}^m$ that is:

$$z_k = Hx_k + v_k \quad (1.3.2)$$

Where v_k and w_k represent the process and measurement noise, respectively, and are assumed to be independent with normal probability distributions. A ($n \times n$) represents the state transition from the previous time step $k - 1$ to the state at the current time step k , and B ($n \times l$) represents the optional control input $u \in \mathfrak{R}^l$ to the state x . Matrix H ($m \times n$) relates the state to the measurement z_k and x_k denotes the state at time step k [25].

DeepSORT extends the SORT (Simple Online and Realtime Tracking)[23] algorithm by incorporating a deep learning-based model to maintain identities across frames. DeepSORT combines motion information predicted by a Kalman Filter with appearance information gained from a deep neural network to perform robust tracking, even in the presence of occlusion and overlapping objects [26].

DEFT represents a current state-of-the-art tracking-by-detection system. Compared to the methods mentioned above, DEFT optimizes tracking and detection simultaneously in a single deep neural network. The detection backbone of the network extracts object embeddings that are used in a matching head of the network to associate objects during training – the appearance is extracted from multiple receptive fields, which provides additional robustness. The matching head then estimates similarity scores between all pairs of detections across the frame at the current and previous steps. As the motion model DEFT uses motion forecasting based on the LSTM module (a form of Recurrent Neural Network – see 1.4), which replaces the form of motion model based on Kalman Filter [24].

In the case of DLC, the tracking is approached as a global minimization problem, where connecting two candidate tracklets (tracklets represent a path that a particular keypoint takes across a number of consecutive frames) incurs a cost inversely proportional to the likelihood that they belong to the same track, solving the problem by optimization techniques. The simplified problem could be defined as follows:

$$A^* = \arg \min_A \sum_{i,j} C(i,j)A(i,j) \quad (1.3.3)$$

Where A is the assignment matrix, indicating a keypoint i is assigned to keypoint j in the next frame or otherwise, and C is the cost function, typically based on the distance between keypoints and other relevant features. This approach is applied in difficult cases where, for example, occlusions occur. In the case of the frames where multiple subjects are distinguishable by the distance between them, a simple online tracking approach is applied [21].

1.4 Action Recognition

In the field of computer vision, action recognition is a critical task for human-computer interactions, surveillance, and healthcare to behavioral analysis [27]. The task of action recognition is to identify and classify various actions (human or animal) within a sequence of images or videos and such models need to capture the complex spatial and temporal dynamic of the actions.

Such dynamics can be captured in various data modalities, such as hand-crafted features, skeletal data (refer to Section 1.2), or RGB [28]. Traditional methods relying on the hand-crafted features derived from the images (descriptors like Histogram of Oriented Gradients or Optical Flow) classify sequences using machine learning techniques such as Support Vector Machines and Hidden Markov Models [29]. However, due to the widespread use of deep learning, methodologies like CNN, GCN (graph convolution network), RNN (recurrent neural network), or their combination can successfully capture the complex spatial and temporal patterns necessary for recognizing actions.

The GCN defines a first-order approximation of localized spectral filters on graphs, which can be understood as a generalization of CNN to graph-structured data, performing graph convolutions across the nodes connected by edges [30]. The approximation is defined as [30]:

$$X_{out} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X_{in} W^l \right) \quad (1.4.1)$$

Where X_{in} represents the input features, \tilde{A} represents the adjacency matrix, \tilde{D} is the diagonal degree matrix of \tilde{A} , W represents the trainable weights and σ denotes the activation function. The RNN maintains a memory while processing sequential data through internal states, based on which the RNN can make a prediction for the current prediction step [31].

For the CNN approach, the prior inputs are the RGB images of given sequences. The model's backbone is often a 2D CNN, which learns the spatial dependencies of the images and is followed by RNN-LSTM (long short-term memory) layers to capture the temporal dynamics [27]. A 3D CNN can also capture the spatial-temporal dependencies of the sequence, where the images are represented as motion cuboids, which, instead of an original image, represent absolute difference frames – attained by subtracting the earlier frame t with current frame $t + 1$ on a pixel-by-pixel basis [32]:

$$D_t(x, y) = |I_t(x, y) - I_{t+1}(x, y)| \quad (1.4.2)$$

Where $I_t(x, y)$ is the value at image pixel (x, y) , and D_t is the resulting motion cuboid at time step (frame) t . These networks are mainly used for clip-level learning (e.g., 16 frames in each clip) [28].

The CNN approach can also be used on the skeletal data (3D coordinates), where an image representing a given sequence is obtained from the coordinates of given joints (JTM – joint trajectory map) [33]. Each pixel in the image can encode the spatial position of the joint (e.g., x,y,z coordinates, rotation with respect to the other joints) and

the temporal dependencies [34]. The performance of the coordinate visualization methods based CNN models can be enhanced using a proper tree structure with reference joints [35].

The GCN approach focuses on the skeletal data – 2D or 3D coordinates represented as graph structures with nodes and edges. In such a structure, a node can be defined by its spatial coordinates, while the edges can encapsulate a link between the given joints on a body or attributes such as distance between connected nodes. A variety of links (edges between the nodes) can then be employed to capture the spatial and temporal dependencies in the sequences (Spatial-Temporal GCN – ST GCN) [36]. ST GCN approach can then be extended to an Actional-Structural GCN (AS-GCN) or Adaptive GCN, which employ encoders [37] or backpropagation algorithms, respectively, to adapt the graph topologies dynamically [36].

Aggregating these modalities and approaches in a multi-modal network can result in more accurate and robust action recognition [28]. For example, the Model-Based Multimodal Network (MMNet), fuses skeleton and RGB modalities by a model-based approach [38].

1.5 3D Reconstruction

To obtain a more detailed view of a scene acquired on an image or video in two-dimensional space, 3D reconstruction techniques can be employed to extrapolate spatial depth and provide a three-dimensional representation – improving the understanding and analysis of the spatial relationships within the scene. The three-dimensional representation can be reconstructed if multiple views of the scene are available. The most common methodologies for 3D reconstruction from 2D keypoints are Stereo Triangulation or Direct Linear Transformation (DLT).

The stereo triangulation can be described such as – given a point P in 3D coordinate system and its projections p_1 and p_2 in two different camera views, the 3D coordinates of P can be determined by finding the intersection of two straight lines originating from the camera centers and passing through p_1 and p_2 , respectively, defined as follows [39]:

$$\begin{aligned} p_1 &= M_1 P \\ p_2 &= M_2 P \end{aligned} \tag{1.5.1}$$

Where M_1 and M_2 are the projection matrices of the two cameras (which can be computed out of the camera’s intrinsic and extrinsic parameters). Solving these equations leads to a reconstructed point P in the 3D coordinate system, typically with linear triangulation.

DLT establishes a linear relationship between the point P in the 3D coordinate system and its 2D projections into the image through the corresponding camera matrices, which can be defined as follows [39]:

$$\begin{aligned} P &= [X, Y, Z, 1]^T \\ p_{2D} &= [x, y, 1]^T \end{aligned}$$

Where the relationship between the 3D point P and its 2D projection p_{2D} can be defined as $p_{2D} = MP$ with M being the camera projection matrix. A set of linear equations is constructed for every point in the 2D projection based on the camera matrices into a matrix A , and the solution is found by a Singular Value Decomposition (SVD) [39].

Chapter 2

Data

2.1 Hardware Setup

For the execution of experiments, model training, data acquisition, and preprocessing, we assembled the following hardware setup:

- Processor: Intel Core i9-9900K with 8 physical cores (16 threads)
- Motherboard: GIGABYTE Z390 UD - Intel Z390
- CPU Cooler: Scythe SCFM-1100 FUMA Rev.B
- RAM: HyperX Fury Black 32GB (2x16GB) DDR4 3200
- System Drive: WD SSD Blue 3D NAND, 2.5 - 500GB
- Data Processing Drive: Transcend MTE220S, M.2 - 1TB
- Data Storage and Archiving Drive: Seagate IronWolf, 3.5 - 6TB
- Power Supply: Seasonic Focus Plus Gold - 750W
- Graphics Card: GeForce RTX 3080 VISION OC 10 GB
- Additional Hardware: Mounted with 4x USB 3 PCIe cards AXAGON PCEU-43V
- Software: Microsoft Windows 10 Pro EN 64bit DVD OEM
- Cameras: 4x Intel® RealSense™ Depth Camera D435

2.2 Camera Setup

To record the behavior of the rats within the arena, four Intel® RealSense™ D435 cameras were positioned in a quad view setup, each at a ninety-degree angle. Utilizing a Python 3.10 environment and the pyrealsense2 library (version 2.53.1.4623), these cameras were configured to capture videos at a resolution of 1280 x 720 at a rate of 30 frames per second. Due to the absence of necessary pins for RGB frame acquisition on the camera chip, hardware synchronization of the cameras was not feasible. Therefore, software synchronization was implemented. While this approach posed a risk of losing a few frames, any such loss was insignificant and did not proportionately impact our data acquisition process. The camera quad view setup with the rat's arena can be seen in Figure 2.1.



Figure 2.1. The setup of the quad view camera system built for capturing our experiment video footage.

2.3 Animals

TgF344-AD (N = 8 males, AD group) and F344 (N = 8 males, WT group) rats were used for social interaction tests. The rats were bred in the Animal Facility of the Institute of Physiology CAS and used for the experiments at the age of 6 and 10 months. The rats were housed in pairs in a room with controlled conditions (22 °C, 50–60% humidity, 12 h light/dark cycle) and dimmed light (11 lux) to avoid retinal degeneration. At the age of 3 weeks, a small piece of tissue was collected from the tip of the tail and used for genotyping. All animal treatment complied with the Animal Protection Code of the Czech Republic and the European Community Council directive (2010/63/EC).

2.4 Experiment Setup

For each experimental session, a ten-minute video of two rats interacting in an open arena was acquired by the quad-view camera setup. To observe the behavioral patterns that could potentially be related to AD, the pairs of rats were placed on a square-shaped acrylic arena with dimensions 50x50x45 [cm]. The arena was cleaned before each session (the 10-minute video-recording of two rats) with a 30% alcohol solution. To differentiate between the studied rats, we added a mark in the form of color stripes on the rat's tails. We used two groups of rats: the transgenic model TgF344-AD (will be described as AD, refer to Section 1.1.4) and the healthy control specimen F344 (WT as wild type). Individual pairs of rats were placed in the arena to interact. The pairing based on the type was WT-AD, WT-WT, AD-AD. We performed two experiments - Experiment 1 when the rats were 6 months old and Experiment 2 when the rats were 10 months old. Each experiment consisted of two subsequent recording days - Day 1 (habituation, familiar rats in pairs) and Day 2 (interaction, unfamiliar rats in pairs). Habituation day combinations are described in Table 2.1.

Specific unfamiliar rat pairings were established for Day 2 for each experiment to focus on the rat's social interactions rather than their exploration of the arena. Interaction day combinations are described in Table 2.2 (the index after the rat type indicates the specific identity of the rat within that type).

Day 1 (Habituation) Combinations	
AD1	AD2
AD3	AD4
AD5	AD6
AD7	AD8
WT1	WT2
WT3	WT4
WT5	WT6
WT7	WT8

Table 2.1. Table describing the experiment’s pairings of Day 1 (Habituation).

Day 2 (Interaction) Combinations			
Age: 6 Months		Age: 10 Months	
AD1	WT1	AD1	WT2
AD5	AD7	AD5	AD8
AD2	WT2	AD2	WT1
WT5	WT7	WT5	WT8
AD3	WT3	AD3	WT4
AD6	AD8	AD6	AD7
AD4	WT4	AD4	WT3
WT6	WT8	WT6	WT7
AD1	AD3	AD2	AD3
AD5	WT5	AD6	WT5
WT1	WT3	WT2	WT3
AD6	WT6	AD5	WT6
AD2	AD4	AD1	AD4
AD7	WT7	AD8	WT7
WT2	WT4	WT1	WT4
AD8	WT8	AD7	WT8

Table 2.2. Table describing the experiment’s pairings of Day 2 (Interaction).

To see the dependency of the rats’ social interactions on the progression of AD, this experiment was conducted on two different ages of the rats - at six months and ten months of age, as stated above.

Note: The experiments conducted at the different ages of the rats were conducted in a different combination for Day 2 to pair unfamiliar rats. The experiments were conducted in low-light conditions.

2.5 Camera Calibration

For each experiment set, a camera calibration was done. Cameras were calibrated using a checkerboard pattern and MATLAB calibrations toolbox. A video capturing the checkerboard moving within the quad camera setup’s field of view was recorded, and representative images for each camera were extracted and utilized within the calibration toolbox to compute the intrinsic and extrinsic parameters of the cameras.

The checkerboard pattern was generated as follows: 9 x 12 black and white squares with a side length of 31 millimeters.

In our setup, Camera 1 was stated as the reference camera, with its rotation and translation parameters set to 0. Therefore, the spatial configuration of the other cameras was determined relative to Camera 1. The calibration process involved calibrating Camera 1 with each of the other cameras in pairs, i.e., 1-2, 1-3, and 1-4.

2.6 Data Acquisition

After acquiring experiment video footage by the quad view camera setup, pose estimation and tracking were employed by DeepLabCut (DLC). The DLC model was trained to recognize a pose of nine selected keypoints on the rat’s body. These keypoints were selected to represent the whole body of the given rat with consideration of the actions that we wanted to recognize – limbs and tail were omitted. The selected keypoints are described in Table 2.3. As the DLC model, we trained DLCRNet-MS5.

Keypoint Index	Keypoint
P1	Snout
P2	Left Ear
P3	Right Ear
P4	Spine 1
P5	Spine 2
P6	Spine 3
P7	Spine 4
P8	Spine 5
P9	Tail-Base

Table 2.3. Table of keypoints and their corresponding anatomical locations.

DLC setup based on this configuration provided the representation of the rat’s body in each frame with the identification of the rat based on the DLC tracking (if successful). Cases where DLC may have misinterpreted the pose or tracking are discussed and addressed in the following sections. To view the DLC configuration and setup, refer to Appendices Section.

2.7 Video analysis

Videos were divided into twenty-second segments using the FFmpeg library (version 0.2.0) to limit the possibility of long-term identity swaps. Each segment is then processed independently. A preliminary five-second gap is introduced before the video cutting and analysis to allow for camera stabilization and experiment initialization. Manual identification of subjects is performed in the first available frame. The segmented videos are subsequently analyzed using the trained DLC model, employing a skeleton tracking method for increased reliability (refer to Appendices Section).

2.8 Data Pre-processing

Following the acquisition of two-dimensional coordinates for each video frame from DLC analysis, it was imperative to preprocess the coordinates to ensure their reliability and robustness across all models. To briefly summarize the DLC pose estimation and tracking, given a video with 600 frames, considering four views for each of the two animals, we obtained 4800 instances of tracked points. Each instance represents a frame from a specific view and contains the spatial coordinates of all identified keypoints for the animal present. Out of these, 3600 instances had correctly assigned positions for markers (estimated or predicted keypoint with information of their position in the image, identification of the animal, and identification of the body part – such as snout). The correctly assigned position was determined by the likelihood of the marker’s coordinate estimation provided by DLC. The likelihood threshold ($T_l = 0.95$) was set for the sum of the likelihoods of the nine selected points in a single frame for one rat. Based on T_l , approximately 0.75 fraction of the analyzed frames accurately represented one rat’s position. The preprocessing stage involved missing data imputation in a spatial and temporal manner and outlier detection to address this issue. The preprocessing stage also involved determining the location of 2D coordinates from multiple views in a 3D coordinate system using the extrinsic and intrinsic parameters of the calibrated cameras and triangulation methods.

2.9 3D Coordinate Reconstruction from Multi-View 2D Projections

To depict the rat’s activity, orientation, and movement patterns, we decided to transform the 2D coordinates into 3D space. Given the option of quad-view recording the system, where cameras were arranged at 90-degree angles relative to each other, we could capture the essential features from all sides of the rat’s body. Two primary methods were considered to perform the 3D reconstruction from 2D coordinates: Direct Linear Transformation (DLT) and stereo triangulation (refer to Section 1.5). The 3D reconstruction was done for each marker estimated in at least two of the four views for a given frame.

The stereo triangulation was employed with the function ‘triangulate’ available in MATLAB, which utilizes pairs of coordinates from various camera view combinations and their corresponding camera matrices as inputs. After processing all view combinations, the final 3D coordinates were derived by computing the column-wise mean of the resulting matrix.

To employ the DLT, we used Singular Value Decomposition (SVD) as described in Section 1.5. The right singular vectors from matrix V (computed from the SVD) corresponding to the three largest singular values are utilized to derive the homogenous coordinates of the 3D point. To transform these coordinates back to the Cartesian system, they are divided by the fourth component of the corresponding singular vector.

Triangulation methods were compared using a sum of squared reprojection errors across all four views. With our current setup and DLC estimated markers, the DLT method proved more robust, in addition to a faster computation.

To optimize the solution provided by DLT, we used the acquired 3D coordinates as an initial guess for a Levenberg-Marquardt iterative method (employed in MATLAB). The objective function for the optimization was the reprojection error. The error was computed for each 3D point by projecting it back onto each camera plane using the

respective camera matrices and then computing the Euclidian distances between the reprojected 2D points. The total reprojection error to be minimized was then computed as the sum of these distances across all camera views, defined as follows:

$$E^* = \sum_{i=1}^V \sum_{j=1}^M E_{i,j} = \sum_{i=1}^V \sum_{j=1}^M \left\| P'_{2D,i,j}(x, y) - P_{2D,i,j}(x, y) \right\| \quad (2.9.1)$$

Where M is the number of markers (18; both rats), V is the total number of views (4), E^* is the total reprojection error, $E_{i,j}$ is the reprojection for the given view i and marker j , $P'_{2D,i,j}$ is the reprojected marker's j -th position for view i in the 2D coordinate system, and $P_{2D,i,j}$ is the marker's j -th position for view i in the 2D coordinate system estimated position by DLC.

2.10 3D Coordinate Spatial Imputation

The number of instances where markers were missing in the 3D space was minimized using the information from multiple camera views if estimated in at least two of them. When a marker was not estimated in one or two views – potentially due to occlusion or limitations within the DLC estimations, it could have been projected to the 3D space using other views. However, if the point was not estimated in more than two views simultaneously for a given frame, then the marker was absent in the 3D coordinate system, necessitating the implementation of spatial imputation methods. To address these instances, we employed two primary methods for spatial imputation: Point Distribution Models (PDM) and geometric rule-based approaches (refer to Sections 2.10.1 and 2.10.2, respectively). These methods enabled a robust reconstruction of the 3D coordinates, even if the position of the marker or more was missing. The geometric rule-based imputation preceded the PDM to lower the number of markers the PDM had to estimate and the mean squared error (MSE) – refer to Experiments sections 4.2 and 4.3.

2.10.1 Point Distribution Model for 3D coordinate Imputation

To train a PDM [40], each of the shapes (defined by nine markers in the 3D coordinate system) had to be aligned, and valid shapes for training had to be selected. Such shape was again selected by the likelihood of the DLC estimations, using the threshold T_l (see Section 2.8), however, the sum of the likelihoods was done across all views for the given instance. To define whether a shape in the 3D coordinate system is considered valid, the following criterion was used:

$$\sum_{i=1}^V \sum_{j=1}^M L_{i,j} \geq T_l VM \quad (2.10.1.1)$$

Where V is the total number of views, M is the total number of markers, and $L_{i,j}$ is the likelihood of estimating marker j in view i .

A reference shape had to be selected to rigidly align all valid shapes, and the transformation (rotation, translation, and scaling) minimizing the sum of squared distances in the 3D space had to be found. The objective function was defined as follows:

$$E = \sum_{i=1}^M w_i \left\| sR(\alpha, \beta, \gamma) \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} - \begin{pmatrix} x_{0i} \\ y_{0i} \\ z_{0i} \end{pmatrix} \right\|^2 \quad (2.10.1.2)$$

Where M is the number of markers, w_i are weights for each marker (set as one for each marker being equally important), R is the rotation matrix (refer to equation 2.10.1.3), s is a scaling factor, (t_x, t_y, t_z) are the translations, (x, y, z) are the coordinates of the markers to be aligned and (x_0, y_0, z_0) are the reference markers. The rotation matrix R is defined as:

$$R(\alpha, \beta, \gamma) = R_z(\gamma)R_y(\beta)R_x(\alpha) \quad (2.10.1.3)$$

This matrix represents the general 3D rotation matrix obtained by the product of the rotation matrices for yaw, pitch, and roll by angles γ , β , and α respectively.

The minimization of the objective function $E(\alpha, \beta, \gamma, s, t_x, t_y, t_z)$ was decomposed into an outer minimization with respect to α, β, γ and an inner minimization with respect to s, t_x, t_y, t_z . The inner minimization was performed by setting the partial derivatives of the corresponding variables (s, t_x, t_y, t_z) to zero, which led to solving a system of linear equations (2.10.1.4).

$$\frac{\partial E}{\partial s} = 0, \quad \frac{\partial E}{\partial t_x} = 0, \quad \frac{\partial E}{\partial t_y} = 0, \quad \frac{\partial E}{\partial t_z} = 0 \quad (2.10.1.4)$$

The outer minimization was performed numerically by employing the function ‘fminunc’ in MATLAB. After aligning the selected shapes to a reference one, obtaining $\{x^1, \hat{x}^2, \dots, \hat{x}^N\}$, the mean shape was calculated and aligned with x^1 (resulting in adjusted mean \bar{x}). The mean shape \bar{x} was then subtracted from the obtained aligned shapes matrix $S_{centered}$ ($3M \times N$) before employing principal component analysis (PCA) to determine the principal components of the model, which are found as the eigenvectors of the covariance matrix of $S_{centered}$. The resulting PCA was executed in such a way as to retain principal components that cumulatively explained at least 95% of the variance in the data.

The principal components were then used to solve a linear system in a subspace determined by the observed markers to impute the missing markers in the given instance (the observed markers were aligned with the reference shape by a transformation T_o). This process involved computing the deviation of the observed markers in the incomplete shape from the mean shape and determining the weights that, when multiplied with the principal components, optimally represent the deviation within the observed subspace. The linear system was defined as follows:

$$P_{K,observed}b_K = \tilde{x}_{observed} - \bar{x}_{observed} \quad (2.10.1.5)$$

Where $\bar{x}_{observed}$ is the mean shape in the observed subspace, $P_{K,observed}$ is the reduced eigenvector matrix in the observed subspace, b_K are the weights, and $\tilde{x}_{observed}$ is the observed aligned shape.

The derived weights were subsequently used to reconstruct the complete shape \tilde{x} , defined as:

$$\tilde{x} = \bar{x} + P_K b_K \quad (2.10.1.6)$$

Where \bar{x} is the mean shape, P_K is the reduced eigenvector matrix, b_K are the weights, and \tilde{x} is the completely aligned shape.

An inverse transformation to T_o was employed to complete the imputation process and transform the complete shape \tilde{x} to its proper position, rotation, and scaling. The inverse transformation used the transposed rotation matrix R (defined in equation 2.10.1.3), division by the scaling parameter, and translation by the translation parameters derived in equation 2.10.1.4.

To make the model robust for the poses of a given analyzed time window of the video, it was trained from valid shapes selected from each analyzed window as defined in Section 2.7. Experiments were conducted to see how the model behaves with an increasing number of missing markers (refer to Experiments section 4.2). We depict the alignment of the same set of markers to the reference shape x^1 in Figure 2.2. The PDM's visualization of the estimation of missing markers is depicted in Experiment Section 4.2.

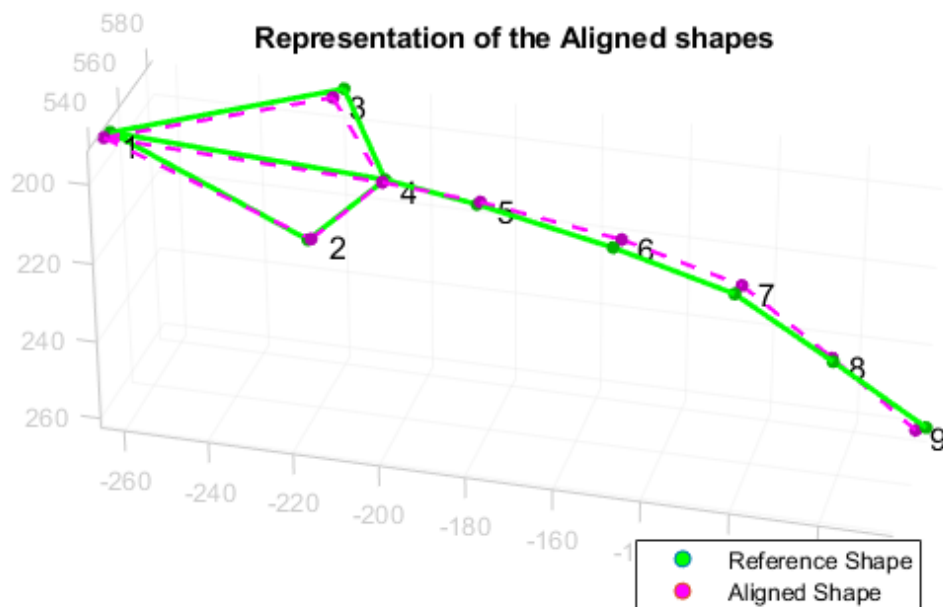


Figure 2.2. The alignment of two shapes (same set of markers to a reference shape) by PDM representing the rodent's markers estimated by DLC and reconstructed into a 3D coordinate system via triangulation and estimation in multiple views.

2.10.2 Geometric Rule-Based Imputation

The positions of markers representing the snout, left ear, and right ear exhibit symmetrical properties. When one of these markers' positions in the 3D coordinate system could not be triangulated, but the corresponding symmetrical markers were present, we could estimate their positions based on geometric rules. The snout estimation will be defined as an example (the remaining two markers - left ear and right ear, follow the same approach). The following equations define the model:

$$\begin{aligned}
 p_m &= \frac{1}{2} \cdot (p_2(x, y, z) + p_3(x, y, z)) \\
 \vec{v} &= \frac{p_m(x, y, z) - p_4(x, y, z)}{\|p_m(x, y, z) - p_4(x, y, z)\|} \\
 p_{estimated} &= p_4 + \vec{v}S\|p_m(x, y, z) - p_4(x, y, z)\|
 \end{aligned}
 \tag{2.10.2.1}$$

Where p_m is the midpoint between the position of two given markers – left (p_2) and right ear (p_3) in the provided example, \vec{v} is a directional unit vector describing the

direction of the estimated marker – p_4 being the first spinal marker, and x, y, z are the 3D coordinates representing the given marker.

To estimate the position of the snout, we first computed the midpoint p_m between the positions of the left ear (p_2) and right ear (p_3). We then formed a vector \vec{v} from the first spinal marker (p_4) to this midpoint. This vector was normalized to convert into a unit vector, maintaining the direction of \vec{v} but with a magnitude of one, as shown in the equation. By multiplying this unit vector by a scaled version of the distance from the first spinal marker to the midpoint, we extended along the direction of \vec{v} to approximate the position of the snout. The scaling factor S was determined based on training of valid shapes (following extraction described in Equation 2.10.1.1), minimizing the Mean Squared Error (MSE) between the approximations and valid shapes. A different scaling factor was set for the snout and ears, and more specifically, the scaling factor was determined for each of the examined rodents in the given video subsets. This approximation leverages the geometric and symmetrical properties of the rat's body, assuming the snout lies along the line that extends from the first spinal marker and the midpoint of the rat's ears.

To find the optimal scaling factor S , a systematic search was conducted for each rat and specified marker. This involved computing and minimizing the MSE between valid shapes and their respective copies with either snout or ear markers omitted. The optimal scale factor was determined within a predefined range from zero to six, evaluating 200 distinct values. Graphs in Figure 2.3 represent the systematic search throughout 400 valid shapes for each rat. The imputation is depicted in Figure 2.4.

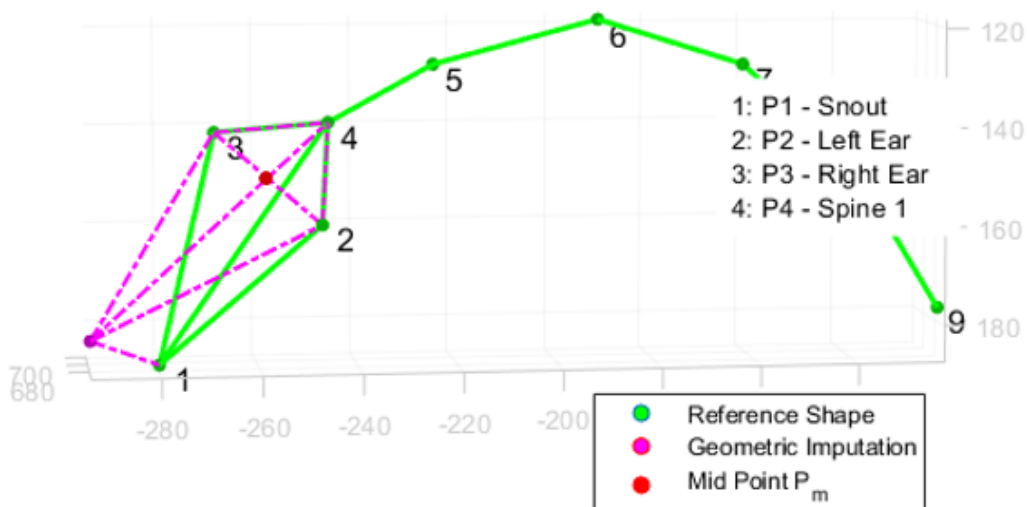


Figure 2.4. Example of the geometric rule-based imputation of the missing marker (Example of snout imputation). Green markers represent the reference shape, purple marker represents the estimated snout, and red marker represents midpoint p_m (see equations 2.10.1.2).

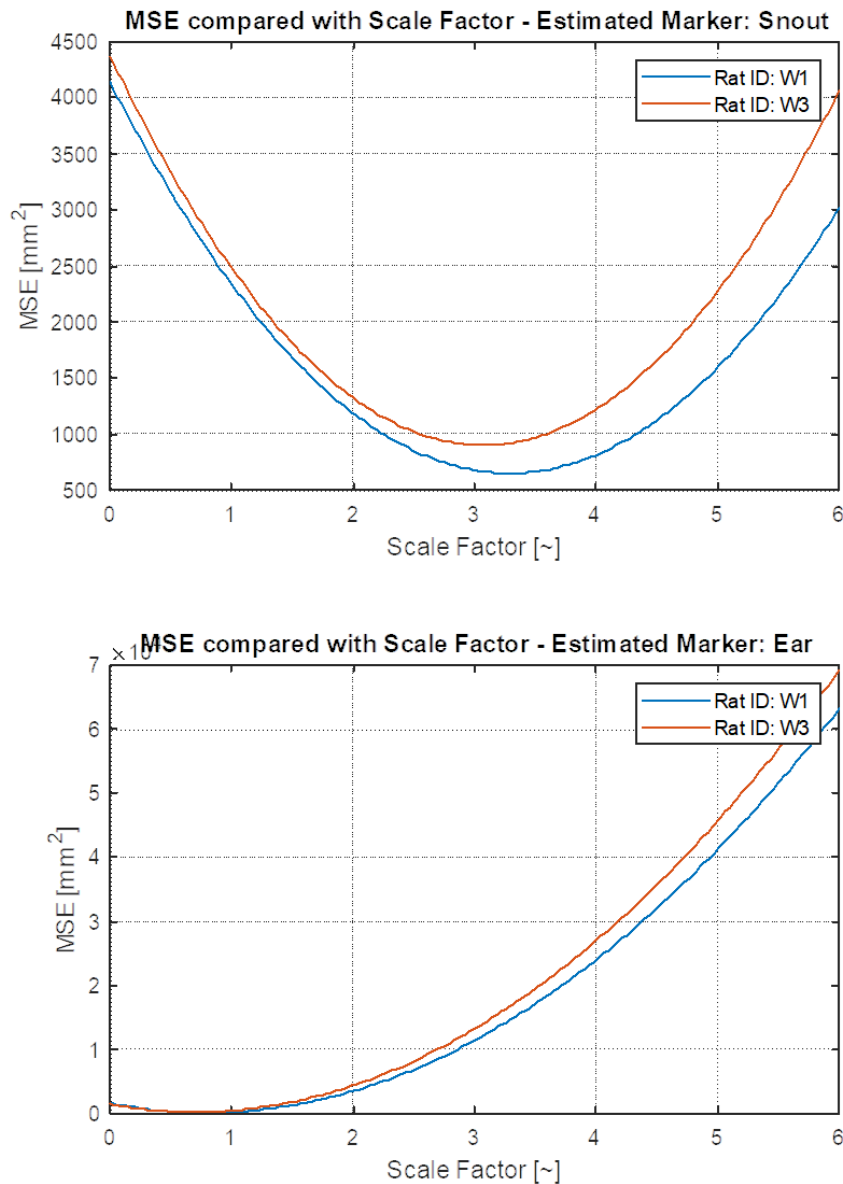


Figure 2.3. The representation of the systematic search for the optimal scale factor and the specified marker based on MSE, comparing two distinct exemplary rat's estimated markers by DLC.

2.11 3D Coordinate Temporal Imputation

For the temporal imputation of the 3D coordinates, we utilized a piecewise cubic Hermite interpolating polynomial (PCHIP) method built in MATLAB. This choice was based on a simple test, where we compared ground truth sequences and sequences with removed instances (instance in this section contains not a single view 2D markers for a given time step, but the markers in the 3D coordinate frame), interpolated by linear interpolation, piecewise cubic spline, previous non-missing value, nearest non-missing value, and PCHIP. Based on the sum of squared errors (refer to Experiments

4.4), PCHIP was the most suitable choice for the rat’s movement, possibly due to its limitation of peaks and oscillations [41].

A threshold of 15 instances was established as the maximum allowable duration for these discontinuities. Instances exceeding this threshold were identified as invalid and subsequently excluded from training and evaluation processes. It was possible for such instances to be present in our dataset, however, only 3.65% of all instances were found as such.

2.12 Outlier Detection

We identified and eliminated certain markers within specific frames and views for particular rats through visual inspection and assessment of the likelihood associated with DLC marker estimations. The decision to remove specific markers was primarily based on their estimation likelihood, with a threshold of 0.6 as the minimum acceptable likelihood for a marker to be retained. Markers with estimation likelihood below this threshold were removed and imputed using the PDM or geometrical rule-based imputation methods (see Section 2.10).

In certain frames, it was observed that the DLC estimation lost track of the rat’s identity, leading to incorrect marker assignments. Such inaccuracies manifested as either both identities being assigned to a single rat or only one rat’s markers being estimated. We excluded the entire view from the triangulation process to address this, ensuring the invalid instances would not affect the 3D coordinates. For the potential identity swap (in a temporal manner, a rat was assigned the other rat’s identity, which was not manually assigned to it – refer to Section 2.7), we developed a combinatorial optimization and iterative 3D tracking (refer to Section 3.1). We represent the correction in Figure 2.5 and Figure 2.6.

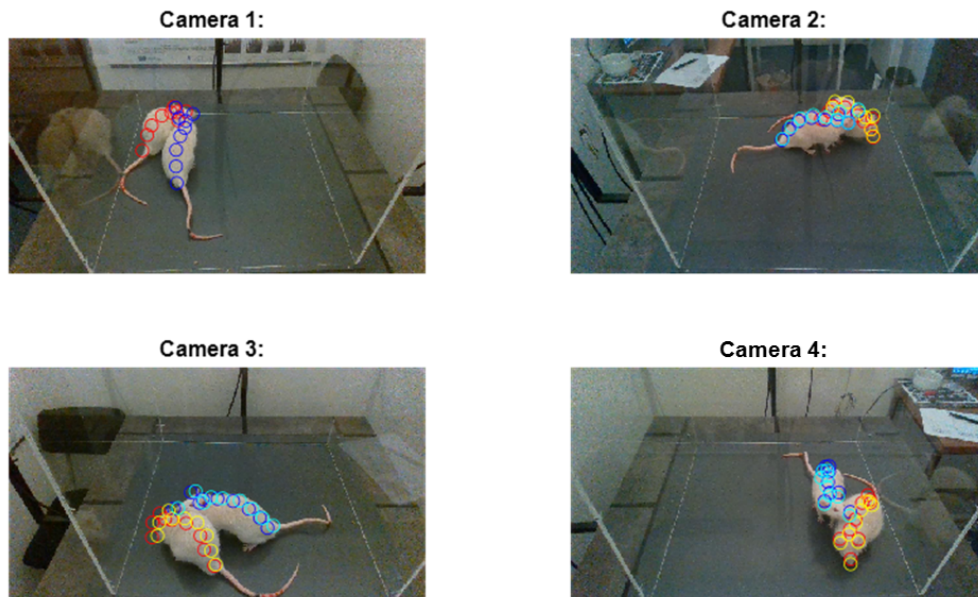


Figure 2.6. An example while excluding view camera 1 from the triangulation leads to lower reprojection error, and the triangulated points are unaffected by the wrongly assigned identity (refer to Figure 2.5).

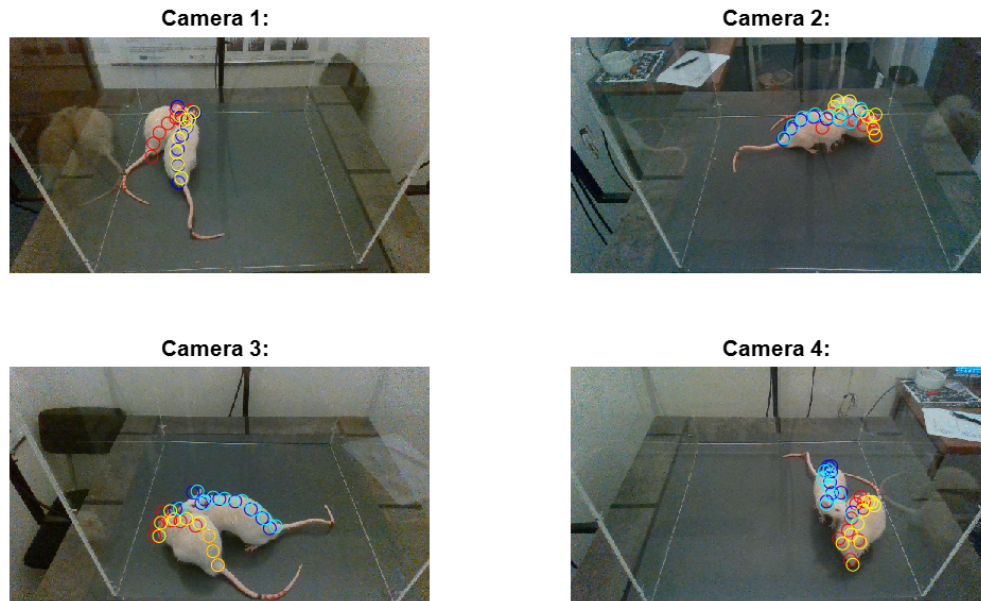


Figure 2.5. The estimated markers of two rats for the given views by DeepLabCut and their reprojections after triangulation [Subset: WT1WT4, frame 385]. The yellow and cyan-colored markers represent the estimations, and the red and blue markers represent the reprojections after triangulating the estimations. Notice a missing estimation for one of the rats in camera view 1. We can see a situation where DeepLabCut wrongly assigns identity and estimates the pose for only one animal (after losing track of the rats in frame), leading to wrongly triangulated markers (visible on reprojection track of camera view 1-2 red markers).

2.13 Datasets

2.13.1 Custom Datasets

The dataset collected for the primary approach of recognizing the social interactions of two rats originates from the Laboratory of Neurophysiology of Memory, Czech Academy of Sciences in Prague. The dataset contains video footage from four camera views of two rats interacting in a box-shaped arena with low light conditions (refer to Section 2.4). Each subset (four videos of each camera view) averages a duration of 9 minutes (for one camera view). A total of 48 subsets were collected.

Subsets were manually annotated to label specific rat social interactions. The label was assigned via viewing sequence windows of fifteen consecutive frames in all four camera views and determining the label for a given action. Subsets were annotated by a single person. Two datasets were labeled. Primarily, the first minutes of the subsets were labeled – the appearances of the examined actions were more prevalent during these initial moments of the experiments.

The first dataset was designed for the evaluation of the Rule-Based Model (see Section 3.2.1). Five following classes were denoted (classes denoted with apostrophe had the initiator of given action labeled):

- Absence of Social Interaction [NS] - The action was labeled as such if the rats did not exhibit any social interest in each other.
- The head of one rat is close to the head of the other rat [HH] - The action was labeled as such if the heads of the rats were close to each other - contact distance.

- Snout of one rat is close to the tail-base of the other rat [SB]' - The action was labeled as such if the snout of one rat was examining the tail-base of the other rat - contact distance.
- One of the rats approaches the other [AP]' - The action was labeled as such if the rat started moving towards the other - from enough distance (not contact distance)
- Passive interaction - close proximity with no active exploration [PS] - The action was labeled as such if the rats were still and close to each other - contact distance.

A total of 336 sequences were denoted for class [NS], 20 sequences for class [HH], 137 sequences for class [SB], 21 sequences for class [AP], and 58 sequences for class [PS]. We denote the dataset as Rule-TgF344AD Pair 18.

The second larger dataset followed the same sequence lengths and subsets. However, we employed different classes to evaluate and learn the deep learning action recognition methods. This divergence in class selection was driven by the strengths of deep learning in capturing more complex patterns, which might be challenging to define strictly through rules. Classes are the following:

- Absence of Social Interaction [NS] - The action was labeled as such if the rats did not exhibit any social interest in each other.
- Mutual Social Engagement-exploration [EX] - The action was labeled as such if both of the rats examined each other.
- Olfactory Investigation (one animal inspecting the other by sniffing) [OI]' - The action was labeled as such if only one of the rats examined the body parts of the other.
- One of the rats approaches the other [AP]' - The action was labeled as such if one of the rats suddenly starts moving towards the other rat (even upper body sudden movements from a closer distance).
- Passive interaction - close proximity with no active exploration [PS] - The action was labeled as such if the rats were still and close to each other - contact distance.
- Disengagement (leaving) [DS]' - The action was labeled as such if one of the rats lost interest in the other and started moving away from the other rat.
- Mounting Behaviour [MT]' - The action was labeled as such if one of the rats started rearing above the other - contact distance.
- Synchronized (mimicking) action (the animals simultaneously exhibit rearing behavior) [MM] - we noticed that sometimes one of the rats starts mimicking the rearing behavior of the other. If so, the action was labeled as such without being dependent on the distance between the rats.

Totally 3143 sequences were denoted – 1859 for class [NS], 121 sequences for class [EX], 632 sequences for class [OI], 163 sequences for class [AP], 55 sequences for class [PS], 74 sequences for class [DS], 191 sequences for class [MT], and 48 sequences for class [MM]. We denote the dataset as TgF344AD Pair 18 (the AD type, two animals per subset, and 18 markers for both animals). The example of labeled sequences, represented by one frame, is in Figures 2.7 and 2.8.

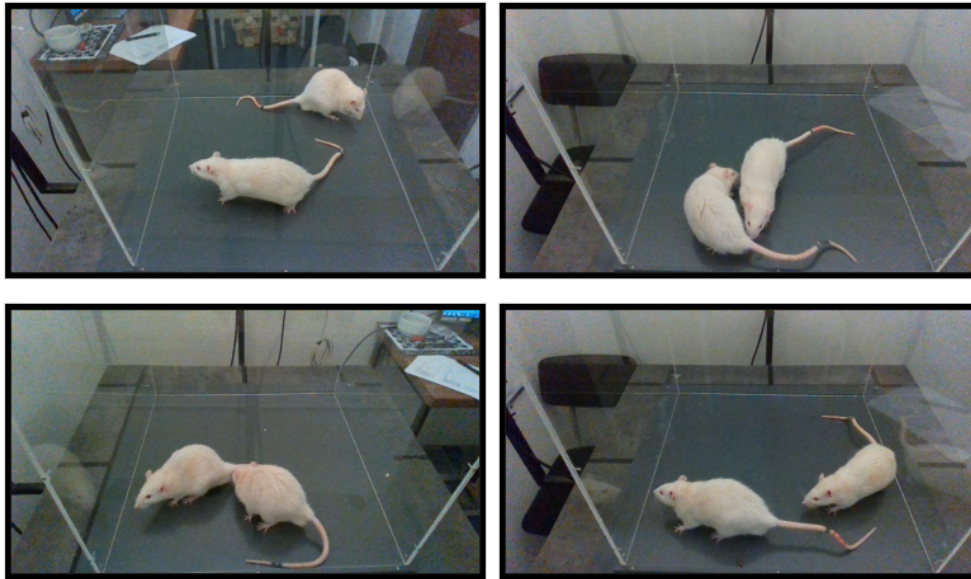


Figure 2.7. An exemplary frame from labeled sequences. The left upper image depicts [NS], and the left lower image depicts [OI]. The right upper image depicts [EX], and the right lower image depicts [AP].



Figure 2.8. An exemplary frame from labeled sequences. The left upper image depicts [PS], and the left lower image depicts [MT]. The right upper image depicts [DS], and the right lower image depicts [MM].

2.13.2 PAIR-R24M Dataset

The PAIR-R24M is a large dataset for multi-animal 3D pose estimation, which contains 24.3 million frames of RGB video (sampled at a rate of 30 Hz) and 3D ground-truth motion capture of dyadic interactions in laboratory rats. The dataset contains data from 18 distinct pairs of rats and 24 different viewpoints. Data were annotated with eleven behavioral labels and four interaction categories [42]. We will use up to 80,000 annotated sequences (15 consecutive frames) with corresponding 3D ground truth co-

ordinates from the dataset to evaluate and build the deep learning models, which we will then train on our datasets.

The behavioral interaction sequences from the PAIR-R24M dataset were classified into the following categories:

- Absence of Social Interaction [NS]
- Mutual Social Engagement-exploration [EX]
- Chase-synchronized locomotion [CS]
- Passive interaction-close proximity with no active exploration [PS]

A total of 75566 sequences were utilized: 48880 sequences for class [NS], 17677 sequences for class [EX], 8681 sequences for class [CS], and 328 sequences for class [PS].

■ 2.13.3 DLC Dataset

For training the DLC DLCRNet-MS5 model, we labeled 845 frames with nine markers for each rat (18 markers for each frame; refer to Section 2.6 to see the specific types of markers). Frames were derived from 10 subsets of videos – a total of 40 videos, with a similar portion for each of the camera views.

Chapter 3

Methods

We outline the methodologies applied in this study, focusing on the techniques developed or employed for recognizing and analyzing rat’s social interactions. We explore the classification of the rat’s actions and their initiators by methods defined as a set of mathematical rules (refer to Section 3.2.1 Rule-Based Model) and deep learning methods used, developed, or fine-tuned, such as graph-based models (ST GCN) or convolutional networks such as TSRJI-CNN and CNN-LSTM (refer to Section 3.3 for a detailed implementation). These methods process the obtained 3D point trajectories or corresponding image sequences.

We introduce tracking adjustments while projecting the markers into a 3D coordinate system to ensure a correct identity across the analyzed sequences (refer to Section 3.1). A comprehensive overview of the methods used for data pre-processing (such as spatial imputation or 3D reconstruction) can be found in the Data Section. To provide a visual summary leading to the acquisition of corrected sequences for the following sections, the flow chart diagram presented in Figure 3.1 outlines the entire process.

We develop the methods on datasets Rule-TgF344AD Pair 18, TgF344AD Pair 18, or Pair-R24M. We focus on denoting the most suitable methods and modalities for our acquired video subsets together with the corresponding frame-by-frame 3D coordinates and their concatenation into a multi-modal network (refer to Section 3.3.9). We then use the methods to analyze the acquired video subsets and statistically determine if there is a significant difference between the interactions and behavior of the rat types TgF344-AD and F344 (refer to Section 3.4). We analyze the actions determined by the Rule-Based model, acquiring time spent in the stricter actions, such as snout to tail-base contact, and the denoted multi-modal network for the more complex actions, such as approach or mutual exploration (the actions and dataset details are described in Section 2.13).

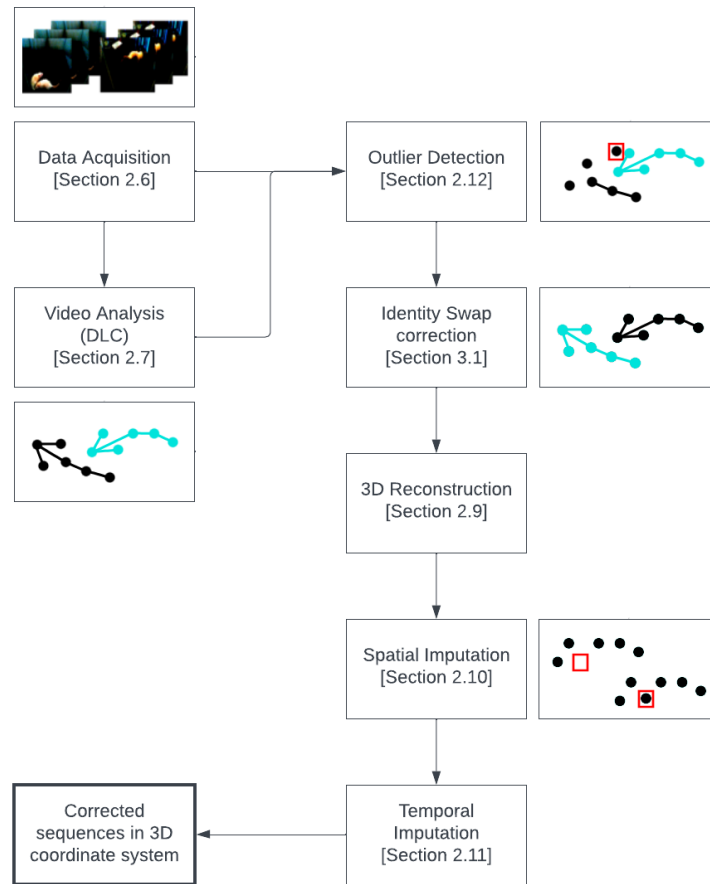


Figure 3.1. Flow Chart depiction of data acquisition, preprocessing, and methods leading to corrected sequences in 3D coordinate system, which are used as the input to the action recognition methods denoted below. The data pre-processing section is focused on reliable correction and imputation of the DLC estimations using the information from multiple camera views, 3D reconstruction, and imputation models.

3.1 Pose Estimation and Tracking in 3D

We used DeepLabCut (DLC) to estimate the markers and track the rat’s identity (as described in the Data Section). While spatial imputation techniques were employed to address inaccuracies in the estimated pose, solely shortening the analyzed video sequences (as described in Section 2.7) and manually re-identifying each sequence proved insufficient in handling potential identity swaps – as the problematic occlusion for DLC might occur at the start of the sequences.

We employed a combinatorial optimization technique that minimized the reprojection error to address the potential identity swaps. We aimed to determine the optimal identity assignment for each instance to minimize the reprojection error across all views. If DLC lost tracking in one camera view but maintained accurate tracking in the remaining views, we could correctly assign the sequence identities for that instance. The problem was defined as follows:

- A set of camera views $V = \{v_1, v_2, \dots, v_4\}$.
- A set of rat identities $R = \{r_1, r_2\}$.

Objective: The optimal assignment of rat identities to views that minimizes the reprojection error defined as $\min_{r \in R, v \in V} E(T(r, v), P_{2D})$, where E represents the total reprojection error function (defined in Equation 2.9.1), T represents the reprojections of the triangulated points for a given combination, and P_{2D} represents the original estimated coordinates.

The solution space consists of all possible combinations of rat identities across views, leading to 2^4 potential combinations. For each combination, we compute the total reprojection error and select the combination that yields the minimum error.

While this solution is valid for tracking the identity when DLC misassigns the identity in one of the views, if the identity swap occurs in two or more views simultaneously (such a scenario was observed to be much less probable but possible to occur), the solution fails (the total reprojection errors might be the same for two combinations simultaneously).

Another criterion was added (inspired by the DLC approach, which regards tracking as a global minimization problem – see Section 1.3). The criterion added was one such that it minimizes the Euclidean distance traveled by the identified rat between the previous instance (with assigned markers) and the instance being examined for a potential swap in the 3D coordinate system. The selected combination was then the one that minimized the total reprojection error and the traveled distance in the 3D coordinate system. The total Euclidean distance between instances to be minimized was defined as:

$$d_{total} = \sum_{n=1}^M \left(\left\| P_{n,j}^X(x, y, z) - P_{n,j-k}^X(x, y, z) \right\| + \left\| P_{n,j}^Y(x, y, z) - P_{n,j-k}^Y(x, y, z) \right\| \right) \quad (3.1.1)$$

Where $P_{n,j}^X$ denotes the position of marker n in instance j for rat X (similarly for rat Y), and k denotes the difference between instance j and the previous valid instance. The identity swap detection was experimentally evaluated in experiments section 4.5.

3.2 Action Recognition

3.2.1 Rule-Based Model

The rule-based model was designed to identify and categorize the behavior patterns of rats. More specifically, the model distinguishes between the two rats involved and determines the type of interaction based on predefined rules and states. The rule-based model can distinguish between actions denoted as [NS], [HH], [SB], [AP], and [PS] from the dataset Rule-TgF344-AD Pair 18. The primary inputs to this model are the 3D coordinates and a set of specially derived features. The features used are the following (features denoted with an apostrophe were derived for each rat):

Note: Throughout the following equations, P^X and P^Y will represent the markers for the two distinct rats identified as X or Y.

$$D_H = \{d_{h,1}, d_{h,2}, \dots, d_{h,N}\}$$

$$d_{h,i} = \left\| \frac{1}{3} \sum_{m=1}^3 P_i^X(x_m, y_m, z_m) - \frac{1}{3} \sum_{m=1}^3 P_i^Y(x_m, y_m, z_m) \right\| \quad (3.2.1.1)$$

Where x_m, y_m, z_m are the 3D coordinates representing the marker m , employed to obtain the center of the head of the given rat by computing the mean value of the first

three markers representing snout and ears. N is the number of instances (frames), and D_H represents a set of distances $d_{h,i}$ computed for each instance i .

$$\begin{aligned} D'_{SB} &= \{d_{sb,1}, d_{sb,2}, \dots, d_{sb,N}\} \\ d_{sb,i} &= \|P_i^X(x_s, y_s, z_s) - P_i^Y(x_b, y_b, z_b)\| \end{aligned} \quad (3.2.1.2)$$

Where x_s, y_s, z_s are the 3D coordinates representing the marker (snout) of the given rat X, and x_b, y_b, z_b are the 3D coordinates representing the marker (tail base) of the given rat Y. N is the number of instances (frames), and D'_{SB} represents a set of distances $d_{sb,i}$ computed for each instance i .

$$\begin{aligned} V'_R &= \{0, v_{r,2}, \dots, v_{r,N}\} \\ v_{r,i} &= \frac{1}{\tau} \left\| \frac{1}{M} \sum_{m=1}^M P_i^X(x_m, y_m, z_m) - \frac{1}{M} \sum_{m=1}^M P_{i-1}^X(x_m, y_m, z_m) \right\| \end{aligned} \quad (3.2.1.3)$$

Where $v_{r,i}$ is the speed of the rat X's head in an instance i , M is the number of markers representing the rat X's head (3), x_m, y_m, z_m are the 3D coordinates representing the marker m , τ is the time constant representing time between instances calculated as $\tau = 1/f$, where $f = 30$ Hz, and V'_R represents a set of velocities $v_{r,i}$.

$$\begin{aligned} R' &= \{\vec{r}_{X,1}, \vec{r}_{X,2}, \dots, \vec{r}_{X,N-k}\} \\ \vec{r}_{X,i} &= \frac{1}{k} \sum_{f=i}^{i+k-1} \frac{\bar{P}_{i+f+1}^X - \bar{P}_{i+f}^X}{\|\bar{P}_{i+f+1}^X - \bar{P}_{i+f}^X\|} \end{aligned} \quad (3.2.1.4)$$

Where $\vec{r}_{X,i}$ is the directional unit vector of the rat for a given window starting in an instance i , k is the window size, $\bar{P}_{i+f+1}^X, \bar{P}_{i+f}^X$ are the mean values of the rat's markers in the 3D coordinate system in an instance $i+f+1$ and $i+f$, respectively, and R' represents a set of $\vec{r}_{X,i}$.

$$\begin{aligned} RM &= \{r\vec{m}_{XY,1}, r\vec{m}_{XY,2}, \dots, r\vec{m}_{XY,N}\} \\ r\vec{m}_{XY,i} &= \frac{\bar{P}_i^X - \bar{P}_i^Y}{\|\bar{P}_i^X - \bar{P}_i^Y\|} \end{aligned} \quad (3.2.1.5)$$

Where $r\vec{m}_{XY,i}$ is the directional unit vector representing the direction between rat X and Y in an instance i , \bar{P}_i^X, \bar{P}_i^Y are the mean values of the rat's markers in the 3D coordinate system in an instance for rat X and Y, respectively, and RM represents a set of $r\vec{m}_{XY,i}$.

$$\begin{aligned} SD' &= \{\sigma_{X,1}, \sigma_{X,2}, \dots, \sigma_{X,N-w}\} \\ \sigma_{X,i} &= \frac{1}{M} \sum_{m=1}^M std(P_{m,i:i+k}^X) \end{aligned} \quad (3.3.1.6)$$

Where $std(P_{m,i:i+k}^X)$ is the standard deviation of the rat's marker m for a given window starting in an instance i , k is the window size, M is the number of markers, and SD' represents a set of mean standard deviation values for an instance i .

Five primary rules describe the model. To summarize, two rules are based on the distance determining the contact of markers (snout, tail base, head centers) [SB]', [HH] (refer to equations 3.2.1.9 and 3.2.1.10, respectively). The rule described by equation 3.2.1.7 is based on the directions and speed of the rat to determine the approach [AP]'

with a subsequent distance-based rule to determine a successful approach – refer to equation 3.2.1.8. Lastly, a rule described by equation 3.2.1.11 is based on the standard deviation of markers to determine a passive contact of the rats [PS]. The rules are defined as follows:

$$AP_{i:i+k} = \begin{cases} X & \text{if } R^X_i RM_i > R^Y_i(-RM_i) \text{ and } V^X_{R_i} > t_v \\ Y & \text{if } R^X_i RM_i < R^Y_i(-RM_i) \text{ and } V^Y_{R_i} > t_v \\ 0 & \text{otherwise} \end{cases} \quad (3.2.1.7)$$

Where t_v is the speed threshold for the approach of the rat to be valid in an instance i , the rest of the features described in the rule are detailed in the equations above (see 3.2.1.4 and 3.2.1.5). The decision criterion for identifying the approaching rat (identity X and Y) is determined by comparing the magnitudes of the calculated dot products. The dot product measures the alignment between each rat by multiplication of each rat's directional vector components and the components of the vector pointing from one rat to the other. A larger dot product value signifies a higher alignment between a rat's directional vector and the vector connecting the two rats. We describe an example of the directional vectors in Figure 3.2.

Consequently, the rat with the larger dot product was considered to be actively approaching the other (while also satisfying the determined speed threshold). For the approach criterion to be valid, a condition considering mutual distances had to be satisfied as follows:

$$AP_{i:i+k} = \begin{cases} X \text{ or } Y & \text{if } \left\| (\bar{P}_f^X) - (\bar{P}_f^Y) \right\| + t_{ma} < \left\| (\bar{P}_i^X) - (\bar{P}_i^Y) \right\| \\ & \text{and } \left\| (\bar{P}_i^X) - (\bar{P}_i^Y) \right\| > t_{ms} \end{cases} \quad (3.3.1.8)$$

Where $\left\| (\bar{P}_f^X) - (\bar{P}_f^Y) \right\|$ and $\left\| (\bar{P}_i^X) - (\bar{P}_i^Y) \right\|$ are the Euclidean distances between the mean value of rat X and Y for instance $f = i + k$ and i , respectively. t_{ma} is a threshold value determining the minimal Euclidean distance traveled for the approach criterion to be valid, and t_{ms} is a threshold value determining the minimal Euclidean distance between rats before the approach starts.

$$SB_i = \begin{cases} X & \text{if } D^X_{SB,i} < t_{sb} \\ Y & \text{if } D^Y_{SB,i} < t_{sb} \\ XY & \text{if } D^X_{SB,i} < t_{sb} \text{ and } D^Y_{SB,i} < t_{sb} \\ 0 & \text{otherwise} \end{cases} \quad (3.2.1.9)$$

Where t_{sb} is the minimal Euclidean distance threshold between snout and tail base markers for the contact of the markers to be valid in an instance i , the rest of the features are described in equation 3.2.1.2.

$$HH_i = \begin{cases} XY & \text{if } D_{H,i} < t_h \\ 0 & \text{otherwise} \end{cases} \quad (3.2.1.10)$$

Where t_h is the Euclidean distance threshold that needs to be reached between the head centers of both rats for the contact of the markers to be valid in an instance i , the rest of the features are described in equation 3.2.1.1.

$$PS_{i:i+k} = \begin{cases} XY & \text{if } \left(\left\| (\bar{P}_f^X) - (\bar{P}_f^Y) \right\| \text{ and } \left\| (\bar{P}_i^X) - (\bar{P}_i^Y) \right\| \right) < t_{mp} \text{ and} \\ & \left(\bar{V}^X_{R_{i:i+k}} \text{ and } \bar{V}^Y_{R_{i:i+k}} \right) < t_{vp} \\ 0 & \text{otherwise} \end{cases} \quad (3.3.1.11)$$

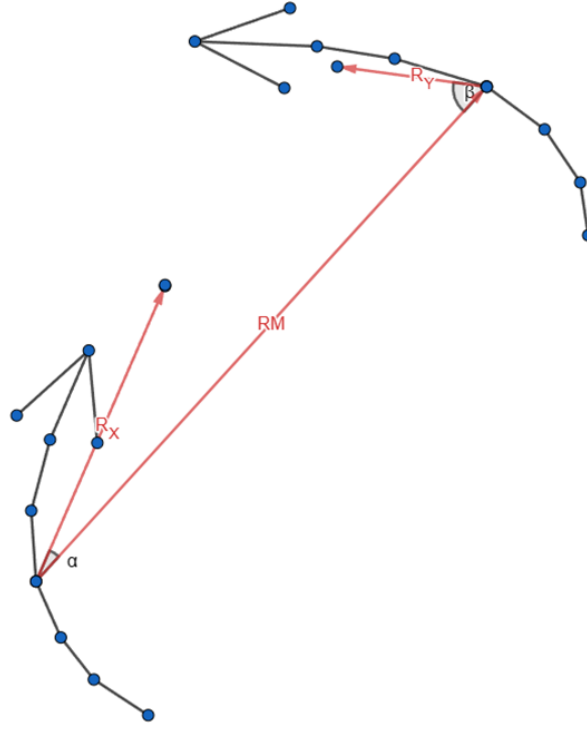


Figure 3.2. The image representation of the directional vectors in the approach rule denotation in equation 3.2.1.7

Where t_{vp} is the maximum speed threshold for mean velocities (\bar{V}_R^X) and (\bar{V}_R^Y) across the given window ($i:i+k$) for both rats, t_{std} is the maximum standard deviation threshold that needs to be between head centers of both rats for the contact of the markers to be valid in an instance i , and t_{mp} is the maximum Euclidean distance threshold between the center of both rats for the rats to stay in passive contact.

The arrays determining a label for each class (denoted in the first paragraph of Section 3.2.1) for each instance have been merged in the following order: firstly, the instances are classified by label [PS], secondly [AP], followed by [HH], and lastly [SB]. The remaining instances with an unassigned class are classified as no social interaction [NS]. This succession was determined by the importance of given classes and observed succession in the sequences (the approach of the rat usually preceded the snout and tail base contact).

The optimal threshold values $T^* = \{t_{mp}, t_{std}, t_h, t_{sb}, t_{ms}, t_{ma}, t_v, t_{vp}\}$ and two window sizes $W^* = \{w_{ap}, w_{ps}\}$ for the approach and passive interaction determination have been found by a Bayesian optimization search with an objective function based on the weighted F_1 score evaluation of the model on prelabeled dataset Rule-TgF344-AD Pair 18. For the results of the model, refer to the experiments section 4.6. To specify the window sizes, w_{ap} denotes the window size (k) used throughout the determination of [AP], and w_{ps} denotes the window size (k) used throughout the determination of [PS].

The optimization was employed in MATLAB by built-in ‘bayesopt’, with 300 iterations. The F_1 score weight w for each class was computed as $w = 1 - \frac{k}{N}$, where k represents the occurrence of true labels for the given class, and N represents the total number of labels. This approach was chosen because the classes being determined are underrepresented compared to [NS] (class defining no social behavior for given in-

stances). The weights were subsequently normalized. The total weighted F_1 score was defined as $F_{1_w} = \sum_{i=1}^c F_{1_i} w_i$, where c denotes the number of classes.

3.2.2 Rule-Based Model Label Adaptation

To have a comparable Rule-Based Model with a dataset TgF344-AD Pair 18, we adjust [HH] and [SB] rules denoted in the strict Rule-Based Model. To adapt 3.2.1.10 [HH] to capture class [EX], we denote that the snouts of both rats are close to any marker of the other rat’s markers under threshold t_{ex} . To adapt equation 3.2.1.9 [SB] to capture class [OI], we denote that the distance of the snout of a given rat is close to any marker of the other rat’s markers under threshold t_{oi} . Rules for approach [AP] and passive interaction [PS] keep their form as denoted in Section 3.2.1. We denote a new rule for class [DT] – an adaptation of the approach rule, where the directional vector faces the opposite direction with detach speed threshold t_{vd} and two new distance thresholds are denoted as t_{md} . Threshold t_{md} is the minimal distance that needs to be traveled from the other rat, and t_{ds} is the threshold of the starting distance of the action (following the concept of equation 3.2.1.8). The window size threshold for detach is denoted as w_{dt} . To classify one rat mounting [MT] (rearing above the other), we denote the rule as follows:

$$MT_i = \begin{cases} X & \text{if } P_{i,HX}(z) > t_z \text{ and } P_{i,HY}(z) < t_z \text{ and } |P_{i,mX} - P_{i,mY}| < t_{mt} \\ Y & \text{if } P_{i,HY}(z) > t_z \text{ and } P_{i,HX}(z) < t_z \text{ and } |P_{i,mX} - P_{i,mY}| < t_{mt} \\ 0 & \text{otherwise} \end{cases} \quad (3.2.2.1)$$

Where $P_{i,HX}$ is the center of rat X’s head in an instance i , as denoted in 3.2.1.3 (using the z coordinate), $P_{i,HY}$ is the center of rat Y’s head in an instance i , as denoted in 3.2.1.3 (using the z coordinate), $|P_{i,mX} - P_{i,mY}|$ is the Euclidean distance between the rat’s mean positions, t_z is the threshold denoting the minimal height a rat needs to rear to, and t_{mt} is the minimal distance between the rats.

To classify one rat mimicking the rearing of the other rat [MM], we denote the rule as follows:

$$MM_i = \begin{cases} XY & \text{if } P_{i,HX}(z) > t_{mm} \text{ and } P_{i,HY}(z) > t_{mm} \\ 0 & \text{otherwise} \end{cases} \quad (3.2.2.2)$$

Where t_{mm} is the threshold in the z -axis denoting the minimal height both rats need to reach. The coordinates were aligned to the XY plane denoted by the arena corners so that the z -axis is the vertical axis of the 3D coordinate system, following a Rodriguez transformation denoted in 3.2.3.4a, before applying the rules above.

The added thresholds to the adaptation of the model follow the same evaluation protocol denoted in 3.2.1, using the dataset TgF344-AD Pair 18 (refer to the experiments section 4.7).

3.3 Action recognition based on Deep Learning Models

We employed and implemented six methods based on the techniques suitable to our action recognition task (ST GCN, CNN, etc.) fully outlined in the introduction (refer to Section 1.4) and the modalities derived from our datasets. The modalities are represented as sequence windows of 3D coordinates (skeleton of one rat defined through nine markers in one time step), images from camera views, and features derived from the coordinates. A sequence window depicts 15 consecutive steps.

The methods section below outlines the implementation of ST GCN, TSRJI-CNN, Reference Pivot TSRJI-CNN, CNN – LSTM model based on sequences of images from four camera views at the scene, 3D CNN model based on motion cuboids defining the sequence (refer to Section 1.4.1), and a 1D CNN learning from the derived features sequence vectors. Each model will be discussed as its method with its corresponding data processing before connecting the denoted methods into a multi-modal network. The models are implemented using the PyTorch 2.0.1 library with Python 3.10. Throughout the models, we use the training protocol defined in Section 3.5.

3.3.1 ST GCN Model

We denote graphs $G_s(V, E)$ and $G_t(V, E)$ representing the spatial relationships of the rat’s skeletons and the temporal dependencies of a given node (refer to Figure 3.3 for an image representation), respectively, where V is a set of n body joints represented as nodes and E is a set of m bones represented as edges.

We define matrix $A_s \in \{0, 1\}^{2n \times 2n}$ as the adjacency matrix of the spatial graph G_s , where $A_{s,i,j} = 1$ represents a connection between nodes i and j ($n = 9$, multiplied by 2 for the representation of both rats), 0 otherwise. Similarly, we define matrix $A_t \in \{0, 1\}^{t \times 2n}$ as the adjacency matrix of the temporal graph G_t , where t is the sequence window length – defined as a clip-level sequence window of 15 steps because the movement of the rat’s actions is rather fast. Node $X \in \mathfrak{R}^{t \times 2n \times 3}$ represents the position of an n -th body joint in a 3D coordinate system at time step t , I_s represents the edge of the spatial graph G_s and I_t represents the edge of the temporal graph G_t .

Two graph topology configurations were tested – a configuration c_1 with determined connections for the ST GCN, where we set up a specific spatial and temporal topology of the graphs, and configuration c_2 with fully connected data-driven topology for the spatial graph G_s (where the adjacency matrix of G_t remains the same as for c_1). A comparison of the methods is described in the experiments section 4.9. The topology configuration c_1 is defined by an illustration in Figure 3.3. The spatial graph G_s is presented to the network as an undirected graph, while the temporal graph G_t is a directed graph, where the direction corresponds to the time flow of the sequence.

The ST GCN consists of blocks with the adjacent matrix A_s and A_t , extracting the spatial and temporal features. We use the GCN layer from Pytorch Geometric Module function `GCNConv` [30], defined in equation 1.4.1, where the adjacent matrix is A_t or A_s , depending on the temporal stream t or spatial stream s . X_{in} represents the input features (representation of the body joints as defined above).

The spatial stream s consists of two GCN layers with the adjacent matrix A_s followed by ReLU activations. The first layer maps an input feature vector X to a 64-dimensional feature vector X_{s1} . The second layer further transforms X_{s1} into a 128-dimensional feature vector X_{s2} . A dropout layer is added between the hidden states with a 20% dropout probability.

$$\begin{aligned} X_{s1} &= \text{ReLU}(\text{GCNConv}(X, A_s, 64)) \\ X_{s2} &= \text{ReLU}(\text{GCNConv}(X_{s1}, A_s, 128)) \end{aligned} \tag{3.3.1.2}$$

The temporal stream t consists of two GCN layers with the adjacent matrix A_t followed by ReLU activations. The first layer maps an input feature vector X to a 64-dimensional feature vector X_{t1} . The second layer further transforms X_{t1} into a 128-dimensional feature vector X_{t2} . A dropout layer is added between the hidden states

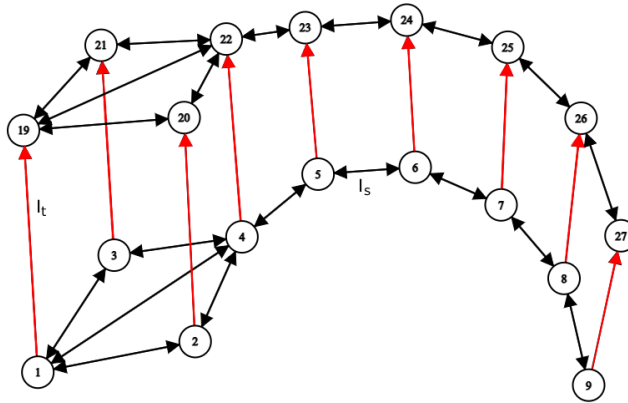


Figure 3.3. A definition of topology for graphs G_s and G_t for one rat in two exemplary time steps (the sequence window consists of 15 such steps) in configuration c_1 . The connections between nodes 1 to 9 and 19 to 27 define the topology of G_s in each time step. An undirected edge I_s represents the spatial connection (example: between nodes 5 and 6, edge colors: black), and the directed edge I_t (example: presented between nodes 1 and 19, edge colors: red) represents the temporal connection between time steps. The configuration c_2 is a fully connected graph across the spatial dimension (for both rats) but is not illustrated due to its complexity.

with a 20% dropout probability.

$$\begin{aligned} X_{t1} &= \text{ReLU}(\text{GCNConv}(X, A_t, 64)) \\ X_{t2} &= \text{ReLU}(\text{GCNConv}(X_{t1}, A_t, 128)) \end{aligned}$$

The derived features of streams s and t , X_{s2} and X_{t2} , respectively, are concatenated and followed by two linear layers with ReLU activations, a 20% probability dropout layers, and a final linear layer.

We also denote a model architecture augmentation as ST GCN-s. The architectural difference in ST GCN-s is that the streams are not parallel [43] but are put in a series [44] consisting of four GCN modules. Each module consists of two GCN layers, starting with the temporal topology applied GCN layer, followed by the spatial topology GCN layer [44]. The output channels of each module are 64, 128, 256, and 512, followed by fully connected layers used as for configurations c_1 and c_2 . Training and validation of the models are described in the experiments section 4.9. Both architectures of the ST GCN layers are depicted in Figure 3.4.

Note: We also made several augmentations, such as representing the node of the graph as a polar coordinate with respect to the other markers and its 3D coordinates, resulting in a 30-dimensional vector for a node. The ST GCN then managed to learn the proposed actions. However, the method did not outperform the TSRJI-CNN model nor the CNN-LSTM (defined in Sections 3.3.3 and 3.3.5, respectively) and was not as computationally effective. Refer to the experiments section for the evaluation of the mentioned methods.

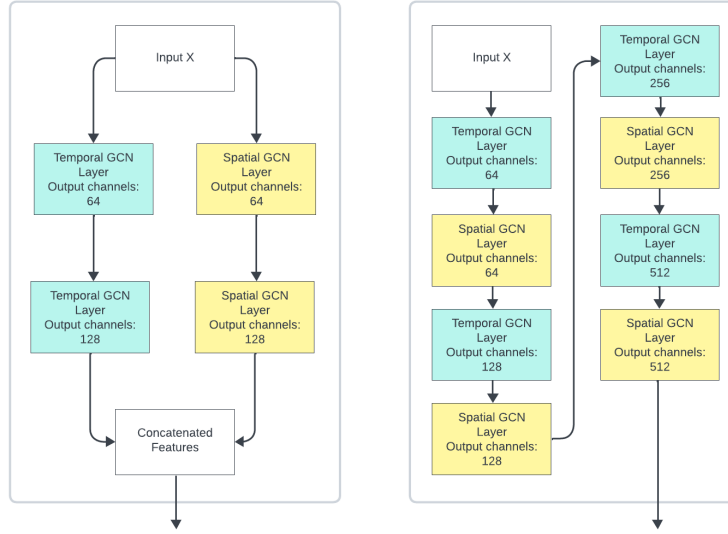


Figure 3.4. Illustration of the ST GCN layers preceding the fully connected layers. The left diagram depicts the architecture of the parallel concept of the ST GCN, and the right diagram depicts configuration c_3 with the ST GCN layers put in series.

3.3.2 Coordinate Data Augmentation

To mitigate overfitting in 3D coordinates-based models, we introduced two augmentation methods on the coordinates – rotation, translation, and mirroring. The classification of actions should be invariant to the rat’s position within the arena. Initially, we align the Z-coordinate to be perpendicular to the plane of the arena. Three non-collinear points define the arena plane, the triangulated corners of the arena, denoted as P_1 , P_2 , and P_3 . The normal to the plane, n , is computed as $n = \frac{-v_1 \times v_2}{\|v_1 \times v_2\|}$ where $v_1 = P_2 - P_1$ and $v_2 = P_3 - P_1$. The rotation axis r , and rotation angle θ , are calculated using the equations $r = n \times z$ and $\theta = \arccos(n \cdot z)$, respectively, with $z = [0, 0, 1]^T$. The rotation matrix R is defined by Rodrigues’ rotation formula [45]:

$$R = \cos(\theta)I + \sin(\theta)N_{\times} + (1 - \cos(\theta))rr^T \quad (3.3.2.1)$$

Where N_{\times} represents the skew matrix of r .

Each point P representing the rat’s coordinates is then rotated, resulting in $P' = RP$ (we also align the corner points of the arena from P'_{C1} to P'_{C4}). This allows us to augment the training dataset represented via 3D coordinates. The augmentation is applied with a 50% probability for each sequence being fetched for the batch. We define the rotation as follows:

From the aligned arena corners P'_{C1} to P'_{C4} , we compute the center of the arena as a midpoint of the minimum and maximum x,y coordinates of these corners as:

$$C(x, y) = \left(\frac{\min(P'_C(x)) + \max(P'_C(x))}{2}, \frac{\min(P'_C(y)) + \max(P'_C(y))}{2} \right) \quad (3.3.2.2)$$

For rotation augmentation, each point P' representing the rat’s coordinates is translated to a new coordinate system centered at C and then rotated around the Z-axis using a rotation matrix $R_z(\rho)$, where ρ is the rotation angle uniformly generated between 0 and 2π . After the rotation around center C , the point is translated back to its original coordinate system. We add a translation factor ϕ (uniformly generated between

0 and 1) into the equation to augment the translation. The rotation and translation augmentation on a point is then defined as:

$$P'' = R_z(\rho)(P' - C) + \phi C \quad (3.3.2.3)$$

For mirroring augmentation, with a 50% probability, we reflect the coordinates across the vertical or horizontal axis passing through the center C . The reflection along the X-axis (mirror X) and Y-axis (mirror Y) can be represented as $P_x'' = 2C_x - P_x'$ and $P_y'' = 2C_y - P_y'$, respectively.

These operations ensure that the augmented coordinates stay within the bounds of the arena while introducing variation in the spatial positioning of the rats. The same augmentation parameters (uniformly generated ρ , mirroring chance, and ϕ) are applied to the entire sequence window, to all points, for both rats. Additionally, we introduce a 50% chance of swapping sets of points in the sequence coordinate matrix for a given rat with those of the other rat. This approach ensures that the 3D coordinate-based models are invariant to the identity of the rats, eliminating bias towards a specific rat being the reason for the action classification.

3.3.3 TSRJI-CNN Model

Tree Structure Reference Joints Image (TSRJI) CNN model processes a TSRJI using a series of convolution layers. The TSRJ image is constructed based on a predefined tree structure with reference joints, each representing a sequence of a given action and the connection between the tree structure and a reference joint. We chose the snout, second spine point, and tail base for our reference joints for each rat. In the TSRJ image, we encode the change of spatial relationships through the sequence of the points of one rat to the reference point on the other rat. This results in six images of 15×9 pixels, where each pixel represents distance and rotational information between two points in the XY-plane and along the Z-axis (polar coordinates with the denoted reference point). Given a sequence window matrix of coordinates $X \in R^{t \times 2n \times 3}$ and the topology of the tree structure defined, we denote the construction of TSRJI as follows:

$$\begin{aligned} d_{t,ij} &= \|X_{t,i}(x, y, z) - X_{t,j}(x, y, z)\| \\ \alpha_{t,ij} &= \frac{\cos(\arccos(y_{t,j} - y_{t,i}, x_{t,j} - x_{t,i})) + 1}{2} \\ \beta_{t,ij} &= \arccos\left(\frac{z_{t,j} - z_{t,i}}{d_{t,ij}}\right) \cdot \frac{180}{\pi} \end{aligned} \quad (3.3.4.1)$$

Where, for a given time step t and joints i, j , $d_{t,ij}$ is the Euclidean distance between joints, $\alpha_{t,ij}$ is the azimuthal angle representing the orientation of one point relative to another in the horizontal plane (XY), and $\beta_{t,ij}$ represents the polar angle between the Z-axis and the vector between the two joints. The pixel I in position t and n of the TSRJ image is then defined as $I_{t,n} = (R = \frac{d_{t,ij}}{d_{\max}}, G = \alpha_{t,ij}, B = \frac{\beta_{t,ij}}{180})$, where R, G, B represent the standard RGB color space values for the pixel. We present the RGB representation in Figure 3.5.

The tree structures are organized based on a systematic pattern: each structure connects every joint of one rat to a single reference joint of the other rat. We form three structures per rat, each focusing on one of the three reference joints. This approach results in six unique configurations, where the first set of three structures connects each joint of Rat X to the snout, second spine point, and tail-base of Rat Y, respectively. The second set mirrors this approach, connecting each joint of Rat Y to the corresponding

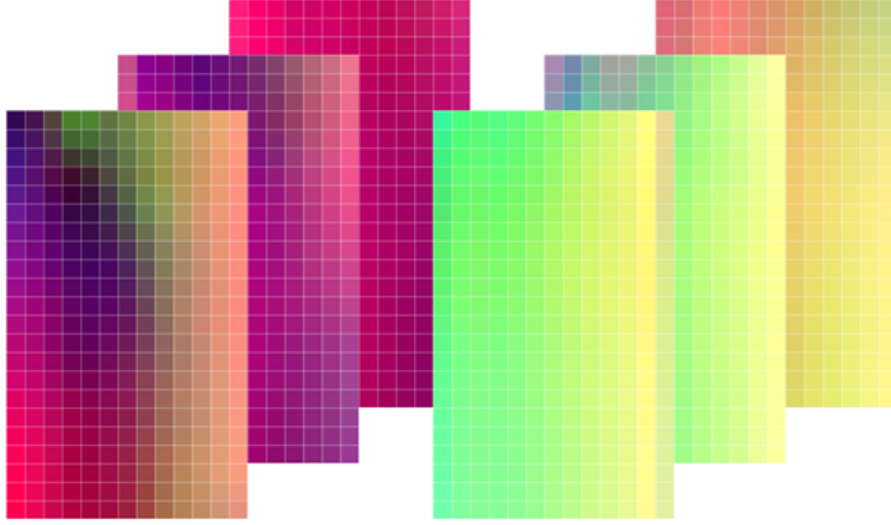


Figure 3.5. Example of the derived TSRJ images representing a sequence window for each rat X (left) and Y (right) reference points.

reference joints of Rat X. Each of the six derived TSRJ images (M) is then passed into a CNN with two 2D convolution layers and one pooling layer, defined as:

$$X_1 = \text{ReLU}(W_1 * X_{in} + b_1) \quad (3.3.4.2)$$

$X_{out,M} = \text{MaxPool}(\text{ReLU}(W_2 * X_1 + b_2))$, where W are the trainable weights and b is the bias. The first convolution layer takes an input with 3 channels and produces an output with 16 channels, using a 3×3 kernel with stride 1 and padding 1. The second convolution layer takes the 16-channel input and outputs a 32-channel feature map using a 3×3 kernel with stride 1 and padding 1. The pooling layer is a 2×2 max pooling operation with stride 2 and no padding. The derived features for each image M are then concatenated and passed to a linear layer with a ReLU activation and a 20% dropout layer, followed by a final linear layer. Training and validation of the model are described in experiments section 4.10. The architecture is depicted in Figure 3.6

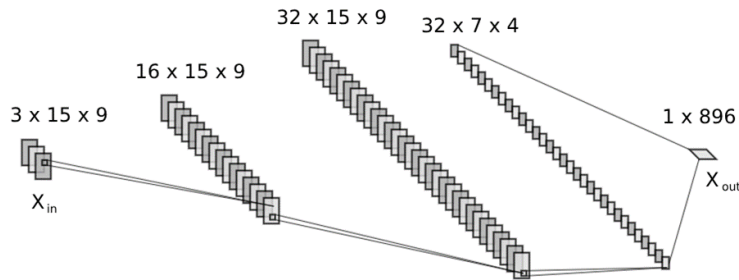


Figure 3.6. Illustration of the convolution and pooling layers of the TSRJI-CNN network for one TSRJ image.

Note: We also test the implementation without normalizing the polar coordinates and presenting them as an RGB image. The results were similar to the RGB representation (refer to Experiment Section 4.10), but the loss on training and validation sequences was diverging, as opposed to the normalized representation.

3.3.4 Pivot TSRJI-CNN Model

Following the experiments evaluating the 3D coordinate methods (ST GCN, TSRJI-CNN), we denote a modification to the TSRJI-CNN model (as the GCN-based method proved insufficient, and the TSRJI otherwise). The modification to the TSRJI-CNN model follows a concept of a mutual action recognition method on skeletal data based on a pivotal point and coordinate transformation, proposed by Shian-Yu Chiu, 2021 [46]. We adjust the sequence matrix of coordinates $X \in R^{t \times 2n \times 3}$ so that a sequence is centered to the center of the arena as denoted in equation 3.3.2.2 (with an addition of the Z coordinate). The center is denoted as $C'(x, y, z) = [x = 0, y = 0, z = 0]$ in the new coordinate system. We denote this center as a pivot to all other points in the matrix (of both rats) and compute the polar coordinates with the pivot set as the reference point. The computation of the TSRJI then follows the method in Section 3.3.3, but inputs only two image maps (for both rats), as we have only one reference point, allowing the model to learn the action features with respect to one point. As a CNN model to classify the Pivot TSRJI images, we use the model denoted in Section 3.3.3. Training and validation of the model are described in experiments section 4.11.

3.3.5 CNN-LSTM Model

The CNN-LSTM model processes the sequence window images of each camera view to classify the actions. We fine-tuned a ResNet CNN to derive the spatial features. We concatenate the features of all views across the sequence window and pass them to an LSTM layer to process the temporal dependencies. Based on datasets sizes and computational resources, the backbone of ResNet-18 was chosen as a default pre-trained network. Prior to the CNN feature extraction, the images undergo resizing and normalization. We resize the input image of size 1280×720 to a fixed size of 112×112 with bilinear interpolation (the ResNet-18 is trained on 224×224 ImageNet dataset with bilinear interpolation, however, due to computational resources, a smaller fixed size was set) [47]. The normalization was based on the standardly used mean and standard deviation values for the ResNet-18 fine-tuning, which are derived from the ImageNet dataset and defined as $\mu_{ImageNet} = [0.485, 0.456, 0.406]$, which represents the mean value of ImageNet, and $\sigma_{ImageNet} = [0.229, 0.224, 0.225]$, which represents the standard deviation of ImageNet (RGB channels) [47]. We present the image representation passed into the CNN in Figure 3.7.

From the ResNet18, we output a 512-dimensional feature vector for each of the views for each image in the sequence. Those spatial features are further concatenated into a 2048-dimensional feature vector X_M (images from four views) for each time step in the sequence window. We then pass X_M to an LSTM layer with 512-dimensional output X_{out} , with a subsequent mean of the output weights X_{out} , capturing the whole sequence window, defined as:

$$X_{out} = \frac{1}{T} \sum_{t=1}^{T=15} LSTM(X_M, 128) \quad (3.3.6.1)$$

Where T is the sequence window length, X_{out} is then passed to a 20% probability dropout layer, followed by a linear layer with a leaky ReLU activation, a linear layer with ReLU activation, and a final linear layer. Training and validation of the model are described in experiments section 4.12.

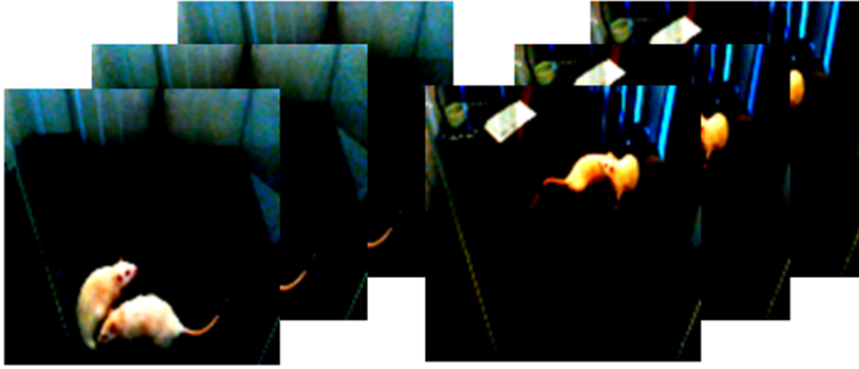


Figure 3.7. Example of the preprocessed input images to the CNN-LSTM model. The example depicts three images of the sequence window of camera view 3 (left) and camera view 4 (right).

3.3.6 View Image Data Augmentation

We introduced augmentation to the image tensors to prevent overfitting in the camera image-based models, using the Torchmetrics 1.1.2 library. We adjust rotation, brightness, and contrast augmentations to the images, with a 50% chance for the augmentation to be applied while fetching the sequence for the model. The same augmentation is applied across the entire sequence window for a given view. The augmentation parameters r (rotation angle), C (contrast factor), and B (brightness factor) are uniformly generated within the following ranges: $r \in (-15, 15)$ and $C, B \in (0.85, 1.15)$.

3.3.7 3D CNN Model

We employed a 3D convolutional network to process the spatial and temporal dependencies of the image sequence windows. The approach followed the CNN-LSTM model (refer to 3.2.6) – we preprocessed the images and passed the sequence windows of images from the four camera views into a backbone of 3D ResNet-18 – an augmented variant of the ResNet-18 designed to perform convolution in a 3D space, replacing the LSTM layer. The difference from the LSTM approach was in representing the sequence window as a motion cuboid, following the equation in 1.4.2. For each camera view, we computed a motion cuboid of the sequence window as:

$$X_{in,t,1-4} = |I_{t,1-4} - I_{t+1,1-4}| \quad (3.2.7.1)$$

Where I denotes the RGB image for the given view 1 to 4, and t denotes the time step in the sequence window. The resulting sequence window is shorter by one time step. Motion cuboids $X_{in,1-4}$ are then passed to a 3D ResNet-18. The resulting features are concatenated and passed to a linear layer with a ReLU activation, followed by a 20% probability dropout layer and a final linear layer. Training and validation of the model are described in experiments section 4.13.

3.3.8 1D CNN Model

To utilize the sets of features D_H (head-to-head distance), D_{SB} (snout-to-tail-base distance), and V_R (head speed) we denoted in 3.2.1, we employ a 1D CNN model. Each feature set is passed as a 15×1 vector into a 1D convolutional layer for a given sequence window. The feature matrix F is defined as $F = [D_H \quad D_{SB}^X \quad D_{SB}^Y \quad V_R^X \quad V_R^Y]$.

The convolution layer takes an input with 1 channel and produces an output with 16 channels, using a kernel of size 3 with padding 1. The derived features for each feature vector $F_{out,x}$ are then concatenated and passed to a linear layer with a ReLU activation, followed by a 20% probability dropout layer and a final linear layer. Training and validation of the model are described in experiments section 4.14.

3.3.9 Multi-Modal Model

We denote a multi-modal model to connect the qualities of the successful methods described above. Based on the experiment results (refer to Section 4.8), we combine the TSRJI-CNN model and the CNN-LSTM model. We approach the mentioned model's combination by three distinct strategies: model configurations c_1 , c_2 , and c_3 .

In configuration c_1 , we concatenated the feature output of each modality – the TSRJI-CNN feature output and the CNN-LSTM feature output right after the first fully connected layer of the models. The concatenation is followed by three fully connected layers, with leaky ReLU activation, ReLU activation, and a final layer, respectively, with two 20% dropout layers in between the fully connected layers.

In configuration c_2 , we omit the final fully connected layers and concatenate the features into a 16-dimensional vector (eight final weights for each modality). The vector is passed to a fully connected layer with the same output size (16) to assess the concatenated weights with a leaky ReLU activation, a 10% dropout layer, and then passed into the final fully connected layer.

In configuration c_3 , we use the TSRJI-CNN model and the CNN-LSTM model with pre-trained weights (the choice of the pre-trained weights was based on the least validation loss throughout the training of the models – refer to experiments section 4.8). We then omit the final fully connected layers of each of the models and retrain the concatenated features from the pre-trained models (vector of size $128 + 64$) with a fully connected layer with a ReLU activation and a 20% dropout layer, followed by a final fully connected layer. Each configuration is reported in experiments sections 4.15 and 4.16.

3.3.10 Action Initiator classification

To classify the initiator of actions predicted as [OI], [AP], [DT], and [MT] (for the given sequence window), we train an individual model for each of the classes. As for the model for the initiator classification, we choose the denoted method TSRJI-CNN (refer to Section 3.3.3) based on the Experiment 4.10 results and the fact that the TSRJ image carries the information about the rat's identity (as opposed to the CNN-LSTM view image-based model). The classification task for the model is then a binary classification problem. We kept the identity swap random transformation – in that case, we changed the resulting label accordingly. The results of the experiment are reported in experiments section 4.18.

3.3.11 Human Predictions

We prepare a labeling GUI to assign and acquire human predictions to compare the final method denoted to classify actions on TgF344AD Pair 18. The labeling process involved labeling two subsets - AD1AD3, six months, and AD3AD4, ten months. The person assigned the labeling task was shown and explained the examples of actions to be classified as described in Section 2.13.1. The person assigns a label to a hundred sequence windows (for each window viewing 15 consecutive frames of camera views 1

and 2 - one action for half a second of the video subset) from each subset (from time 5 seconds to 55 seconds of the video subset).

We calculated three types of accuracy for the comparison of the human and model predictions. Firstly, we denote accuracy A_{c1} as:

$$A_{c1} = \frac{1}{N} \sum_{i=1}^N \iota(y_i = \hat{y}_i) \quad (3.3.11.1)$$

Where N is the total number of sequence windows, y_i is the human label for a sequence window i , \hat{y}_i is the model's prediction for a sequence window i , and ι is the indicator function that returns one if the argument is true and 0 otherwise. Because we have visually inspected situations where the labeler or model differed by one sequence window - for example, the model predicted [AP] a sequence window earlier, while the person noted the approach from a closer distance, we introduced latency-based accuracy A_{lb} as:

$$A_{lb} = \frac{1}{N} \sum_{i=1}^N \max_{j=i-L, i+L} \iota(y_i = \hat{y}_j) \quad (3.3.11.2)$$

Where L represents the latency tolerance in terms of sequence window count. We also denote the event-based occurrence accuracy A_{eb} for both correct and incorrect predictions, comparing the frequency of occurrence of each class, as:

$$A_{eb} = 1 - \frac{1}{N} \sum_{k=1}^M |f_k^H - f_k^M| \quad (3.3.11.3)$$

where M is the number of classes, f_k^H and f_k^M are the frequencies of class k in the human labels and model predictions, respectively. The comparison results of 200 sequence windows (N) are described in experiments section 4.19.

3.4 Data Analysis

We use statistical methods to analyze the different rat types within our dataset TgF344 AD Pair 18 (TgF344 AD (AD) and F344 (WT)). We differentiate between the age groups of six months and ten months. The details of the experiment setup are described in Section 2.4. We use the derived model MMc_3 trained on the labeled part of TgF344 AD Pair 18 (see section 2.13.1 and experiments section 4.16) to predict the behavioral actions between rat pairs of obtained video segments of experimental day 2 (interactions) for every 15 consecutive frames (one window) of the segments for each age group. We analyzed 520 seconds of each recorded video subset (starting at 5 seconds of the recording to 525 seconds of the recording). We split the analysis into two parts: a comparison of all the times spent in given actions between different rat pairings which are AD-AD, WT-WT, and AD-WT) and a comparison of time spent in given actions as an initiator of the subset of the actions with an initiator determined (actions [OI], [AP], and [MT]) between different rat types (AD or WT). Class [DT] was analyzed as a [NS] class based on experimental results (see experiments section 4.19).

To compare the different rat pairings for each classified interaction, we used the Shapiro-Wilk Test to test the normality of derived data. We use this test as our minimal sample size is four [48] (there are four subsets of AD-AD pairings, WT-WT pairings, and eight subsets for AD-WT pairings for a given age group in TgF344 AD Pair 18), followed by the non-parametric Kruskal-Wallis test (the choice of the non-parametric

method is based on results of the normality test - see Section 5.1). In our case, the Kruskal-Wallis Test determines if the populations of times spent in given actions differ across different rat type pairings.

To compare the time spent as an initiator for given actions between the AD and WT rat types in a given age group, we use the Shapiro-Wilk Test to test the normality (sample size is 16 for given rat type), followed by the Mann-Whitney rank sum test as a non-parametric method to compare the samples for each initiator action.

We repeat the denoted analysis for the predictions on dataset TgF344 AD Pair 18 by the Rule-Based Model (see Section 3.2.1) on chosen classes [HH] and [SB]. The [SB] class is analyzed with an initiator, as denoted above.

The time spent t_a in a given interaction is calculated as a sum of each individual prediction (one predicted sequence window equals 0.5 seconds). We differentiate t_a for each action, rat pairings (AD-AD, WT-WT, and AD-WT), and rat type and identity while differentiating the initiator of the action. The results of predictions made by the MMc_3 model are depicted in Section 5.1. The results of predictions made by the Rule-Based Model are depicted in Section 5.2. To perform the statistical tests, we employed MATLAB built-in methods.

3.5 Training protocol

We train the models with the Adam optimizer, with a learning rate $l_r = 0,001$. We employ a learning rate scheduler multiplying the learning rate with a factor of 0.1 if the validation loss plateaus. The objective function is the Cross-Entropy loss function. We implement a weighted sampling strategy to address the imbalance in class representation within dataset TgF344-AD Pair 18. This approach assigns a distinct weight to each sample, influencing its likelihood of being selected during training epochs. The weight for a class is assigned and normalized as $w_{class} = \frac{N}{count_{class}^p}$, where N is the total number of samples, and $count_{class}^p$ is the count of the given class in the dataset with a power adjustment p (a set hyperparameter). The normalization of the weights is defined as $w_{norm.class} = \frac{w_{class}}{\sum_{class=1}^C w_{class}}$, where C is the number of classes. Batch size is denoted for each method in the experiment section based on the computational resources.

Chapter 4

Experiments

4.1 Evaluation of DLC Pose Estimation

We evaluated the DLCRNet-MS5 (refer to Section 2.6) based on root mean squared error (RMSE) on the dataset consisting of 845 denoted frames with markers (refer to Section 2.8), where 0.85% of frames was used for training, 0.15% was used for testing. The tables Table 4.1 and Table 4.2 depict the pixel mean RMSE for each marker (also differentiated between the two rats) computed with a likelihood threshold for retaining the markers in a given frame (refer to Section 2.12). The tables display RMSE evaluation on the training and testing portion of the dataset. The dependency of likelihood-based marker cut-off and RMSE is depicted in Figure 4.1.

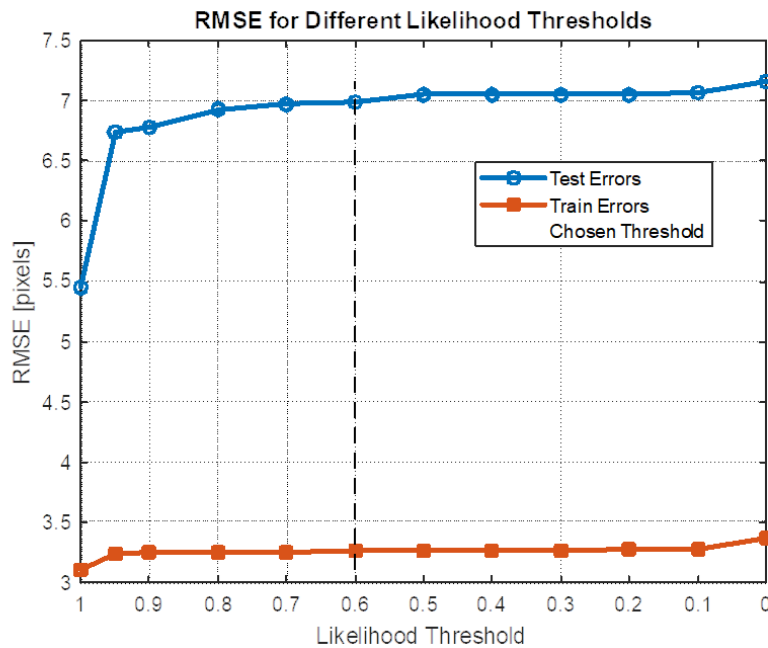


Figure 4.1. Pixel RMSE computed for different likelihood threshold cut-off for estimated markers by the trained DLCRNet-MS5 on training and testing dataset.

RMSE on Training Dataset			
Body Part	Mean RMSE	Rat X Mean RMSE	Rat Y Mean RMSE
M1: Snout	3.939	3.939	3.939
M2: Left Ear	3.241	3.275	3.207
M3: Right Ear	3.318	3.375	3.261
M4: Spine 1	3.278	3.228	3.329
M5: Spine 2	2.887	2.843	2.931
M6: Spine 3	3.184	3.156	3.211
M7: Spine 4	3.267	3.252	3.282
M8: Spine 5	3.549	3.483	3.616
M9: Tail Base	3.703	3.622	3.783

Table 4.1. Pixel RMSE computed for estimations by the trained DLCRNet-MS5 on the training dataset (refer to 2.13.3) for each denoted marker.

RMSE on Testing Dataset			
Body Part	Mean RMSE	Rat X Mean RMSE	Rat Y Mean RMSE
M1: Snout	6.477	7.350	5.605
M2: Left Ear	4.670	4.440	4.899
M3: Right Ear	4.873	4.974	4.772
M4: Spine 1	5.803	5.836	5.770
M5: Spine 2	7.766	7.620	7.912
M6: Spine 3	8.676	8.083	9.268
M7: Spine 4	9.528	8.841	10.215
M8: Spine 5	8.682	8.581	8.782
M9: Tail Base	7.947	8.104	7.790

Table 4.2. Pixel RMSE computed for estimations by the trained DLCRNet-MS5 on the testing dataset (refer to 2.13.3) for each denoted marker.

4.2 Point Distribution Model (PDM)

The PDM performance was evaluated by the Mean Squared Error (MSE) between a set of 800 valid shapes and their respective copies, from which a predetermined number of markers were removed to be estimated by the PDM.

In each iteration, the indices of the markers to be removed were randomly selected, to ensure a robust assessment of the PDM’s performance across various instances (the random selection was employed in MATLAB by a built-in ‘randperm’ function, using the uniform pseudorandom number generator). The MSE compared to a number of markers randomly removed from the original shape and estimated by the PDM is depicted in Figure 4.4, with a comparison of MSE while using only the mean shape to fill in the missing markers. The visualization of the estimated markers (of three, four, five, and six markers) is depicted in Figures 4.2 and 4.3.

Based on a subjective visual evaluation of the PDM, the PDM performs rather well when it estimates the position of markers between peripheral markers, but the accuracy diminishes if the PDM tries to estimate the rotation of peripheral markers. We depict the visualization of PDM in Figures 4.2 and 4.3. On average, given our dataset and the estimations from DLC, the PDM estimates about 3.7 % markers per instance for a single rat. Typically, one to three markers are being imputed if given the situation of missing or removed markers (refer to Section 2.12).

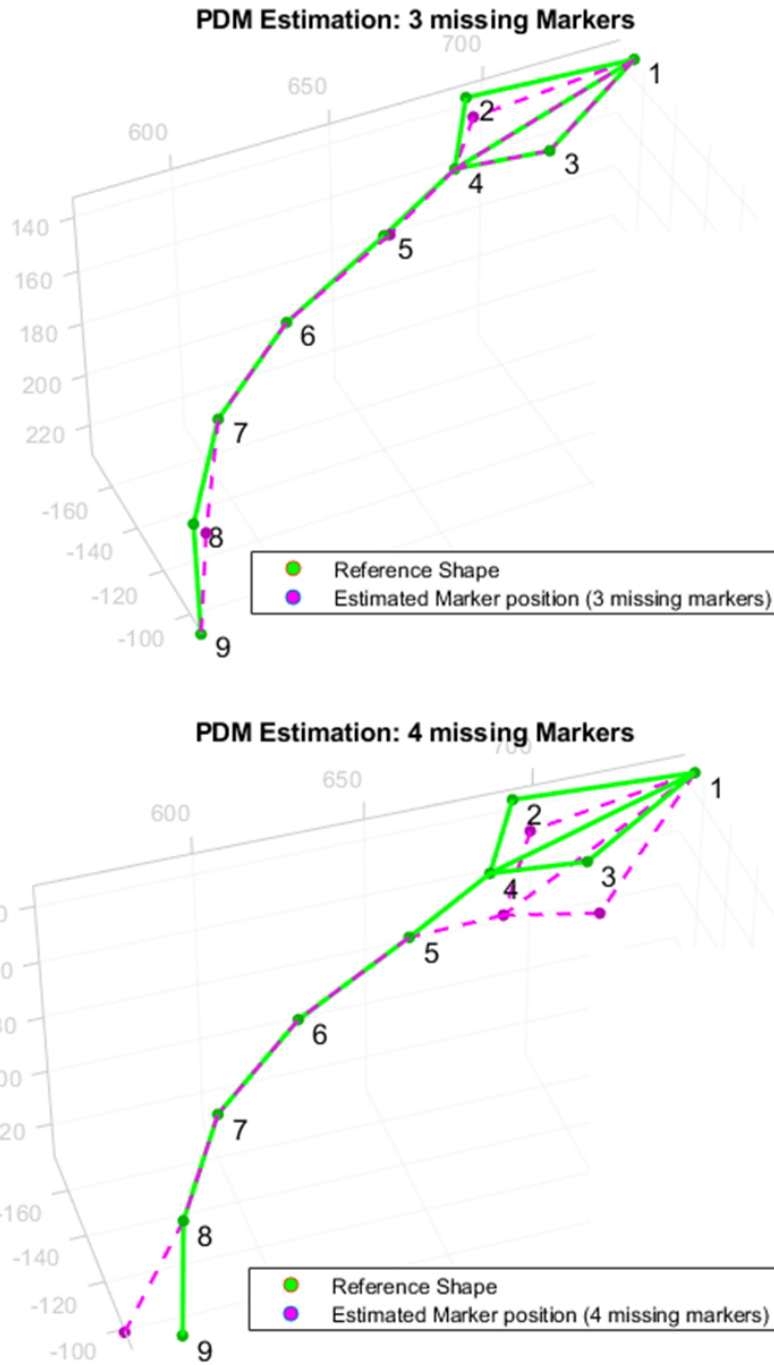


Figure 4.2. PDM estimation of three missing markers (indexes 2, 5, 8) and four missing markers (indexes 2, 3, 4, 9). The estimated markers: purple, reference shape: green. Marker 1 - snout, 2 - left ear, 3 - right ear, 4 - spine 1, 5 - spine 2, 6 - spine 3, 7 - spine 4, 8 - spine 5, 9 - tail-base.

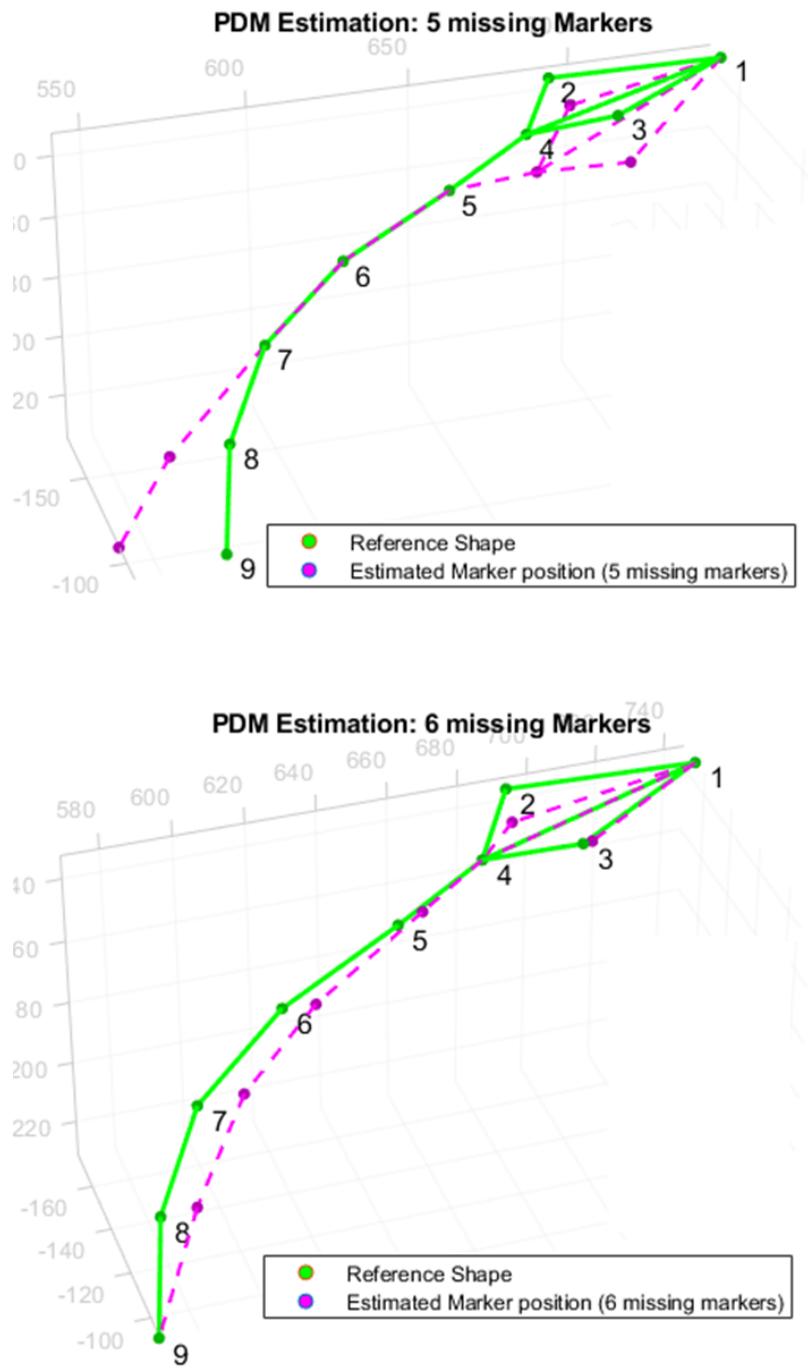


Figure 4.3. PDM estimation of five missing markers (indexes 2, 3, 4, 8, 9) and six missing markers (indexes 2, 3, 5, 6, 7, 8). The estimated markers: purple, reference shape: green. Marker 1 - snout, 2 - left ear, 3 - right ear, 4 - spine 1, 5 - spine 2, 6 - spine 3, 7 - spine 4, 8 - spine 5, 9 - tail-base.

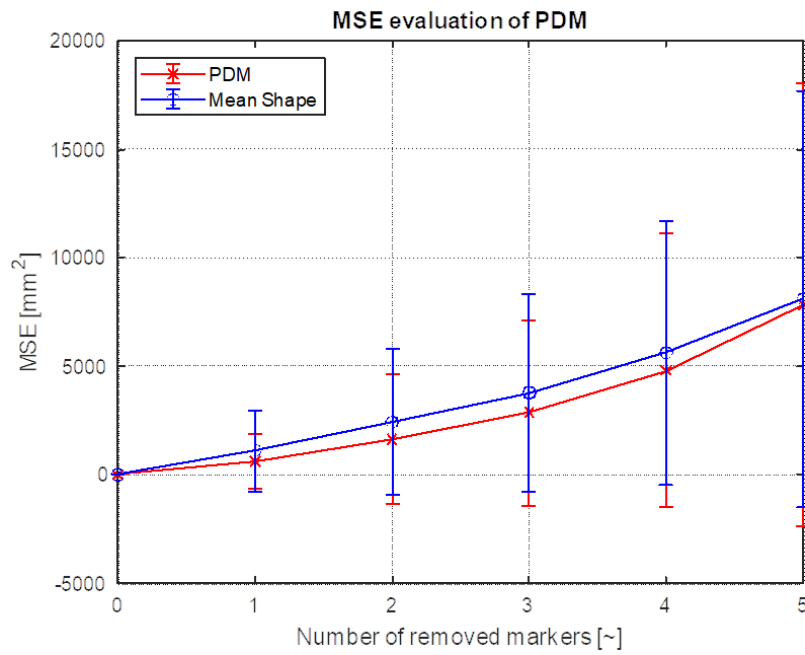


Figure 4.4. MSE with error bars between original shapes and same shapes with a determined number of markers randomly removed and estimated by the PDM. For visual purposes, we display up to five removed markers – the MSE of estimating six markers is around 12000.

4.3 Comparison of PDM Imputation with PDM and Geometric Rule-Based Imputation

To see if the PDM imputation performs with a lesser MSE in conjunction with the geometric rule-based imputation, we conducted an experiment that mirrors the methodology of the experiment described in section 4.1. We compared the imputation solely by PDM and the imputation by PDM, that if the shape to be corrected had the markers suitable for geometric imputation (that meaning only one of the markers identified as snout – 1, left ear – 2, or right ear – 3 and present marker spine1 – 4), used the geometric imputation beforehand.

The experiment results show that the geometric imputation should be included before using the PDM. The comparison is visible in the graph Figure 4.5 and Figure 4.6, and visualization in Figure 4.7.

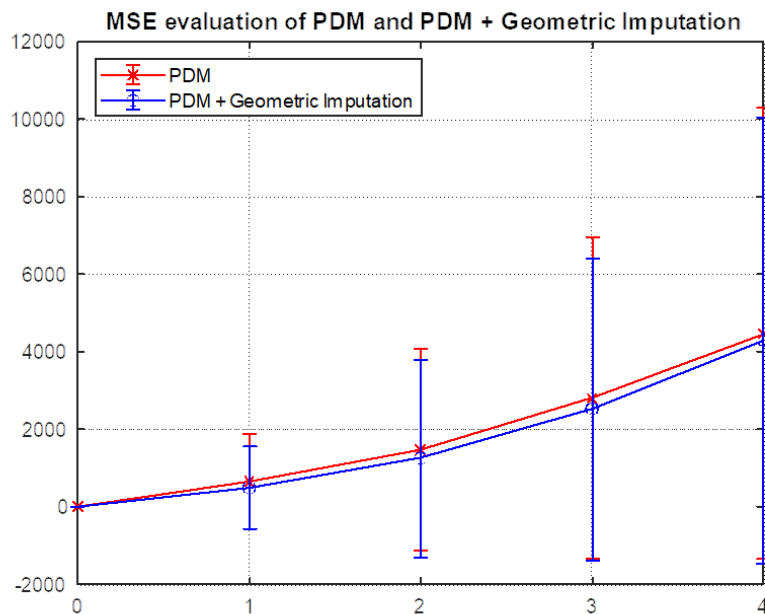


Figure 4.5. Comparison with error bars of using solely PDM (red) and PDM (blue) with a preceding geometric imputation (as defined in 2.10.2). The results show that the MSE is lesser with a preceding geometric imputation to the PDM.

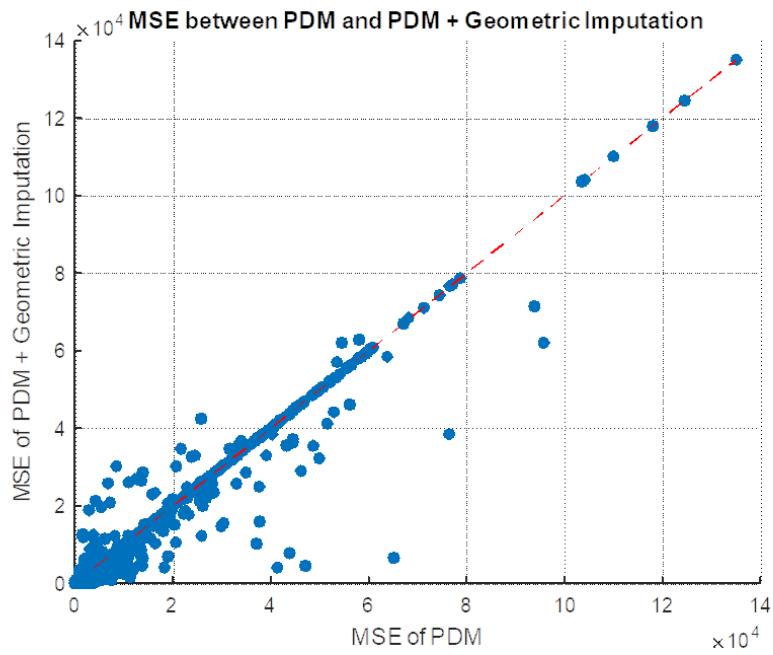


Figure 4.6. Scatter plot between MSE of solely PDM estimation and PDM with preceding geometric imputation. Points below the red line indicate instances where preceding geometric imputations enhanced PDM performance.

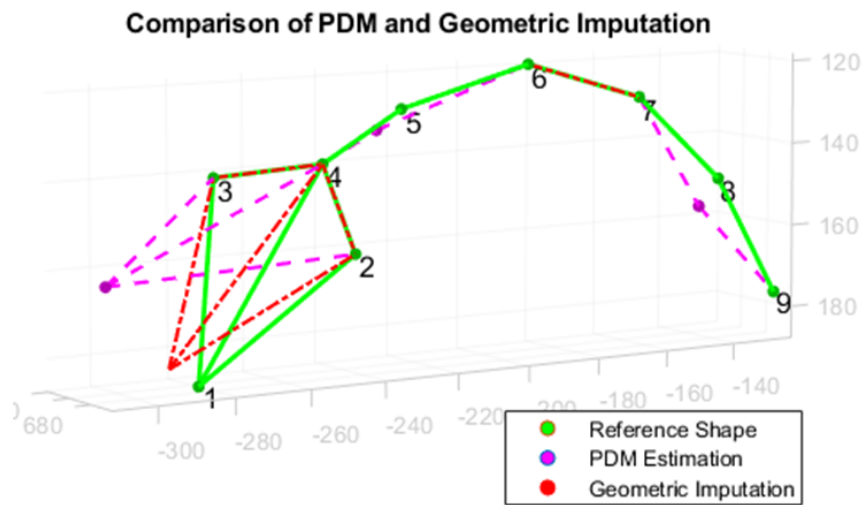


Figure 4.7. The visual comparison of PDM – estimating markers indexed 1,5,8 and geometric imputation of the snout – 1.

4.4 Temporal Imputation method

We decided on the most suitable temporal imputation method for the 3D coordinates based on MSE measured while interpolating randomly removed sequences (of the chosen maximal gap size of 15 instances). The sequence was randomly removed throughout 1000 iterations (tested throughout four subsets of 2-minute length). The MSE was measured for both rats across the whole imputed sequence between the original sequences and randomly chosen imputed ones (for the randomized algorithm, refer to Section 4.1). The results of MSE are depicted in Figure 4.8. The best method for temporal imputation on the rat sequences was the piecewise cubic Hermite interpolating polynomial (PCHIP).

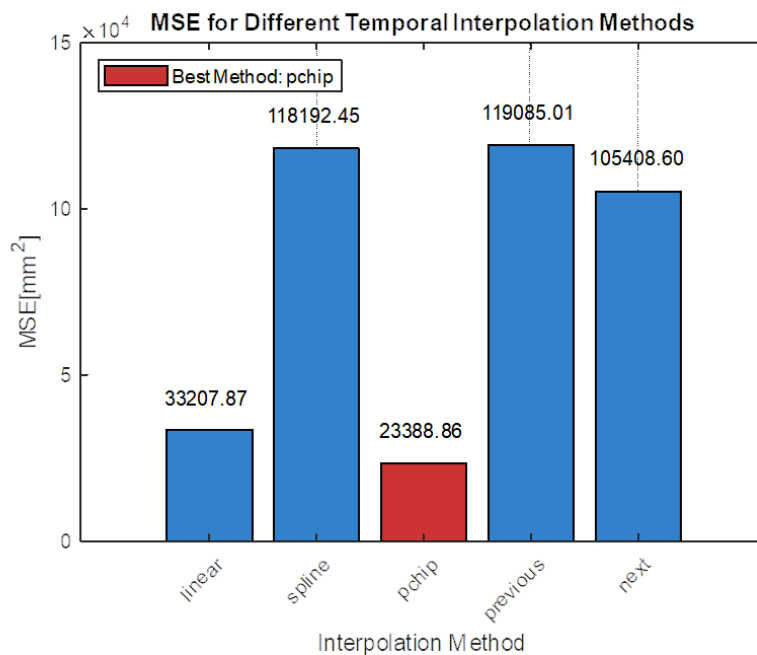


Figure 4.8. Choice of method for temporal imputation of 3D coordinates based on MSE. The red bar represents the chosen method as a piecewise cubic Hermite interpolating polynomial – refer to Section 2.11.

4.5 Identity Swap correction

We evaluated the implementation of the identity swap correction (refer to Section 3.1) by intentionally swapping the identities of rats in the sequences and checking the number of corrected frames. We randomly selected thirty subsequences of a random length (ranging from zero to fifteen) for a hundred iterations. We then randomly selected one, two, three, and four views and intentionally swapped the rat’s identities in those selections (for the randomized algorithm, refer to Section 4.1). After correcting the sequences, we obtained the result depicted in Figure 4.9. The instance was considered successfully corrected if the markers with their assigned identities matched the markers before the intentional identity swap across all the views.

The experiment results show a slow decrease in the success rate of the correction with a rising number of views with swapped identities, ranging from 96,5% to 89%. The total number of instances with intentionally swapped identities was 12466 in four video subsets.

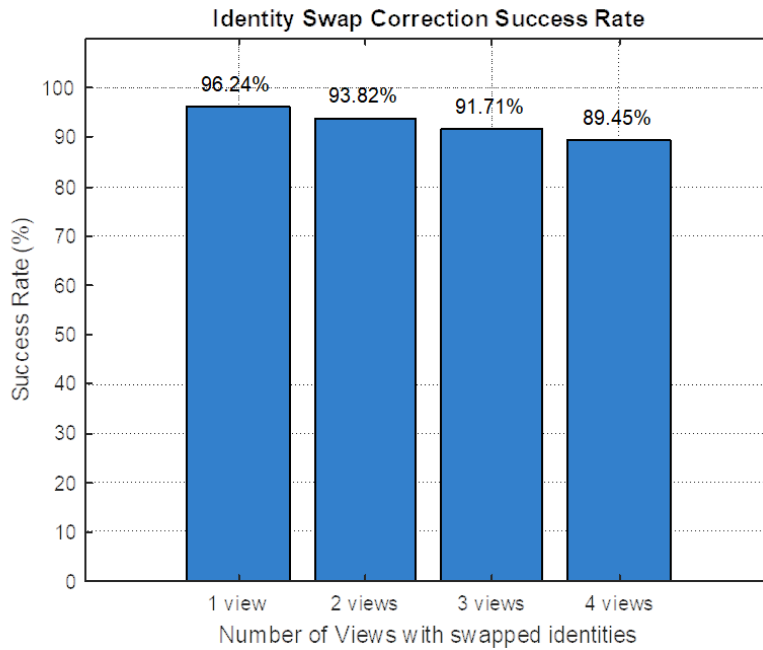


Figure 4.9. The Identity Swap correction success rate of the method described in Section 3.1. The blue bars represent the percentage of successfully corrected instances of intentionally swapped identities across all the views.

4.6 Rule-Based Model Evaluation

The Rule-Based Model was evaluated on 596 sequence windows (one sequence window consisting of fifteen frames) from dataset Rule-TgF344AD Pair 18, denoted in 2.13.1. Table 4.3, confusion matrix in Figure 4.11, and Figure 4.10 depict the evaluation of the Rule-Based Model with thresholds T^* and window sizes W^* found by the Bayesian optimization (refer to Section 3.2.1).

Rule-Based Model Class Performance Metrics			
Class	F_1 Score	Precision	Recall
[NS]	0.885	0.895	0.904
[HH]	0.800	0.828	0.857
[SB]	0.836	0.827	0.818
[AP]	0.469	0.566	0.714
[PS]	0.829	0.687	0.586

Table 4.3. Table showing the F_1 Score, precision, and recall for each class predicted by the Rule-Based Model on dataset Rule-TgF344AD Pair 18.

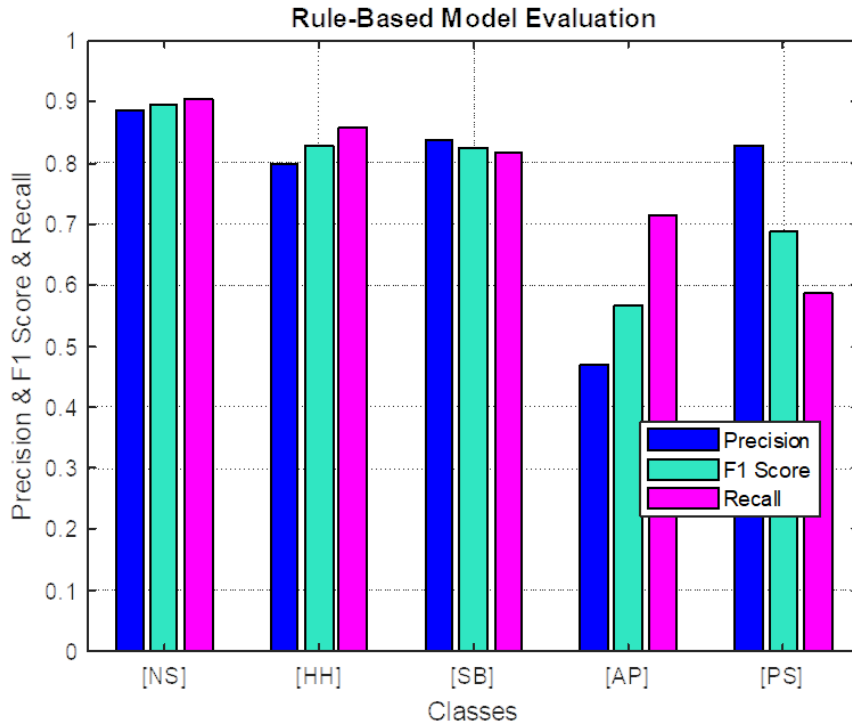


Figure 4.10. F_1 Score (cyan), precision (blue), and recall (purple) evaluated for each of the classes predicted by the Rule-Based Model (refer to Section 3.2.1) on dataset Rule-TgF344AD Pair 18.

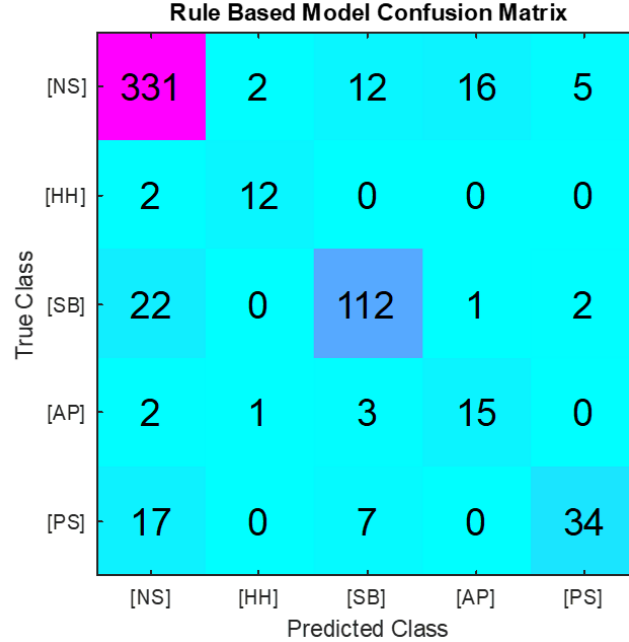


Figure 4.11. Confusion Matrix of the classes predicted by the Rule-Based Model (refer to Section 3.2.1) on dataset Rule-TgF344AD Pair 18.

The model was evaluated using the F_1 score, precision, and recall. The F_1 score was calculated for each sequence window (and not instances) to follow up the labeling process. The label for the sequence window was determined by a majority vote from the 15 consecutive predictions (the sequence window length). Based on this evaluation, the Total F_1 Score of the Rule-Based Model is 0,769 on dataset Rule-TgF344AD Pair

Rule-Based Model Thresholds			
Variable	Threshold Denotation	Value	Units
Maximal passive interaction distance	t_{mp}	164	mm
Maximal Standard deviation	t_{std}	1068	mm
Minimal head contact distance	t_h	79	mm
Minimal snout and tail-base distance	t_{sb}	52	mm
Minimal approach starting distance	t_{ms}	171	mm
Minimal approach traveled distance	t_{ma}	90	mm
Maximal approach speed	t_v	92	mm s^{-1}
Maximal passive speed	t_{vp}	108	mm s^{-1}
Approach window size	w_{ap}	28	
Passive interaction window size	w_{ps}	76	

Table 4.4. Table showing thresholds found by the Bayesian Optimization for the Rule-Based Model on dataset Rule-TgF344AD Pair 18 denoted in Section 3.2.1.

18. The found threshold values for set T^* and window sizes set W^* (for details denoted in Section 3.2.1) are depicted in Table 4.4.

4.7 Rule-Based Model – TgF344AD Pair 18 adaptation Evaluation

Adapting the rule-based model to comprehend the classes denoted for deep learning models in dataset TgF344AD Pair 18 had a lesser expected performance. Classification of classes [NS] or [AP] and others without detecting contact classes of mutual exploration or olfactory investigation had an expected decrease in the evaluation metrics. As for the class [MM], we managed to detect it throughout the dataset, but after the majority vote on the clip sequence windows, its detection stayed at zero samples. The evaluation protocol followed the evaluation of the rule-based model denoted in Section 3.2.1. The overall F_1 score of the model is 0,2794 on dataset TgF344AD Pair 18. Table 4.5 and Figure 4.12 depict the evaluation of the Rule-Based Model adaptation. The Bayesian optimization found thresholds are denoted in Table 4.6.

Rule-Based Model Ad. DClass Performance Metrics			
Class	F_1 Score	Precision	Recall
[NS]	0.938	0.840	0.761
[EX]	0.107	0.168	0.400
[OI]	0.252	0.374	0.725
[AP]	0.232	0.285	0.370
[DT]	0.250	0.302	0.382
[MT]	0.214	0.110	0.074
[PS]	0.179	0.155	0.136
[MM]	0.000	0.000	0.000

Table 4.5. Table showing the F_1 Score, precision, and recall for each class predicted by the Rule-Based Model Adaptation on dataset TgF344AD Pair 18.

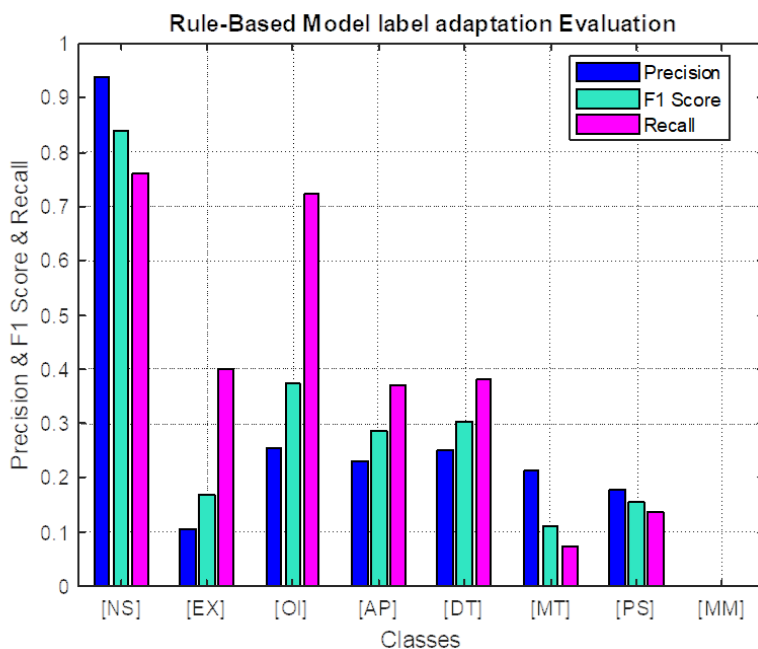


Figure 4.12. F_1 Score (cyan), precision (blue), and recall (purple) evaluated for each of the classes predicted by the Rule-Based Model Adaptation (refer to Section 3.2.2) on dataset TgF344AD Pair 18.

Rule-Based Model Adaptation Thresholds				
Variable	Threshold Denotation	Value	Units	
Maximal passive interaction distance	t_{mp}	84	mm	
Maximal Standard deviation	t_{std}	1075	mm	
Minimal mutual exploration distance	t_{ex}	61	mm	
Minimal olfactory exploration distance	t_{oi}	63	mm	
Minimal approach starting distance	t_{ms}	156	mm	
Minimal approach traveled distance	t_{ma}	85	mm	
Maximal approach speed	t_v	187	mm s^{-1}	
Maximal passive speed	t_{vp}	72	mm s^{-1}	
Approach window size	w_{ap}	34		
Passive interaction window size	w_{ps}	58		
Minimal detach starting distance	t_{md}	175	mm	
Minimal detach traveled distance	t_{ds}	210	mm	
Minimal detach speed	t_{vd}	81	mm s^{-1}	
Detach window size	w_{dt}	50		
Minimal mounting threshold in z-axis	t_z	592	mm	
Minimal mounting distance threshold	t_{mt}	59	mm	
Minimal rear mimicking threshold in z-axis	t_{mm}	1062	mm	

Table 4.6. Table showing thresholds found by the Bayesian Optimization for the Rule-Based Model Adaptation on dataset TgF344AD Pair 18 denoted in Section 3.2.2.

4.8 Experiments: Deep Models Evaluation

We trained and evaluated the models for each of the proposed methods for action recognition in the methods section. Each model was trained on our dataset TgF344AD Pair 18, and for the 3D coordinate modality ST GCN-based models, we also include the report on dataset Pair-R24M. For training purposes, we use 80% of the datasets and validate on 20%. The random split for the training and testing dataset is set with a permanent seed to ensure consistency across different methods. The split is performed using stratified sampling to ensure a uniform representation of each class

across both the training and validation datasets. In the pages below, we report the training and validation based on the F_1 score for each class, precision, and recall, the overall accuracy, loss, and confusion matrices if the model or the proposed method was denoted as sufficient on dataset TgF344AD Pair 18. We report the F_1 scores, precision, and recall otherwise for simplicity. The reported model performance values are the ones with minimal validation loss throughout the training process of a given model. For a comparison of all denoted methods, refer to experiment section 4.17.

4.9 ST GCN Model Training and Evaluation

We train and evaluate the ST GCN (refer to Section 3.3.1) on datasets TgF344AD Pair 18 and Pair-R24M. In Table 4.7 and Figure 4.13, we display the TgF344AD Pair 18 results. We report F_1 score values for each class (as denoted in datasets - refer to Section 2.13), overall accuracy and loss for topology configurations passed to the ST GCN c_1 and c_2 and architectural augmentation ST GCN-s using the topology configuration c_2 . Based on the evaluation, the ST GCN method proved insufficient across both presented topology configurations (with overall validation accuracy of 0.345 on configuration c_2) and architectural configuration on dataset TgF344AD Pair 18. The presented 3D coordinate sequence window was not sufficient to learn the relations for our denoted actions, and based on a comparison of results on the Pair R24-M dataset, where the ST GCN was able to learn to classify the actions, our denoted dataset TgF344AD Pair 18 did not comprehend enough labels for the ST GCN model. The model was trained on batch sizes of 48 and 100 epochs. Results on the Pair R24-M are mentioned in the following paragraph.

ST GCN Class Performance Metrics on TgF344AD Pair 18						
Class	[T] ST c_1	[V] ST c_1	[T] ST c_2	[V] ST c_2	[T] ST GCN-s	[V] ST GCN-s
[NS]	0.420	0.548	0.422	0.500	0.299	0.260
[EX]	0.014	0.000	0.003	0.000	0.000	0.000
[OI]	0.276	0.299	0.297	0.350	0.000	0.000
[AP]	0.000	0.000	0.005	0.000	0.000	0.000
[DS]	0.007	0.000	0.025	0.000	0.000	0.000
[MT]	0.290	0.527	0.300	0.569	0.000	0.000
[PS]	0.233	0.322	0.280	0.482	0.000	0.000
[MM]	0.048	0.000	0.258	0.198	0.000	0.000

Table 4.7. Table showing the class performance F_1 Score metrics for training ([T]) and validation ([V]) on dataset TgF344AD Pair 18 across ST GCN (ST) topology configurations c_1 , c_2 and architectural augmentation ST GCN-s.

In Figure 4.14 and Table 4.8, we present the performance of the ST GCN on the Pair R24-M dataset. The configurations P24 c_1 and P24 c_2 utilize the fully connected graph topology for the spatial adjacency matrix A_s used in the spatial GCN layers. Configuration P24 c_1 represents the parallel architecture of the network, as denoted in Section 3.3.1, and configuration P24 c_2 represents the GCN layers arranged in series ST GCN-s, also detailed in Section 3.3.1. We performed this experiment to see if the implementation of ST GCN failed on TgF344 AD Pair 18 due to the model’s architecture or the dataset size. Based on the evaluation on dataset Pair-R24M, with a sufficient number of training sequence windows (refer to Section 2.13.2 to see the Pair-R24M dataset size), the ST GCN implementation was able to learn the spatial and temporal dependencies of the rats’ actions effectively. The configuration P24 c_1 demonstrated superior performance across all classes of the Pair R24-M dataset.

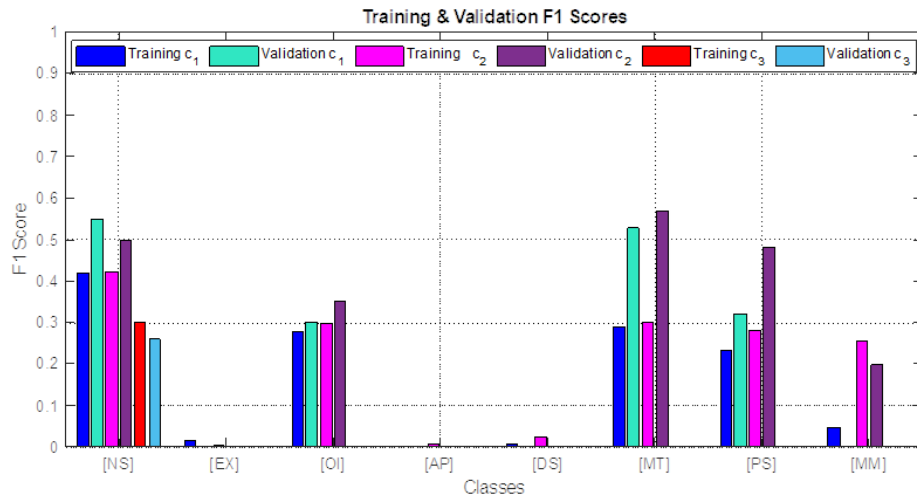


Figure 4.13. F_1 score class performance for training and validation on dataset TgF344AD Pair 18 across ST GCN topology configurations c_1 , c_2 and architectural augmentation ST GCN-s (c_3).

ST GCN Class Performance Metrics on Pair 24-M Dataset				
Class	[T] ST P24 c_1	[V] ST P24 c_1	[T] ST P24 c_2	[V] ST P24 c_2
[NS]	0.927	0.912	0.775	0.767
[EX]	0.747	0.750	0.647	0.636
[CS]	0.724	0.698	0.560	0.560
[PS]	0.736	0.624	0.418	0.396

Table 4.8. Table showing the class performance F_1 score metrics for training ([T]) and validation ([V]) on dataset Pair R24-M across ST GCN (ST) configurations P24 c_1 , P24 c_2 .

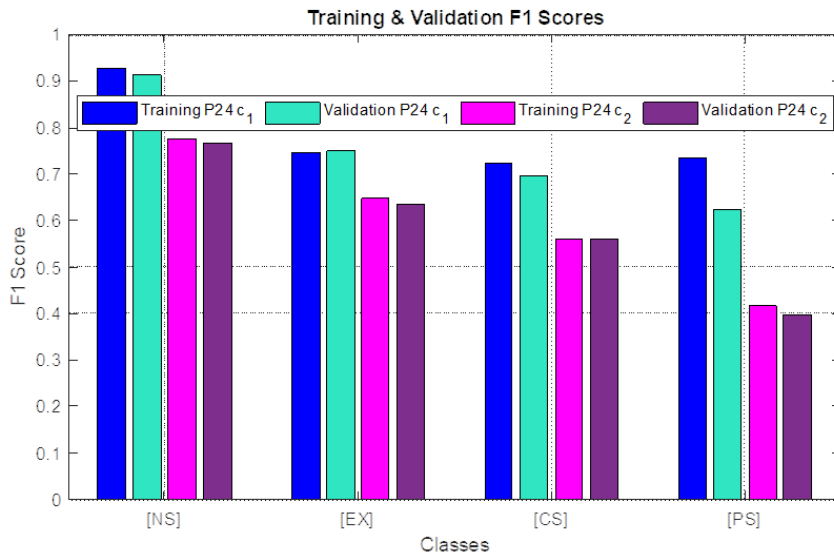


Figure 4.14. F_1 score class performance for training and validation on dataset Pair R24-M across ST GCN configurations P24 c_1 , P24 c_2 .

4.10 TSRJI-CNN Model Training and Evaluation

We train and evaluate the TSRJI-CNN Model (refer to Section 3.3.3) on dataset TgF344AD Pair 18. In Table 4.9, and Figures 4.15 and 4.16, we report F_1 score,

precision, and recall values for each class (as denoted in dataset TgF344AD Pair 18), overall accuracy, loss, and the validation confusion matrix. The model was trained with augmentations denoted in section 3.3.2. In our evaluation, the TSRJI-CNN method demonstrated adequate performance across most classes, achieving an overall validation accuracy of 0.696. However, it exhibited limitations in accurately recognizing the [MT] (mounting) class, which is generally more discernible by human eye observation. The representation of relative rat positions in polar coordinates with reference markers was found to be effective. The model was trained on batch size of 48 and 100 epochs.

TSRJI-CNN Class Performance Metrics on TgF344AD Pair 18						
Class	[T] F_1 Sc.	[V] F_1 Sc.	[T] recall	[V] recall	[T] precision	[V] precision
[NS]	0.784	0.699	0.726	0.825	0.873	0.644
[EX]	0.875	0.791	0.938	0.785	0.838	0.822
[OI]	0.733	0.557	0.705	0.651	0.787	0.510
[AP]	0.855	0.764	0.865	0.770	0.862	0.777
[DS]	0.891	0.667	0.950	0.537	0.866	0.944
[MT]	0.941	0.358	0.949	0.279	0.947	0.538
[PS]	0.888	0.826	0.944	0.897	0.852	0.793
[MM]	0.947	0.754	0.988	0.740	0.920	0.820

Table 4.9. Table showing the F_1 Score (F_1 Sc.), precision, and recall evaluated for the training ([T]) and validation ([V]) of the TSRJI-CNN Model (refer to Section 3.3.3) on dataset TgF344AD Pair 18 on each class.

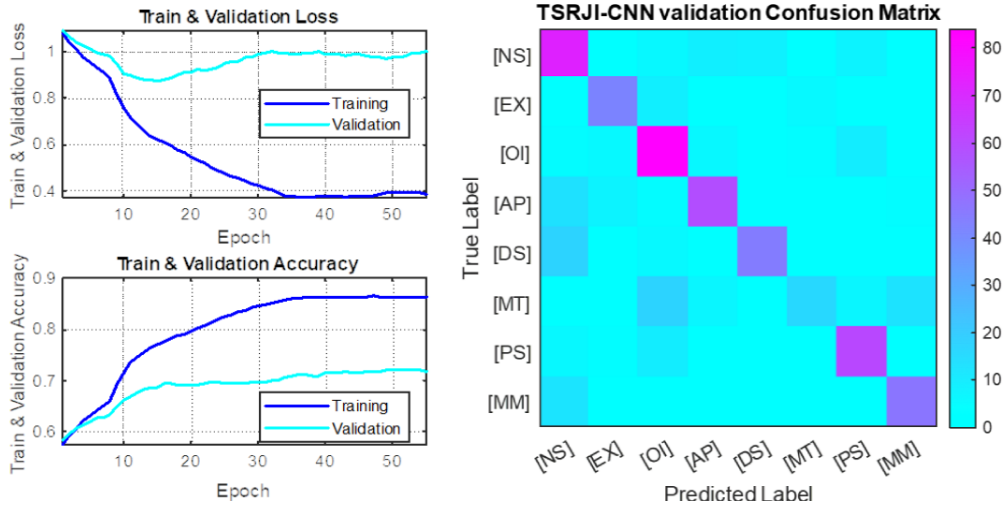


Figure 4.15. Training and validation loss, accuracy, and the validation Confusion Matrix for TSRJI-CNN Model on dataset TgF344AD Pair 18.

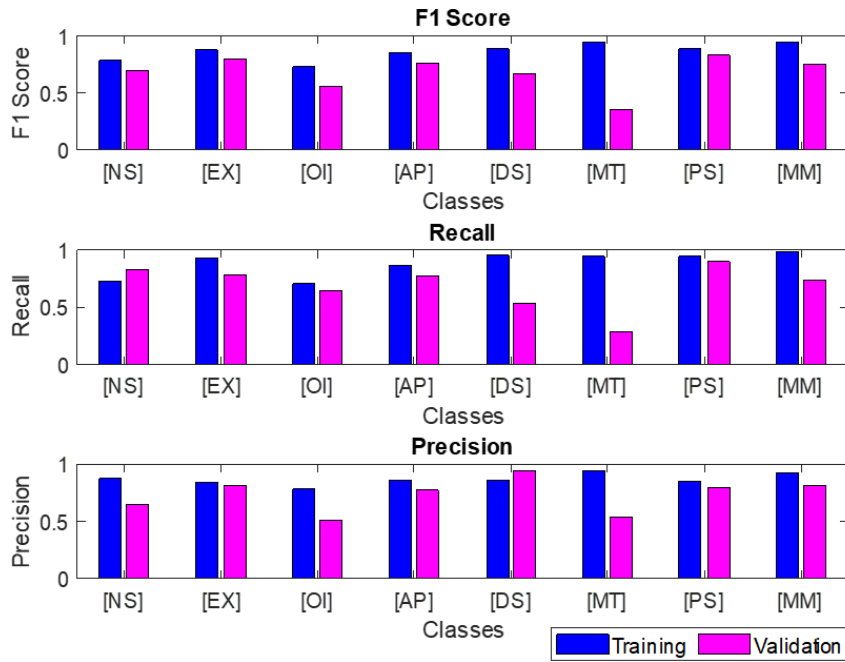


Figure 4.16. Table showing the F_1 Score (F_1 Sc.), precision, and recall evaluated for the training ([T]) and validation ([V]) of the TSRJI-CNN Model (refer to Section 3.3.3) on dataset TgF344AD Pair 18 on each class.

4.11 Pivot TSRJI-CNN Model Training and Evaluation

We train and evaluate the Pivot TSRJI-CNN Model (refer to 3.3.4) on dataset TgF344AD Pair 18. In Table 4.10, and Figure 4.17, we report the F_1 score, precision, and recall values for each class (as denoted in dataset TgF344AD Pair 18). The model was trained with augmentations denoted in section 3.3.2. The Pivot TSRJI-CNN method demonstrated superior performance compared to certain proposed models, such as the ST GCN and 3D CNN, when tested on the dataset TgF344AD Pair 18. However, it did not surpass the performance of the TSRJI-CNN method utilizing multiple reference joints, achieving an overall validation accuracy of 0.492. The model was trained on batch size of 48 and 100 epochs.

Pivot TSRJI-CNN Class Performance Metrics on TgF344AD Pair 18						
Class	[T] F_1 Sc.	[V] F_1 Sc.	[T] recall	[V] recall	[T] precision	[V] precision
[NS]	0.454	0.447	0.477	0.510	0.470	0.410
[EX]	0.481	0.565	0.594	0.701	0.438	0.523
[OI]	0.287	0.123	0.262	0.104	0.342	0.171
[AP]	0.409	0.366	0.400	0.398	0.511	0.376
[DS]	0.522	0.445	0.503	0.391	0.597	0.642
[MT]	0.582	0.540	0.625	0.617	0.586	0.531
[PS]	0.557	0.641	0.634	0.617	0.543	0.697
[MM]	0.746	0.638	0.804	0.747	0.731	0.576

Table 4.10. Table showing the F_1 Score (F_1 Sc.), precision, and recall evaluated for the training ([T]) and validation ([V]) of the Pivot TSRJI-CNN Model (refer to Section 3.3.4) on dataset TgF344AD Pair 18 on each class.

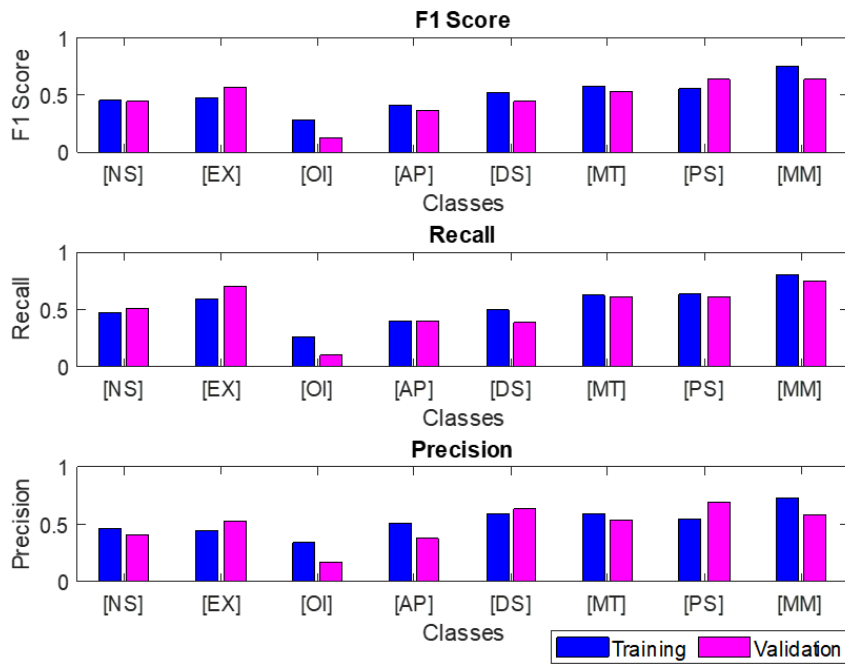


Figure 4.17. F_1 Score, precision, and recall evaluated for the training and validation of the Pivot TSRJI-CNN Model (refer to Section 3.3.4) on dataset TgF344AD Pair 18 on each class.

4.12 CNN-LSTM Model Training and Validation

We train and evaluate the CNN-LSTM Model (refer to Section 3.3.5) on dataset TgF344AD Pair 18. In Table 4.11, and Figures 4.18 and 4.19, we report F_1 score, precision, and recall values for each class (as denoted in dataset TgF344AD Pair 18), overall accuracy, loss, and the validation confusion matrix. The model was trained with augmentations denoted in 3.3.6. Based on the evaluation, the CNN-LSTM method proved sufficient for most of the classes with an overall validation accuracy of 0.669. However, the model exhibited limitations in accurately recognizing the classes [AP] (Approach) and [DT] (Detach). The [AP] class, characterized by movements and head orientations towards another rat, presents a challenge due to its potential similarity with mutual and olfactory exploration behaviors. The [DT] class was predominantly misclassified as the [NS] (No Social) class. This observation suggests the feasibility of considering a potential merge of these two classes to improve classification robustness after a discussion with a biologist. Compared with the TSRJI-CNN method, the CNN-LSTM model showed improved recognition in the [MT] (Mounting) class, aligning with its distinguishable characteristics observable by the human eye. To summarize, the sequence of images throughout all views proved to be a sufficient feature for the denoted actions. The model was trained on a batch size 16 (the batch size needed to be lowered due to memory limitations in our hardware setup - refer to Section 2.1) and 60 epochs.

CNN-LSTM Class Performance Metrics on TgF344AD Pair 18						
Class	[T] F_1 Sc.	[V] F_1 Sc.	[T] recall	[V] recall	[T] precision	[V] precision
[NS]	0.454	0.447	0.477	0.510	0.470	0.410
[EX]	0.481	0.565	0.594	0.701	0.438	0.523
[OI]	0.287	0.123	0.262	0.104	0.342	0.171
[AP]	0.409	0.366	0.400	0.398	0.511	0.376
[DS]	0.522	0.445	0.503	0.391	0.597	0.642
[MT]	0.582	0.540	0.625	0.617	0.586	0.531
[PS]	0.557	0.641	0.634	0.617	0.543	0.697
[MM]	0.746	0.638	0.804	0.747	0.731	0.576

Table 4.11. Table showing the F_1 Score (F_1 Sc.), precision, and recall evaluated for the training ([T]) and validation ([V]) of the CNN-LSTM Model (refer to Section 3.3.5) on dataset TgF344AD Pair 18 on each class.

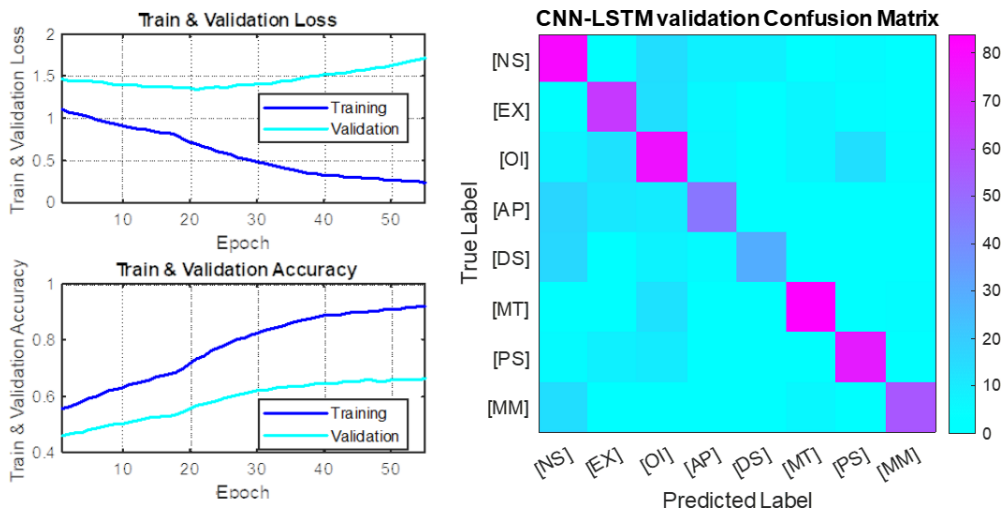


Figure 4.18. Training and validation loss, accuracy, and the validation Confusion Matrix for TSRJI-CNN Model on dataset TgF344AD Pair 18.

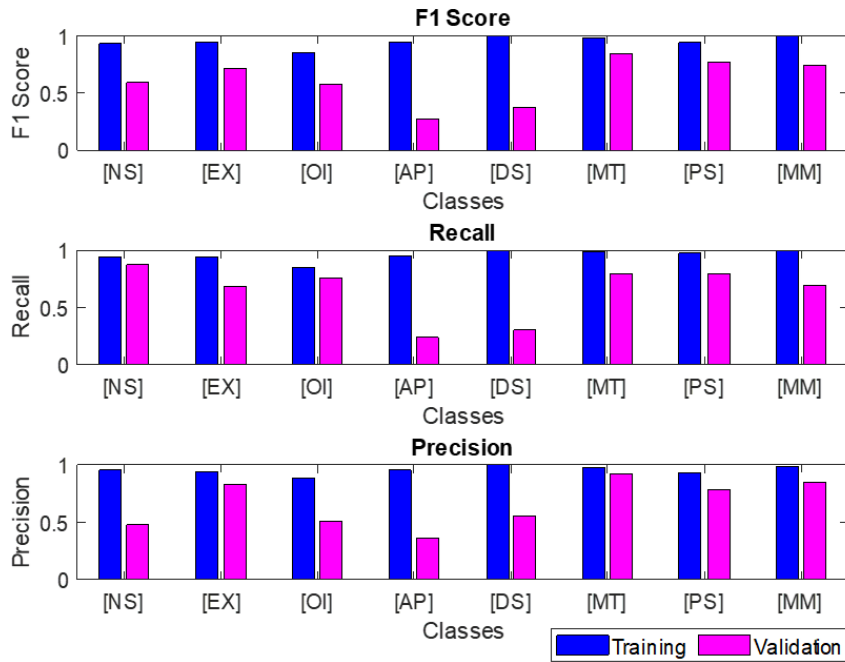


Figure 4.19. F_1 Score, precision, and recall evaluated for the training and validation of the CNN-LSTM Model (refer to Section 3.3.5) on dataset TgF344AD Pair 18 on each class.

4.13 3D CNN Model Training and Validation

We train and evaluate the 3D CNN Model (refer to Section 3.2.7) on dataset TgF344AD Pair 18. In Table 4.12, and Figure 4.20, we report F_1 score, precision, and recall values for each class (as denoted in dataset TgF344AD Pair 18). The model was trained with augmentations denoted in 3.3.7. Based on the evaluation, the 3D CNN method and the Motion Cuboid built from the view image sequences proved highly insufficient, with an overall validation accuracy of 0.296. The model was trained on batch size of 16 (the batch size needed to be lowered due to memory limitations in our hardware setup) and 60 epochs.

3D CNN Model Class Performance Metrics on TgF344AD Pair 18						
Class	[T] F_1 Sc.	[V] F_1 Sc.	[T] recall	[V] recall	[T] precision	[V] precision
[NS]	0.181	0.118	0.193	0.126	0.212	0.147
[EX]	0.223	0.102	0.250	0.105	0.231	0.115
[OI]	0.094	0.120	0.098	0.172	0.109	0.109
[AP]	0.270	0.244	0.295	0.283	0.285	0.243
[DS]	0.529	0.216	0.578	0.202	0.524	0.248
[MT]	0.395	0.262	0.445	0.292	0.395	0.279
[PS]	0.498	0.467	0.537	0.483	0.514	0.500
[MM]	0.495	0.232	0.532	0.225	0.496	0.260

Table 4.12. Table showing the F_1 Score (F_1 Sc.), precision, and recall evaluated for the training ([T]) and validation ([V]) of the 3D CNN Model (refer to Section 3.3.7) on dataset TgF344AD Pair 18 on each class.

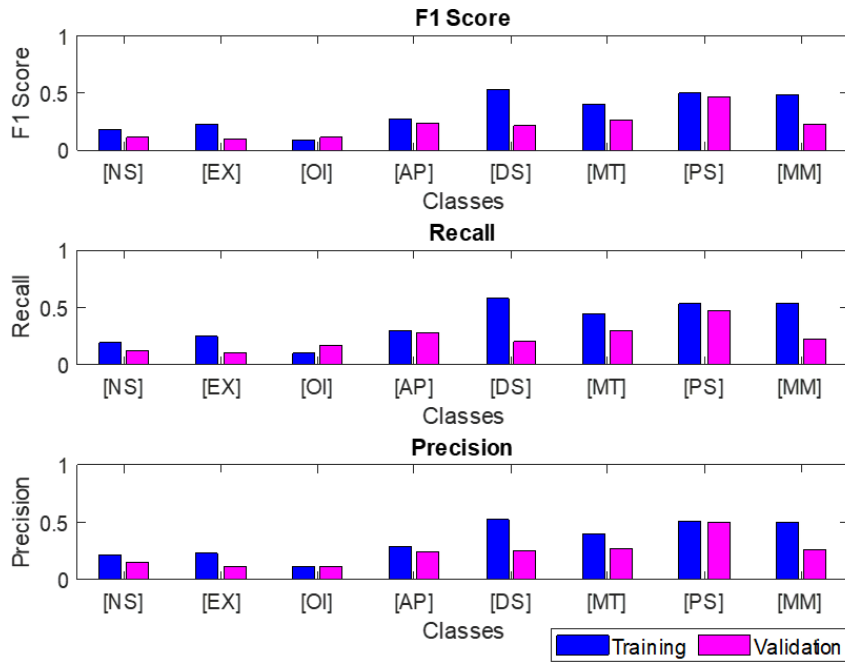


Figure 4.20. F_1 Score, precision, and recall evaluated for the training and validation of the 3D CNN Model (refer to Section 3.3.7) on dataset TgF344AD Pair 18 on each class.

4.14 1D CNN Model Training and Validation

We train and evaluate the 1D CNN Model (refer to Section 3.2.8) on dataset TgF344AD Pair 18. In Table 4.13 and Figure 4.21, we report the F_1 score, precision, and recall values for each class (as denoted in dataset TgF344AD Pair 18). Based on the evaluation, the 1D CNN method and the chosen features sequences $F = [D_H \ D_{SB}^X \ D_{SB}^Y \ V_R^X \ V_R^Y]$ (as denoted in Section 3.2.1) did outperform some of the models (Pivot TSRJI, or ST GCN), but did not outperform the TSRJI-CNN or CNN-LSTM Models with overall validation accuracy of 0.498. The model was trained on batch size of 48 and 100 epochs.

1D CNN Model Class Performance Metrics on TgF344AD Pair 18						
Class	[T] F_1 Sc.	[V] F_1 Sc.	[T] recall	[V] recall	[T] precision	[V] precision
[NS]	0.513	0.554	0.508	0.659	0.544	0.486
[EX]	0.626	0.353	0.787	0.385	0.548	0.359
[OI]	0.440	0.436	0.410	0.453	0.527	0.438
[AP]	0.693	0.547	0.775	0.603	0.681	0.554
[DS]	0.809	0.738	0.942	0.752	0.729	0.753
[MT]	0.652	0.160	0.664	0.147	0.698	0.192
[PS]	0.588	0.572	0.530	0.528	0.705	0.681
[MM]	0.571	0.298	0.576	0.378	0.619	0.319

Table 4.13. Table showing the F_1 Score (F_1 Sc.), precision, and recall evaluated for the training ([T]) and validation ([V]) of the 1D CNN Model (refer to Section 3.3.8) on dataset TgF344AD Pair 18 on each class.

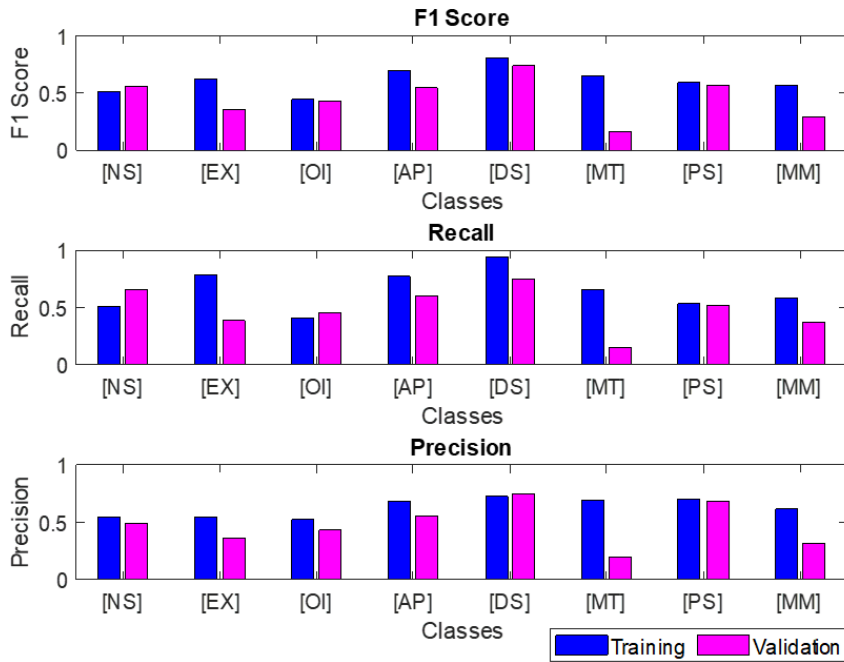


Figure 4.21. F_1 Score, precision, and recall evaluated for the training and validation of the 1D CNN Model (refer to Section 3.3.8) on dataset TgF344AD Pair 18 on each class.

4.15 MultiModal Model Training and Validation

We trained and evaluated three configurations of the MultiModal Model, which concatenates features derived from the TSRJI-CNN and CNN-LSTM models at different stages. Configurations c_1 and c_2 failed to improve classification performance - that could be attributed to the disparate learning parameters and training durations used for each model. Configuration c_3 of the MultiModal Model, utilizing pre-trained weights (checkpoints were selected based on minimal validation loss from the TSRJI-CNN and CNN-LSTM models), outperformed the individual TSRJI-CNN and CNN-LSTM methods, with validation accuracy of 0.896. The F_1 scores, training, and validation results for all configurations are detailed in Table 4.14 and Figure 4.22. For an analysis of the configuration (c_3) on the TgF344AD Pair 18 dataset, refer to the experiments section 4.16.

MultiModal Models Class Performance Metrics on TgF344AD Pair 18						
Class	[T] MM c_1	[V] MM c_1	[T] MM c_2	[V] MM c_2	[T] MM c_3	[V] MM c_3
[NS]	0.825	0.628	0.791	0.603	0.675	0.814
[EX]	0.917	0.593	0.815	0.665	0.615	0.911
[OI]	0.789	0.500	0.717	0.512	0.589	0.588
[AP]	0.856	0.658	0.866	0.681	0.610	0.640
[DS]	0.966	0.538	0.935	0.405	0.814	0.867
[MT]	0.949	0.479	0.959	0.639	0.838	0.902
[PS]	0.857	0.798	0.831	0.722	0.768	0.879
[MM]	0.954	0.580	0.943	0.687	0.782	0.896

Table 4.14. Table showing the class performance F_1 score metrics for training ([T]) and validation ([V]) on dataset TgF344AD Pair 18 of MultiModal (MM) Model configurations c_1 , c_2 and c_3 .

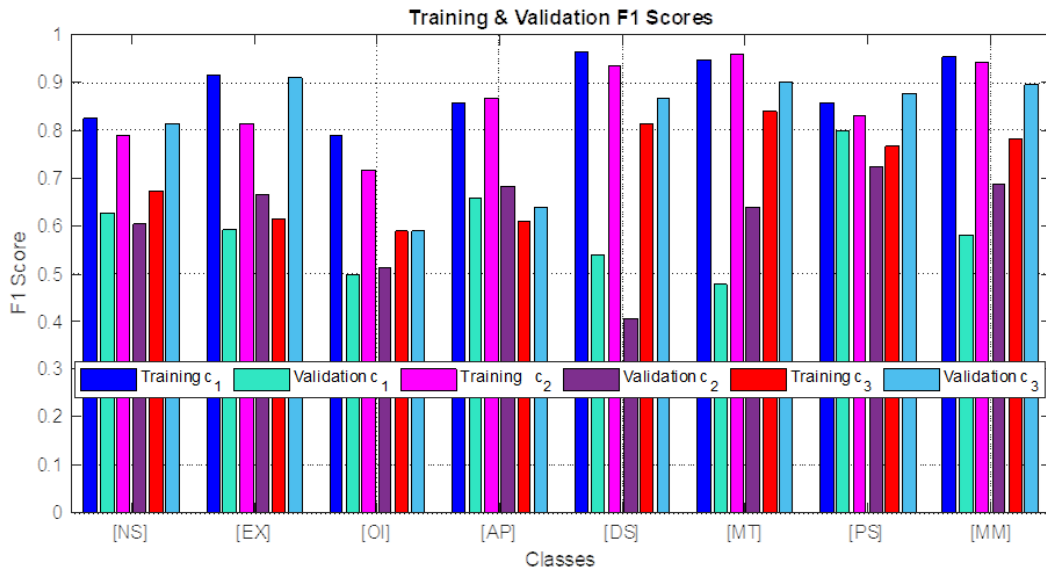


Figure 4.22. F_1 score class performance for training ([T]) and validation ([V]) on dataset TgF344AD Pair 18 of MultiModal Model configurations c_1 , c_2 and c_3 .

4.16 MultiModal Model Configuration c_3 Training and Validation

The MultiModal Model configuration c_3 with separately pre-trained weights on the TSRJI-CNN and CNN-LSTM model. The models successfully extract the features to classify the actions on the validation set of dataset TgF344AD Pair 18 with an overall validation accuracy of 0.896. The model exhibited its lowest performance for the [OI] (olfactory exploration) class, achieving an F_1 score of 0.588. This underperformance was particularly noticeable in scenarios where the rats were in close proximity, with one engaged in self-cleaning behavior while moving its head around the other rat. Such instances often led to misclassifications by the model. A more detailed analysis of these cases, including a comparative study with human predictions and a visual assessment of the model’s predictions, is presented in Section 4.19. Table 4.15, and Figures 4.23 and 4.24 depict the F_1 score, recall, and precision of the model training and validation on dataset TgF344AD Pair 18. For further description, we denote the model as MM_{c_3} .

MultiModal Model configuration c_3 Class Performance Metrics on TgF344AD Pair 18						
Class	[T] F_1 Sc.	[V] F_1 Sc.	[T] recall	[V] recall	[T] precision	[V] precision
[NS]	0.675	0.814	0.706	0.865	0.738	0.801
[EX]	0.615	0.911	0.668	0.919	0.615	0.946
[OI]	0.589	0.588	0.612	0.685	0.644	0.576
[AP]	0.610	0.640	0.612	0.631	0.648	0.692
[DS]	0.814	0.867	0.864	0.854	0.811	0.914
[MT]	0.838	0.902	0.846	0.913	0.866	0.907
[PS]	0.768	0.879	0.803	0.876	0.767	0.889
[MM]	0.782	0.896	0.814	0.903	0.785	0.917

Table 4.15. Table showing the F_1 Score (F_1 Sc.), precision, and recall evaluated for the training ([T]) and validation ([V]) of the MultiModal Model configuration c_3 (refer to Section 3.3.9) on dataset TgF344AD Pair 18 on each class.



Figure 4.23. F_1 Score, precision, and recall evaluated for the training and validation of the MultiModal Model configuration c_3 (refer to Section 3.3.9) on dataset TgF344AD Pair 18 on each class.

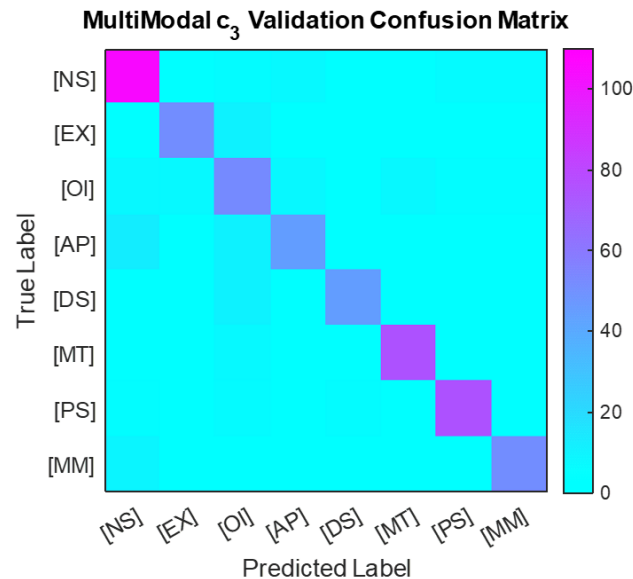


Figure 4.24. Validation Confusion Matrix of the classes predicted by the MultiModal Model configuration c_3 (refer to Section 3.3.9) on dataset TgF344AD Pair 18.

4.17 Model Comparison

In Table 4.16, we compare the implemented methods based on the validation F_1 score and the overall validation accuracy on dataset TgF344AD Pair 18. The best method to predict the behavioral classes on dataset TgF344AD Pair 18 was the MultiModal Model configuration c_3 with a validation accuracy of 0.862 (for its full report, refer to Experiment section 4.16). The second overall best model is the TSRJI-CNN, which has a validation accuracy of 0.696, followed by the CNN-LSTM model, which has a validation accuracy of 0.669.

Model Comparison Validation F_1 Score and Accuracy					
Class / Model	CNN + LSTM	TSRJI	Pivot TSRJI	MM c_2	MM c_3
[NS]	0.595	0.699	0.447	0.603	0.814
[EX]	0.722	0.791	0.565	0.665	0.911
[OI]	0.585	0.557	0.123	0.512	0.588
[AP]	0.273	0.764	0.366	0.681	0.640
[DS]	0.381	0.667	0.445	0.405	0.867
[MT]	0.838	0.358	0.540	0.639	0.902
[PS]	0.773	0.826	0.641	0.722	0.879
[MM]	0.745	0.754	0.638	0.687	0.896
Validation Accuracy	0.669	0.696	0.492	0.667	0.862
Class / Model	ST GCN c_2	3D CNN	1D CNN	Rule-Based	
[NS]	0.433	0.118	0.554	0.840	
[EX]	0.000	0.102	0.353	0.168	
[OI]	0.292	0.120	0.436	0.374	
[AP]	0.000	0.244	0.547	0.285	
[DS]	0.000	0.216	0.738	0.302	
[MT]	0.639	0.262	0.160	0.110	
[PS]	0.514	0.467	0.572	0.155	
[MM]	0.156	0.232	0.298	0.000	
Validation Accuracy	0.345	0.296	0.498	0.279	

Table 4.16. Table showing the F_1 score and validation accuracy across implemented models on dataset TgF344AD Pair 18. MM is a shortcut for the MultiModal model.

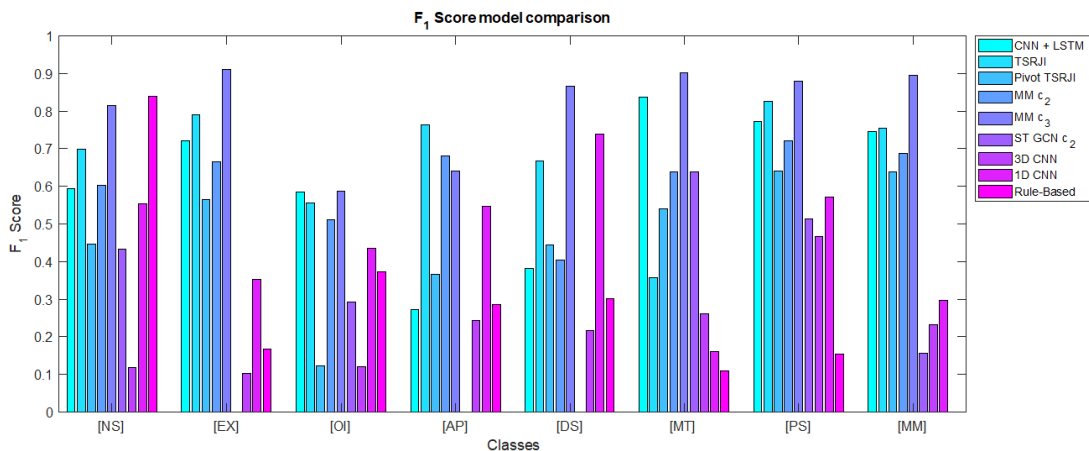


Figure 4.25. Figure depiction of the validation F_1 score across implemented models and denoted actions on dataset TgF344AD Pair 18.

4.18 Action Initiator Classification

The TSRJI-CNN models were specifically trained to identify the initiators in action classes where such identification is meaningful, namely [AP], [DT], [MT], and [OI]. This training was conducted on the relevant class subsets of dataset TgF344AD Pair 18, incorporating the identity swap augmentation method as denoted in 3.3.2, with the initiator labels being adjusted accordingly. The TSRJI-CNN models effectively recognized the initiators across all these action classes. This model was selected because the TSRJ images inherently contain information about the identity of the specific skeleton, unlike the camera view images used in the CNN-LSTM model. Each initiator class TSRJI-CNN model was trained on 30 epochs and batch size of 16. The results are depicted in Table 4.17 and Figure 4.26.

TSRJI-CNN Initiator Models Performance Metrics		
TSRJI-CNN Model	Training Accuracy	Validation Accuracy
Initiator Mounting	0.889	1.000
Initiator Olfactory Exploration	0.939	0.975
Initiator Approach	0.983	1.000
Initiator Detach	1.000	1.000

Table 4.17. Table showing the training and validation accuracy of TSRJI-CNN Models for differentiating initiators of dataset TgF344AD Pair 18 subset initiator classes [AP], [DT], [MT], and [OI].

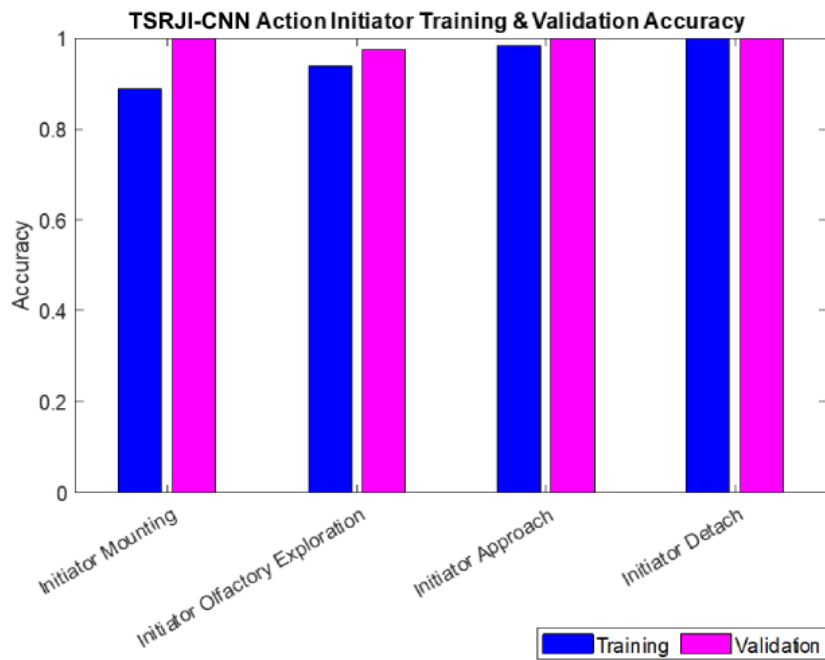


Figure 4.26. Accuracy evaluated for the training and validation of the TSRJI-CNN Initiator Models on dataset TgF344AD Pair 18 on subset initiator classes [AP], [DT], [MT], and [OI].

4.19 Comparison of human expert classification with an automatic method

We compared 200 sequence windows of predictions made by the MMc_3 model and predictions made by a person educated in the labeling process and with expertise in rat behavior. The mentioned individual labeling 200 sequence windows differs from the person creating dataset TgF344 AD Pair 18, used for training of the model MMc_3 . Class [DT] (detach) was considered as a [NS] (No Social Interaction). We compared the human predictions and model predictions by accuracy types denoted in Section 3.3.11. The results are depicted in Table 4.18 and 4.19. Predictions on subsets AD1-AD3 and AD3-AD4 were compared.

The experiment resulted well for the [NS],[OI] and [AP]. As for the [PS] class, the model tends to over-predict the interactions compared to the person - upon visual assessment of the actions. The model would benefit from seeing a longer sequence window for correctly assigning the [PS] class. The person labeling the passive interaction was biased by the knowledge that rats need longer to stay still next to each other before their interaction is classified as such - the model does not have such knowledge. Similarly, the model predicts the [AP] class a sequence window or two sooner, compared to the person with the labeling experiment task - which is not essentially wrong, as we

did label the TgF344AD Pair 18 dataset as such. We will visually discuss the examples of sequence windows in the experiment subsection 4.19.1. From the 200 sequence windows, four were denoted as [1004] - we were not able to pre-process those sequence windows correctly for the model (see Section 2.11 and experiment Section 4.20).

For predicting the actions' initiator, the trained TSRJI-CNN models (refer to Experiment section 4.18) managed to predict 100% of the initiations correctly (we compared the actions with the initiator assigned where the human predictions equaled the model predictions - 37 sequence windows).

Accuracy Comparison	
Accuracy Type	Value
A_{c1}	71.00%
A_{lb} (L=1)	81.50%
A_{lb} (L=2)	86.00%
A_{eb}	64.00%

Table 4.18. Table comparing different types of accuracies: accuracy (A_{c1}), latency-based accuracy with a tolerance of 1 sequence window (A_{lb} (L=1)), latency-based accuracy with a tolerance of 2 sequence windows (A_{lb} (L=2)), and event-based accuracy (A_{eb}).

Class Count Comparison		
Class	Human Predictions	Model Predictions
[NS]	143	107
[EX]	2	10
[OI]	38	46
[AP]	13	20
[PS]	4	10
[MM]	0	3
[1004]	0	4

Table 4.19. Table comparing the count of classes as predicted by a human (not the one creating the training dataset TgF344AD Pair 18) and the model MMc_3 .

4.19.1 Visual inspection of MMc_3 model predictions

The following experiment depicts the visual assessment of the predicted actions denoted in the TgF344 AD Pair 18 dataset (refer to Section 2.13.1). We prepared several video segments from the recorded video subsets, rendered with the recognized action as a text in the left corner of the video. If the segment was also included in the human expert comparison, the human prediction is depicted in the right corner of the video. Each segment contains 20 predictions, which equals 10 seconds of video. The presented subsets were not used for the training and validation of the denoted methods. The actions were predicted by the model MMc_3 (refer to section 3.3.9). In the experiment, we view the correct predictions and try to pinpoint the misclassifications. The full file folder of depicted segments together with concrete video links is provided in the belonging footnote.¹

After visual inspection of the provided segments, we can mostly see the correct predictions on the olfactory inspection [OI] and approach [AP] (the most common classes in the provided segments, together with the no social interaction [NS] class). However, the model tends to predict the [AP] class, even when the movement towards the other rat is small, which can result in a wrong prediction (for example, segment 2 of subset WT2-WT3²) or the approach does not directly end up with snout to body distance - which is right by our labeling logic of the approach class but differs from the human expert labeling of the class (refer to experiment 4.19 above, and video segment 6 of a subset WT2-WT3³, or segment 1 of a subset AD6-WT5⁴).

In video segment 2 of subset AD8-WT7, we can see a wrongly predicted [OI] class instead of passive interaction [PS] - the alignment of the rats and the snout position near the tail base of the other rat all suggest that the class is correctly predicted as [OI], but the interaction should be classified as [PS] because the movement of the rats is rather stationary. In segment 2 of the subset, we can see the correct prediction of the [PS] class. A problematic scenario for the model is situations when the two rats are in close proximity, slowly walking in opposite directions - resembling either [PS] or mutual exploration [EX] (in this case, both rats inspecting the tail base of the other). This can be seen in segments 2 of subset AD7-WT8⁵ and 1 of subset WT2-WT3⁶.

In segment 2 of subset AD6-WT5⁷ we can see a mimicking [MM] class being predicted, even when only one of the rats is rearing - the right example of [MM] can be seen in segment 5 of a subset AD7-WT8⁸.

The segments with human expert predictions are segments 1 and 2 of subset AD1-AD3^{9 10} and segments 1 and 2 of subset AD3-AD4^{11 12}.

¹ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video

² https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/wt2_wt3_seg2.avi

³ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/wt2_wt3_seg6.avi

⁴ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/ad6_wt5_seg1.avi

⁵ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/ad7_wt8_seg2.avi

⁶ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/wt2_wt3_seg1.avi

⁷ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/ad6_wt5_seg2.avi

⁸ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/ad7_wt8_seg5.avi

⁹ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/ad1_ad3_seg1_expert.avi

¹⁰ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/ad1_ad3_seg2_expert.avi

¹¹ https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/ad3_ad4_seg1_expert.avi

¹² https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/-/tree/main/predictions_video/ad3_ad4_seg2_expert.avi

Chapter 5

Data Analysis Results

5.1 Dataset TgF344 AD Pair 18 MMc_3 model action predictions Analysis Results

We used the model MMc_3 to classify the sequences and further analyze the recorded subsets of interaction Day 2 of the experiments (refer to Section 2.4). We analyzed time spent in denoted interactions between rats TgF344-AD (AD) and F344 (WT) of age groups of six and ten months to determine if the transgenic genotype impacts the behavior of the rats across the denoted actions. We compare different rat pairings: AD-AD, WT-WT, and AD-WT. If an initiator was assigned for the given action, we analyze the time spent as an initiator of the interaction between AD and WT rat types. For the methods used in the analysis, refer to Section 3.4. We analyzed the first 5 to 525 seconds of each recording (32 subsets altogether) - the analysis is done for every half-second sequence window of the subset. The results of the 520-second analysis are depicted in Tables 5.1 and 5.2. The analysis is done on the time spent t_a in the given interaction differentiated between the rat types or their pairings. We depict t_a for different actions in the form of boxplots in Figure 5.1 and 5.2. We depict t_a for different initiator rat types and actions in the form of boxplots in Figure 5.3 and 5.4.

We tested the normality of each group denoted (AD-AD, WT-WT, AD-WT, Initiator AD, Initiator WT) for each predicted action (refer to Section 2.13) using Shapiro-Wilk Test. Results of the normality test differentiated between actions, pairings, or initiator types - to maintain consistency in our analysis, we continued the analysis with non-parametric methods.

We used the Kruskal-Wallis test to compare if the populations of t_a for given pairings and actions differ significantly (with a significance alpha level set as 0.05). The results are depicted in Table 5.1 for the 520-second analysis. The null hypothesis H_0 for the Kruskal-Wallis test states that the medians of all groups are equal. As shown in Table 5.1, the null hypothesis was not rejected throughout all the actions of pairings AD-AD, WT-WT, AD-WT, and both age groups. Based on the results, we did not continue with post-hoc analysis.

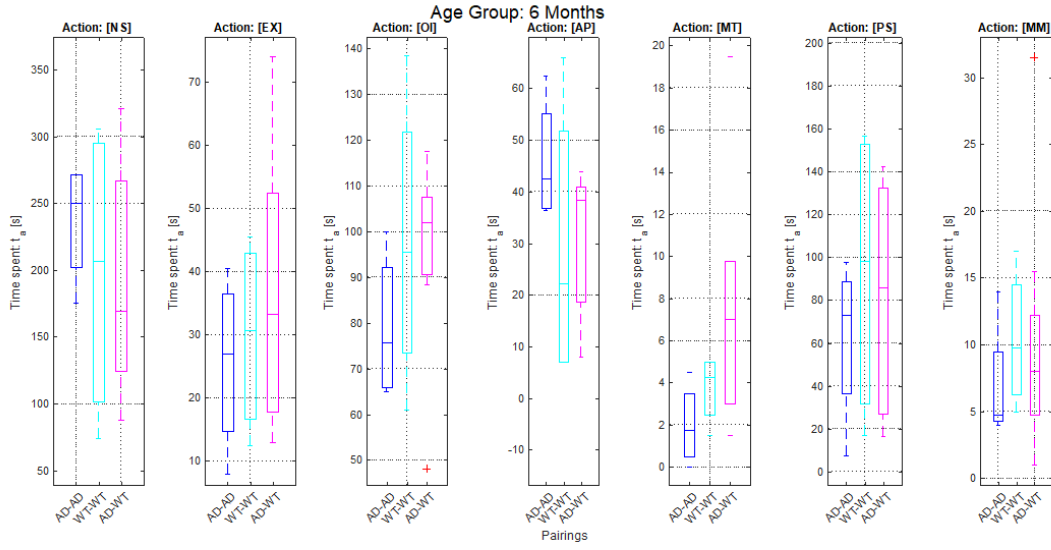


Figure 5.1. The boxplot depiction of rat pairings (AD-AD: blue, WT-WT: cyan, AD-WT: magenta) of time spent t_a in given predicted action for age group 6 months and 520 seconds for subset analysis.

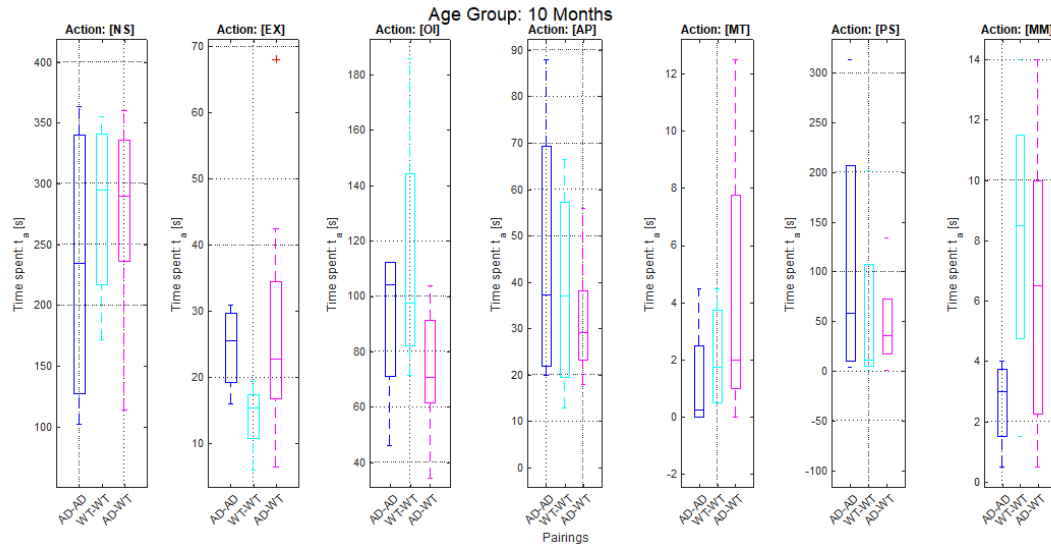


Figure 5.2. The boxplot depiction of rat pairings (AD-AD: blue, WT-WT: cyan, AD-WT: magenta) of time spent t_a in given predicted action for age group 10 months and 520 seconds for subset analysis.

	Actions						
	[NS]	[EX]	[OI]	[AP]	[MT]	[PS]	[MM]
6 months	0.8129	0.6482	0.2844	0.5522	0.0833	0.6723	0.5082
10 months	0.8180	0.1110	0.2410	0.9055	0.2969	0.5873	0.3610

Table 5.1. P-values for different age groups (6 months and 10 months) across time spent t_a in predicted actions in group pairings AD-AD, WT-WT, AD-WT, as determined by the Kruskal-Wallis test.

We used the Mann-Whitney U test to compare if the populations of t_a for given initiator rat types (AD, WT) significantly differ across given actions (with a significance alpha level set as 0.05). The results are depicted in Table 5.2. The null hypothesis H_0 for the Mann-Whitney U test states that the AD and WT rat populations are equal.

As shown in Table 5.2, the null hypothesis was not rejected throughout all the initiator actions, and no significant difference was found between the TgF344-AD (AD) and F344 (WT) rat types.

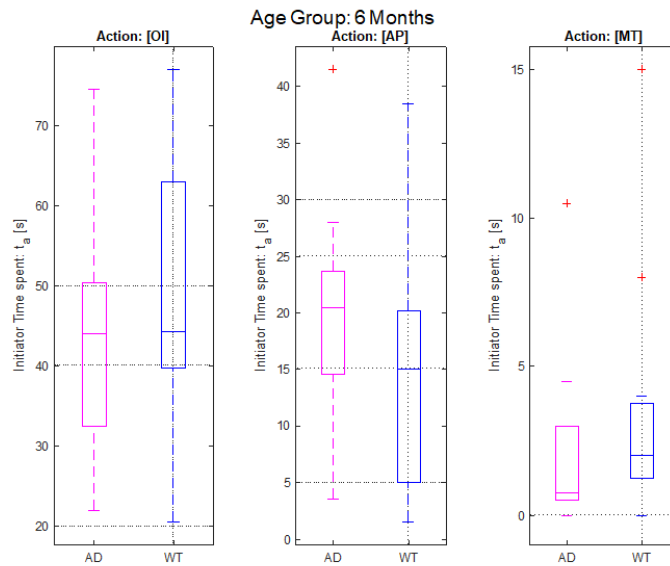


Figure 5.3. The boxplot depiction of time spent t_a in initiation by rat type AD or WT of given predicted action for age group 6 months and 520 seconds for subset analysis.

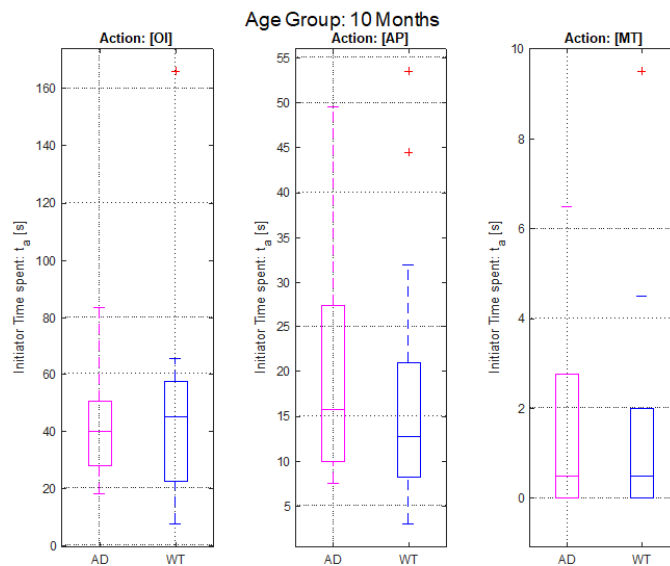


Figure 5.4. The boxplot depiction of time spent t_a in initiation by rat type AD or WT of given predicted action for age group 10 months and 520 seconds for subset analysis.

	Initiator Actions		
	[OI]	[AP]	[MT]
6 months (AD WT)	0.6921	0.1221	0.0989
10 months (AD WT)	0.9100	0.4174	0.7547

Table 5.2. P-values for AD and WT rat type comparisons across different initiator actions ([OI], [AP], [MT]) time spent t_a for age groups of 6 months and 10 months, as determined by the Mann-Whitney U test.

5.2 Dataset TgF344 AD Pair 18 Rule-Based Model action predictions Analysis Results

We used the Rule-Based Model to classify the sequences and further analyze the recorded subsets of interaction Day 2 of the experiments (refer to Section 2.4). The analysis follows the analysis described in Section 5.1, but we compare the action class [HH] for the pairings of the rats and the action class [SB] for the initiator analysis predicted by the Rule-Based Model.

The results of the Kruskal-Wallis test on rat pairings on time spent t_a in action [HH] are $p = 0.936$ for the rats aged six months and $p = 0.181$ for the rats aged ten months. The results of the Mann-Whitney U test on rat types AD and WT populations of time spent t_a as an initiator of action [SB] are $p = 0.608$ for the rats aged six months and $p = 0.805$ for the rats aged ten months. Neither test found a significant difference between the different rat types in actions predicted by the Rule-Based Model as [HH] and [SB]. The boxplots of the time spent t_a in action [HH] throughout different pairings are depicted in Figure 5.5. The boxplots of time spent t_a of initiator in action [SB] for TgF344-AD (AD) and F344 (WT) rat types are depicted in Figure 5.6.

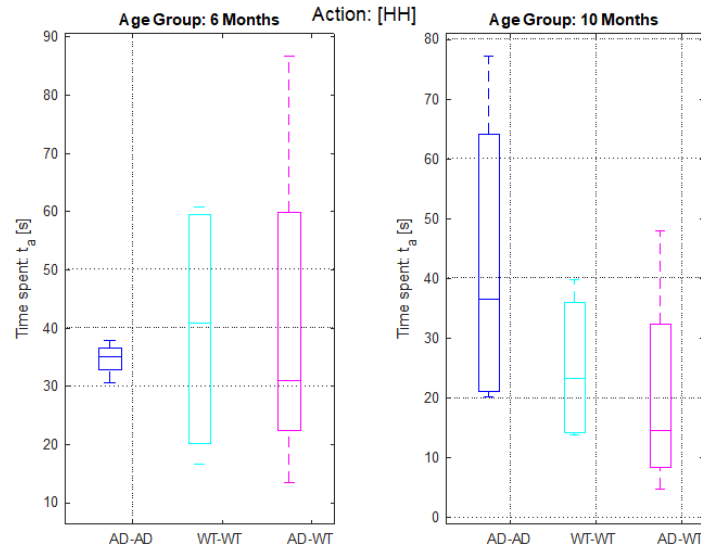


Figure 5.5. The boxplot depiction of time spent t_a in action [HH] throughout pairings AD-AD, WT-WT, and AD-WT for both age groups in the 520 seconds for subset analysis.

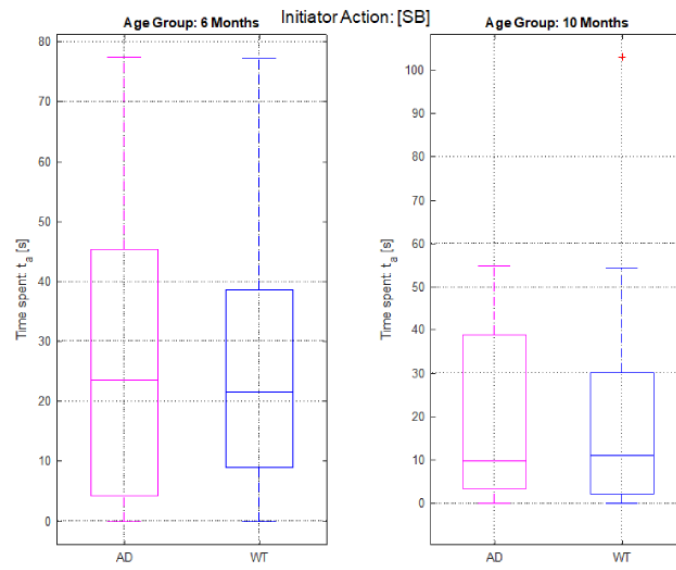


Figure 5.6. The boxplot depiction of time spent t_a of rat type initiators in action [SB] for both age groups in the 520 seconds for subset analysis.

Chapter 6

Conclusion

In this work, we addressed the classification of interactions between two rats from video, an area of interest in behavioral neuroscience. To do so, we have implemented and experimented with several methods and modalities, from which we have built a multi-modal network, denoted as MMc_3 . We have also explored a classification of the rat's interactions by a rule-based classifier. If applicable, we focused on predicting the initiator of denoted actions (e.g. assigning the identity of approaching rat).

We have collected and partially labeled a dataset denoted as TgF344-AD Pair 18 for the purposes of the work.

The video subsets were recorded from four cameras to capture the rats in detail. To obtain the rat's pose in each frame, we have used and trained the state-of-the-art algorithm DeepLabCut (DLC) and the corresponding model DLCRNet-MS5. Leveraging multiple views for pose estimation allowed us to construct a 3D coordinate system representation of the rats. We implemented imputation models, such as point distribution model, to correct the rat's skeleton in frames where DLC encountered limitations, particularly in scenarios of marker misestimation across multiple views.

One of the observed limitations of the DLC algorithm was tracking the identity of the rodent in complex situations where, for example, the animals occluded themselves. We have denoted a method to improve the tracking and correct the misassigned identities, using the information from multiple views.

We have used the implemented methods to analyze the collected dataset TgF344-AD Pair 18 and measure the time spent throughout the denoted actions. We have used statistical methods to compare pairings of different rat types TgF344-AD and F344-control subjects to inspect the impact of Alzheimer's disease on the rat's interactions.

We have trained the multi-modal MMc_3 network with overall validation accuracy of 86.2% on dataset TgF344-AD Pair 18 - a reliable classification of the rat's interactions. We have trained TSRJI-CNN models to classify the initiator of given actions. The rule-based classifier was sufficient for contact classes of particular parts of the rat's body, but the deep learning methods outperformed the set of rules on more complex actions, such as approach. We did not find any significant differences between the TgF344-AD and F344 in denoted actions.

6.1 Future Work

For future work we propose the following:

- Classification of actions without a predefined sequence window length - classification of actions such as passive interaction or approach would both benefit from such methods.
- Larger sample size for the behavioral investigation of the rats - this was a pilot research, and at six months of age, the rats depicted the expected behavior. However, a larger sample size (number of examined rats) is encouraged.
- The automated methods can be retrained and employed on different types of actions or different strains of rodents (e.g. mice, black and white rats, etc.) - with new action and marker labels.

References

- [1] BREIJYEH, Zeinab, and Rafik KARAMAN. Comprehensive Review on Alzheimer's Disease: Causes and Treatment. *Molecules*. 12, 2020, Vol. 25, pp. 5789. Available from DOI 10.3390/molecules25245789.
- [2] BLOOM, George. Amyloid- β and Tau: The Trigger and Bullet in Alzheimer Disease Pathogenesis. *JAMA neurology*. 12, 2014, Vol. 71. Available from DOI 10.1001/jamaneurol.2013.5847.
- [3] KUMAR, Anil, Jaskirat SIDHU, Amandeep GOYAL, and Jack W TSAO. *Alzheimer Disease*. StatPearls Publishing, Treasure Island (FL), 2022. Available from <http://europepmc.org/books/NBK499922>.
- [4] JOST, B C, and G T GROSSBERG. The evolution of psychiatric symptoms in Alzheimer's disease: a natural history study. *Journal of the American Geriatrics Society*. 9, 1996, Vol. 44, No. 9, pp. 1078–1081. ISSN 0002-8614. Available from DOI 10.1111/j.1532-5415.1996.tb02942.x. Available from <https://doi.org/10.1111/j.1532-5415.1996.tb02942.x>.
- [5] SINGLETON, Ellen, Jay FIELDHOUSE, Jochum van 'T HOOFT, Marta SCARIONI, Marie-Paule ENGELEN, Sietske SIKKES, Casper BOER, Diana BOCANCEA, Esther den BERG, Philip SCHELTENS, Wiesje FLIER, Janne PAPMA, Yolande PIJNENBURG, and Rik OSSENKOPPELE. Social cognition deficits and biometric signatures in the behavioural variant of Alzheimer's disease. *Brain*. 12, 2022, Vol. 146. Available from DOI 10.1093/brain/awac382.
- [6] ZETTERBERG, Henrik, and Niklas MATTSSON. Understanding the cause of sporadic Alzheimer's disease. *Expert Review of Neurotherapeutics*. Taylor & Francis, 2014, Vol. 14, No. 6, pp. 621–630. Available from DOI 10.1586/14737175.2014.915740. Available from <https://doi.org/10.1586/14737175.2014.915740>.
- [7] TANZI, Rudolph E. The genetics of Alzheimer disease. *Cold Spring Harbor perspectives in medicine*. 10, 2012, Vol. 2, No. 10, pp. a006296. ISSN 2157-1422. Available from DOI 10.1101/cshperspect.a006296. Available from <https://europepmc.org/articles/PMC3475404>.
- [8] SHEA, Yat-Fung, Leung-Wing CHU, Angel On-Kei CHAN, Joyce HA, Yan LI, and You-Qiang SONG. A systematic review of familial Alzheimer's disease: Differences in presentation of clinical features among three mutated genes and potential ethnic differences. *Journal of the Formosan Medical Association*. 2016, Vol. 115, No. 2, pp. 67–75. ISSN 0929-6646. Available from DOI <https://doi.org/10.1016/j.jfma.2015.08.004>. Available from <https://www.sciencedirect.com/science/article/pii/S0929664615003022>.
- [9] DRUMMOND, Eleanor, and Thomas WISNIEWSKI. Alzheimer's disease: experimental models and reality. *Acta Neuropathologica*. 12, 2017, Vol. 133. Available from DOI 10.1007/s00401-016-1662-x.
- [10] MCKEAN, Natasha, Renee HANDLEY, and Russell SNELL. A Review of the Current Mammalian Models of Alzheimer's Disease and Challenges That Need to Be Overcome. *International Journal of Molecular Sciences*. 12, 2021, Vol. 22, pp. 13168. Available from DOI 10.3390/ijms222313168.
- [11] COHEN, Robert, Kavon REZAI-ZADEH, Tara WEITZ, Altan RENTSENDORJ, David GATE, Inna SPIVAK, Yasmin BHOLAT, Vitaly VASILEVKO, Charles GLABE, Joshua BREUNIG, Pasko RAKIC, Hayk DAVTYAN, Michael AGADJANYAN, Vladimir KEPE, Jorge BARRIO, Serguei BANNYKH, Christine SZEKELY, Robert PECHNICK, and Terrence TOWN. A Transgenic Alzheimer Rat with Plaques, Tau Pathology, Behavioral Impairment, Oligomeric A β , and Frank Neuronal Loss. *The Journal of*

- neuroscience : the official journal of the Society for Neuroscience*. 12, 2013, Vol. 33, pp. 6245–6256. Available from DOI 10.1523/JNEUROSCI.3672-12.2013.
- [12] ANAND, Kuljeet Singh, and Vikas DHIKAV. Hippocampus in health and disease: An overview. *Annals of Indian Academy of Neurology*. 2012, Vol. 15, pp. 239 - 246. Available from <https://api.semanticscholar.org/CorpusID:11156786>.
- [13] PENTKOWSKI, Nathan S, Laura E BERKOWITZ, Shannon M THOMPSON, Emma N DRAKE, Carlos R OLGUIN, and Benjamin J CLARK. Anxiety-like behavior as an early endophenotype in the TgF344-AD rat model of Alzheimer’s disease. *Neurobiology of Aging*. 2018, Vol. 61, pp. 169–176. ISSN 0197-4580. Available from DOI <https://doi.org/10.1016/j.neurobiolaging.2017.09.024>. Available from <https://www.sciencedirect.com/science/article/pii/S0197458017303202>.
- [14] TOURNIER, Benjamin, Cristina BARCA, Aïda FALL, Yesica GLORIA, Léa MEYER, Kelly CEYZERIAT, and Philippe MILLET. Spatial reference learning deficits in absence of dysfunctional working memory in the TgF344-AD rat model of Alzheimer’s disease. *Genes, brain, and behavior*. 12, 2020, Vol. 20, pp. e12712. Available from DOI 10.1111/gbb.12712.
- [15] SIGAL, Leonid. *Human Pose Estimation*. Available from DOI 10.1007/978-3-030-63416-2_584. Available from https://doi.org/10.1007/978-3-030-63416-2_584.
- [16] ANDRILUKA, Mykhaylo, Leonid PISHCHULIN, Peter GEHLER, and Bernt SCHIELE. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014. pp. 3686–3693. Available from DOI 10.1109/CVPR.2014.471.
- [17] LIU, Wu, and Tao MEI. Recent Advances of Monocular 2D and 3D Human Pose Estimation: A Deep Learning Perspective. *ACM Computing Surveys*. 12, 2022, Vol. 55. Available from DOI 10.1145/3524497.
- [18] TOSHEV, Alexander, and Christian SZEGEDY. DeepPose: Human Pose Estimation via Deep Neural Networks. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014. pp. 1653–1660. Available from DOI 10.1109/CVPR.2014.214.
- [19] ZHENG, Ce, Wenhan WU, Taojiannan YANG, Sijie ZHU, Chen CHEN, Ruixu LIU, Ju SHEN, Nasser KEHTARNAVAZ, and Mubarak SHAH. Deep Learning-based Human Pose Estimation: A Survey. *ACM Computing Surveys*. 2020, Vol. 56, pp. 1 - 37. Available from <https://api.semanticscholar.org/CorpusID:195493170>.
- [20] STENUM, Jan, Kendra M CHERRY-ALLEN, Connor O PYLES, Rachel REETZKE, Michael F VIGNOS, and Ryan T ROEMMICH. Applications of Pose Estimation in Human Health and Performance across the Lifespan. *Sensors (Basel, Switzerland)*. 2021, Vol. 21. Available from <https://api.semanticscholar.org/CorpusID:243794363>.
- [21] LAUER, Jessy, Mu ZHOU, Shaokai YE, William MENEGAS, Tanmay NATH, Mohammed Mostafizur RAHMAN, Valentina Di SANTO, Daniel SOBERANES, Guoping FENG, Venkatesh N MURTHY, George LAUDER, Catherine DULAC, Mackenzie W MATHIS, and Alexander MATHIS. Multi-animal pose estimation and tracking with DeepLabCut. *bioRxiv*. Cold Spring Harbor Laboratory, 2021. Available from DOI 10.1101/2021.04.30.442096. Available from <https://www.biorxiv.org/content/early/2021/04/30/2021.04.30.442096>.
- [22] CAO, Zhe, Tomas SIMON, Shih-En WEI, and Yaser SHEIKH. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. pp. 1302–1310. Available from DOI 10.1109/CVPR.2017.143.
- [23] BEWLEY, Alex, ZongYuan GE, Lionel OTT, Fabio Tozeto RAMOS, and Ben UCROFT. Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 3464–3468. Available from <https://api.semanticscholar.org/CorpusID:16034699>.
- [24] CHAABANE, Mohamed, Peter ZHANG, J Ross BEVERIDGE, and Stephen O’HARA. DEFT: Detection Embeddings for Tracking. *ArXiv*. 2021, Vol. abs/2102.02267. Available from <https://api.semanticscholar.org/CorpusID:231802258>.

- [25] WELCH, Greg, and Gary BISHOP. An Introduction to Kalman Filter. In: *International Conference on Computer Graphics and Interactive Techniques*. 1995. Available from <https://api.semanticscholar.org/CorpusID:215767582>.
- [26] WOJKE, Nicolai, Alex BEWLEY, and Dietrich PAULUS. Simple online and real-time tracking with a deep association metric. In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017. pp. 3645–3649. Available from DOI 10.1109/ICIP.2017.8296962.
- [27] MORSHED, Md Golam, Tangina SULTANA, Aftab ALAM, and Young-Koo LEE. Human Action Recognition: A Taxonomy-Based Survey, Updates, and Opportunities. *Sensors*. 2023, Vol. 23, No. 4. ISSN 1424-8220. Available from DOI 10.3390/s23042182. Available from <https://www.mdpi.com/1424-8220/23/4/2182>.
- [28] SUN, Zehua, Qihong KE, Hossein RAHMANI, Mohammed BENNAMOUN, Gang WANG, and Jun LIU. Human Action Recognition From Various Data Modalities: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2023, Vol. 45, No. 3, pp. 3200–3225. Available from DOI 10.1109/TPAMI.2022.3183112.
- [29] SARGANO, Allah Bux, Plamen ANGELOV, and Zulfiqar HABIB. A Comprehensive Review on Handcrafted and Learning-Based Action Representation Approaches for Human Activity Recognition. *Applied Sciences*. 2017, Vol. 7, No. 1. ISSN 2076-3417. Available from DOI 10.3390/app7010110. Available from <https://www.mdpi.com/2076-3417/7/1/110>.
- [30] KIPF, Thomas, and Max WELLING. Semi-Supervised Classification with Graph Convolutional Networks. 12, 2016.
- [31] NGUYEN, Hung-Cuong, Thi-Hao NGUYEN, Rafał SCHERER, and Van-Hung LE. Deep Learning for Human Activity Recognition on 3D Human Skeleton: Survey and Comparative Study. *Sensors*. 2023, Vol. 23, No. 11. ISSN 1424-8220. Available from DOI 10.3390/s23115121. Available from <https://www.mdpi.com/1424-8220/23/11/5121>.
- [32] ARUNNEHRU, J, G CHAMUNDEESWARI, and S Prasanna BHARATHI. Human Action Recognition using 3D Convolutional Neural Networks with 3D Motion Cuboids in Surveillance Videos. *Procedia Computer Science*. 2018, Vol. 133, pp. 471–477. ISSN 1877-0509. Available from DOI <https://doi.org/10.1016/j.procs.2018.07.059>. Available from <https://www.sciencedirect.com/science/article/pii/S1877050918310044>.
- [33] WANG, Pichao, Wanqing LI, Chuankun LI, and Yonghong HOU. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowledge-Based Systems*. 2018, Vol. 158, pp. 43–53. ISSN 0950-7051. Available from DOI <https://doi.org/10.1016/j.knosys.2018.05.029>. Available from <https://www.sciencedirect.com/science/article/pii/S0950705118302582>.
- [34] LIU, Mengyuan, Chen CHEN, and Hong LIU. 3D action recognition using data visualization and convolutional neural networks. In: 2017. Available from DOI 10.1109/ICME.2017.8019438.
- [35] CAETANO, Carlos Antonio, Francois BREMOND, and William Robson SCHWARTZ. Skeleton Image Representation for 3D Action Recognition Based on Tree Structure and Reference Joints. *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. 2019, pp. 16–23. Available from <https://api.semanticscholar.org/CorpusID:202565715>.
- [36] SUN, Zehua, Jun LIU, Qihong KE, and H RAHMANI. Human Action Recognition From Various Data Modalities: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2020, Vol. 45, pp. 3200–3225. Available from <https://api.semanticscholar.org/CorpusID:229349238>.
- [37] LI, Maosen, Siheng CHEN, Xu CHEN, Ya ZHANG, Yanfeng WANG, and Qi TIAN. Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3590–3598. Available from <https://api.semanticscholar.org/CorpusID:139101198>.
- [38] YU, Bruce X B, Yan LIU, Xiang ZHANG, Sheng-hua ZHONG, and Keith C C CHAN. MMNet: A Model-Based Multimodal Network for Human Action

- Recognition in RGB-D Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2023, Vol. 45, No. 3, pp. 3522–3538. Available from DOI 10.1109/TPAMI.2022.3177813.
- [39] HARTLEY, Richard, and Andrew ZISSERMAN. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. ISBN 9780521540513. Available from DOI 10.1017/CBO9780511811685. Available from <https://www.cambridge.org/core/product/identifier/9780511811685/type/book>.
- [40] COOTES, T F, and Chris TAYLOR. Statistical Models of Appearance for computer vision. 12, 2004.
- [41] RABBATH, C A, and D CORRIVEAU. A comparison of piecewise cubic Hermite interpolating polynomials, cubic splines and piecewise linear functions for the approximation of projectile aerodynamics. *Defence Technology*. 2019, Vol. 15, No. 5, pp. 741–757. ISSN 2214-9147. Available from DOI <https://doi.org/10.1016/j.dt.2019.07.016>. Available from <https://www.sciencedirect.com/science/article/pii/S2214914719301187>.
- [42] MARSHALL, Jesse D, Ugne KLIBAITE, Amanda GELLIS, Diego E ALDARONDO, Bence P OLVECZKY, and Tim DUNN. The PAIR-R24M Dataset for Multi-animal 3D Pose Estimation. *NeurIPS*. 2021. Available from https://openreview.net/forum?id=-wVv1_UPr8.
- [43] PANG, Chen, Xuequan LU, and Lei LYU. *Skeleton-based Action Recognition through Contrasting Two-Stream Spatial-Temporal Networks*. Available from DOI 10.48550/arXiv.2301.11495.
- [44] YU, Bing, Haoteng YIN, and Zhanxing ZHU. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In: 2018. pp. 3634–3640. Available from DOI 10.24963/ijcai.2018/505.
- [45] LIANG, Kuo Kan. *Efficient conversion from rotating matrix to rotation axis and angle by extending Rodrigues' formula*.
- [46] CHIU, Shian-Yu, Kun-Ru WU, and Yu-Chee TSENG. Two-Person Mutual Action Recognition Using Joint Dynamics and Coordinate Transformation. In: EAI, 2021. Available from DOI 10.4108/eai.20-11-2021.2314154.
- [47] HE, Kaiming, Xiangyu ZHANG, Shaoqing REN, and Jian SUN. Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. pp. 770–778. Available from DOI 10.1109/CVPR.2016.90.
- [48] MISHRA, Prabhakar, ChandraM PANDEY, Uttam SINGH, Anshul GUPTA, Chinmoy SAHU, and Amit KESHRI. Descriptive Statistics and Normality Tests for Statistical Data. *Annals of Cardiac Anaesthesia*. 12, 2019, Vol. 22, pp. 67–72. Available from DOI 10.4103/aca.ACA_157_18.

Appendix A

DeepLabCut (DLC) Project configuration

Throughout our work, we use DeepLabCut (DLC) version 2.3.5. The following paragraphs discuss the choices and notes for the DLC project configuration found and set throughout experimentation.

Our model operates in a multi-animal project mode to track two animals simultaneously. This mode introduces Part Affinity Fields (PAFs) to model spatial relationships between body parts across different animals and employs multi-instance learning. We also considered enabling the Identity parameter, which helps the model extract features to distinguish between animals (e.g., a red cross on an animal’s back). However, due to the full white color of the rats and the need to limit external markers or scents affecting social interactions (for purposes of the research), we did not enable this feature. This decision was also influenced by recommendations from DLC to avoid reducing model performance.

Given the number of labeled frames, computational efficiency, and the multi-animal context, we chose DeepLabCut Refined Network with Multi-Scale 5 (DLCRNet-MS5) as our default model architecture. The data augmentation method employed was DLC’s multi-animal-imagaug, designed explicitly for multi-animal scenarios. Additionally, due to frequent occlusions observed in the video footage, a skeleton-based tracking method was selected. We set up a data-driven, fully connected skeleton, which effectively handled occlusions between animals and outperformed ellipse and box tracking methods.

To label frames efficiently and select scenarios with a higher probability of being unseen by the model, frames for labeling were extracted using the automatic K-Means algorithm in the DLC library. For each extracted frame, nine markers were labeled for each present animal. Initially, we attempted model training without labeling occluded markers (e.g., if a rat’s head was hidden). However, intentionally losing track of a point introduced more errors than generalizing the network with occlusions. With sufficient occlusions labeled and PAFs set with a higher weight, the model learned to classify markers in those frames correctly. While label identity was not essential during the labeling process, it was maintained for testing purposes (while the identity parameter was enabled). Frames were labeled by a single individual.

To address occlusions in our research setup, we experimentally fine-tuned the DLCRNet-MS5 network’s training configuration with specific modifications. The primary adjustments are outlined below:

- Part Affinity Fields (PAFs) were enabled to facilitate the prediction of spatial relationships between body parts. The PAF width was set at 50, and the pairwise loss weight was configured to 0.6.
- Pairwise Huber loss was activated.
- Additional data augmentation was employed through the activation of the `mirror` setting. This technique expands the training dataset by horizontally flipping the images, enhancing the dataset size.

Considering the hyperparameters, the network was trained on 60,000 epochs with a batch size of eight and multi-step learning with the Adam optimizer. To view the results of the network, refer to section 4.1.

Appendix B

Contents of attachment, Shortcuts

B.1 Contents of attachment

The implemented processing pipeline in MATLAB 2023a, the implementation and processing pipeline for training the methods in PyTorch, camera recording using pyrealsense2 library, DeepLabCut project configuration and analysis code, and examples of videos with predictions of the implemented automatic methods are available online through a GitLab repository at <https://gitlab.fel.cvut.cz/kanoufad/automatic-classification-of-social-interactions-of-rats-from-video/>. The corresponding README file will provide a description of the shown files.

src/matlab-processing-pipeline

Implementation of the processing pipeline in MATLAB used on the recorded video subsets, including data preprocessing and prediction assessing.

src/models-developing-torch-pipeline

Python implementation of the methods with corresponding training and data loading protocols, implementation and preprocessing of modalities.

src/quad-view-pyrealsense-record

Quad view camera setup and recording, using pyrealsense library.

src/deeplabcut-project-network-configurations

DeepLabCut project and trained network configuration files.

src/dlc-subset-analysis

DeepLabCut video analysis using a trained network - called within the MATLAB pipeline.

src/action-recog-trained-weights

Trained model weights.

subset_video_example

Video subset example from 4 camera views.

B.2 Shortcuts

- DLC: DeepLabCut
- PDM: Point Distribution Model
- ST GCN: Spatio-Temporal Graph Convolutional Network
- CNN: Convolutional Neural Network
- LSTM: Long Short-Term Memory
- AD: Alzheimer's Disease
- TSRJI: Tree Structure Reference Joints Image
- PCA: Principal Component Analysis
- DLT: Direct Linear Transformation
- SVD: Singular Value Decomposition
- PAFs: Part Affinity fields