FACULDADE DE ENGENHARIA

Novel tools for Quantification and Automatic Classification of Behavior in Laboratory Animals - Advancing Computational Ethology

Ana Filipa Domingues Gerós

U.PORTO

Ana Filipa Domingues Gerós. Novel Tools for Quantification and Automatic Classification of Behavior in Laboratory Animals - Advancing Computational Ethology

D.FEUP **2022**

U.PORTO

DOUTORAMENTO

ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

Novel Tools for Quantification and Automatic Classification of Behavior in Laboratory Animals - Advancing Computational Ethology

Ana Filipa Domingues Gerós

D

2022

D 2022

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

NOVEL TOOLS FOR QUANTIFICATION AND AUTOMATIC CLASSIFICATION OF BEHAVIOR IN LABORATORY ANIMALS

ADVANCING COMPUTATIONAL ETHOLOGY

ANA FILIPA DOMINGUES GERÓS

DISSERTATION PRESENTED TO THE FACULTY OF ENGINEERING, UNIVERSITY OF PORTO, FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN ELECTRICAL AND COMPUTER ENGINEERING

This thesis was supervised by:

Prof. Dr. Paulo de Castro Aguiar

i3S – Instituto de Investigação e Inovação em Saúde; INEB – Instituto de Engenharia Biomédica; FMUP - Faculdade de Medicina da Universidade do Porto.

Prof. Dr. Jaime dos Santos Cardoso

FEUP – Faculdade de Engenharia da Universidade do Porto; INESC TEC – INESC Tecnologia e Ciência.

Dr. Fabrice de Chaumont

Institut Pasteur, Human Genetics and Cognitive Functions, Paris.

The work presented in this thesis was developed at:

Neuroengineering and Computational Neuroscience Group
i3S - Instituto de Investigação e Inovação em Saúde
INEB - Instituto Nacional de Engenharia Biomédica
Universidade do Porto, Porto, Portugal

The research described in this thesis was financially supported by:

Fundação para a Ciência e a Tecnologia (FCT) and Fundo Europeu de Desenvolvimento Regional (FEDER) through the COMPETE 2020 - Operacional Programme for Competitiveness and Internationalisation (POCI), Portugal 2020, for the PhD grant SFRH_BD_137385_2018 granted to A. Gerós, and Projects PTDC/CTM-NAN/3146/2014 (POCI-01-0145-FEDER-007274) and PTDC/EMD-EMD/31540/2017 (POCI-01-0145-FEDER-031540).

The important thing is not to stop questioning;
curiosity has its own reason for existing.

Albert Einstein

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Paulo. What a crazy journey, ah? And it was not just with the PhD, it started a long time ago, long before the official birth of the NCN. Despite that, it took me almost three years to stop treating you as "*você*", but mission accomplished! Thank you so much for having the patience to teach me something new every day: from neuro/nerd stuff to riding an electrical scooter, from working efficiently while listening to *Epic Powerful Battle Soundtrack*s to mastering the art of the arguments, from building a DIY computer to learning how to be the best business partner. More importantly, thank you for letting me think and screw up on my own (I know it was a hard job for you not to help me at every tiny task). And luckily, over the last seven years (yes, SEVEN YEARS!), the invisible "step" has disappeared: because better than having the best mentor during the working hours, is having a good friend at all hours of the day. Thank you.

I would like to extend my acknowledgments to Prof. Jaime Cardoso. It was really a pleasure to have you as my teacher and co-supervisor, whether for all the fruitful discussions and brainstorming meetings, or for everything I've learned from you over all these years. Also, I would like to thank Fabrice, for being always available to discuss and share his insights. Thank you for every motivational email and comment, it was a pleasure to share this path with you too. A special thanks to Ricardo, for having the patience to put up with my endless lists of questions and unlocking problems that I didn't even know existed. This is why I know you are and will always be a great teacher, whichever path you decide to choose.

Anything is possible when you have the best research group! And that's why I want to thank all the current and former members of the NCN group, not only for being always ready to

help and for the long-lasting meetings, but especially for all the retreats, quiz-nights and lunches, where good science and jokes are always the hot topics! Miguel, what would I do without you? Thank you for all the times you turned on and off (and on again) the workstation (it takes a lot of patience, I know!), for all the times I've caught pneumonia on the i3S terrace or inside the lab (when you were wearing a t-shirt), and for all the silly and happy moments over the last years. Zezas, my life has never been the same since I came home to work: I don't have my favorite desk partner to upset whenever the references go crazy (*Mendeley*, I know!), or help out whenever a parenthesis is missing in your code. Thank you for all the friendship (and memes!), and for everything else to come! (you won't get rid of me that easily!). A special thanks to Andreia, Domingos, and Sara for all your happiness during the group activities (scientific or not!) and for never letting the NCN spirit die!

When good researchers and *party-fiesta* spirits come together, you know you are surrounded by the best people. Cláudia, thank you for showing me that "*work hard, play harder*" is the best balance you can have in life, and if we can do it with a smile, no one will ever stop us. I learned a lot from you (especially outside i3S!), and I can't wait for what's yet to come in the future! A huge thanks to Patrícia, who never let go of my hand at my first i3S retreat, and since then my life has gotten warmer and warmer. Thanks for all the secret conversations (even the ones you don't quite remember!) and for always being around to give a little motivational push (or a toast!). A crazy thanks to Andreia, for all the incredibly well-organized events (go team!) and all the late dances until we twist our feet! A colorful thank you to Ritinha (with a lot of colors on it!), for all the walks on the riverside, *Catitas*, and parties, which helped me remember that I am surrounded by strong and independent people, who fight really hard to achieve their goals. If it's not too much to ask, can you give me a little of your strength? And a special thanks to Xico, Isa, Elsinha, Tiago, Sofia, and Cati, for all the lunches and dinners, serious and crazy conversations, energetic music and colorful confetti. Don't forget that life is better when you all are around, even if this means going to *Auditório*.

An enormous *spasiba* to Lior, who taught me that there is a world beyond our small rectangle, very different from what we are used to, but full of life and happiness. Thank you for sharing some of this happiness with me and for always being so excited to help with my work!

I would also like to express my acknowledgments to all the members of *PontoZeroZeroUm*, in particular to Prof. Marques, that, besides never having given up on me, continues to push me to want more, learn more, and question more. Thank you for filling my days with one of the things I love to do - teaching - and for showing me that there is always so much more beyond our knowledge. A special thanks to Vera, for the companionship and friendly conversations, and Miguel, who, besides *saving my ass* when things get too busy, has the patience to explain me over and over again why physics is sooooooo funny and easy!

A million thanks would not be enough to thank my forever and ever friends, without whom none of this would be possible: Filipa, for all our “failed” conversations that brighten my days since 1997; Tiago, whom always celebrates my achievements as if they were his, even though being miles and miles away; Lia, Pi, and Joana, to whom I turn every time I need honest advice, a friendly hug or just a perfect company; my *Milanese* friends, always ready to liven up the party with their big smiles and bigger hearts; and finally Teresa, with whom I cried when she left the institute, but with who I know will always be there for me.

A greased thank you to Luís Leitxão, who now, more than ever, has the burden of putting up with me every single day. Because I don’t know how to say no to anything, and Luís goes everywhere as long as there’s food. Thank you for always being there for me, even when the days are longer than 24 hours; thank you for believing in me every day, even when I hide the chocolates from you, and thank you for agreeing to all my craziness, even if it means walking 10 km in the rain. None of this would be possible without your support, your courage, or your honesty. And that is why it’s DAAL.

Por último, queria agradecer à minha família louca por me ter acompanhado em todas estas aventuras. À minha mãe, que é a mãe mais maluca que alguém pode ter, mas a que tem o maior coração do mundo. Ao meu pai (um santo!), por toda a motivação (invisível) e encomendas enviadas nos alforges. Ao meu irmão, por ser um exemplo de resiliência e sabedoria (e por, sem dares conta, a transmitires a quem te rodeia). À tia Ameixa e Nés, por estarem sempre por perto (às vezes longe da vista, mas sempre perto do coração). À avó, por todo o carinho embrulhado em meias de lã.

All of you, and each one of you in your own way, make my life happier and teach me every day to never stop:

Learning, Laughing, Loving, Living.

Abstract

Analyzing behavior is a gateway to understanding biological systems in normal and pathological conditions. Behavior exposes the workings of the organisms at the system level, organ, tissue, cellular, down to molecular level. Animal behavior analysis plays, therefore, a fundamental role not only in research (universities, pharmaceutical companies) but also in industry (animal welfare for food production). In the particular case of neuroscience research, the characterization of animal behavior is a central tool in the search for solutions for Parkinson's, Alzheimer's, autism, stroke rehabilitation, among others. Unfortunately, animal behavior is complex to analyze and often relies on human judgment for manual annotation and quantification. This brings subjectivity and low reproducibility, with severe costs to society. This is the reason why, over the last 50 years, the analysis of behavior has become increasingly quantitative. Conventional behavioral assays have been gradually replaced by quantitative methods that resort to video recordings, machine vision and machine learning techniques for automatic and objective behavioral analysis. However, the existing computerized video-analysis solutions still present important limitations: recordings in static or unenriched environments, which may compromise natural behavior; lack of three-dimensional analysis and precision, which limits the detection of more complex behaviors; and absence of beginning-to-end software applications without end-user programming, and easy to configure. These constraints challenge the creation of novel computational tools that can drive more reproducible and high-throughput behavioral experiments, and ultimately provide new insights into the study of neural circuits' functions and the computations underlying them.

This thesis tackles these challenges by developing computational solutions for the quantification and automatic classification of behavior in laboratory animals. The integration of depth-sensing and thermal cameras in laboratory contexts was explored as an alternative to conventional optical cameras. These technologies, working with infrared light, allow recording in dark conditions, without affecting animals’ natural behavior. Besides that, they are independent of animals’ coat color, improving segmentation and tracking performances. In particular, depth-sensing cameras allow a three-dimensional analysis, which is essential to accurately describe the complex animal behavioral patterns. When combined with machine vision and machine learning techniques, segmentation and tracking of animal’s centroid and body parts were successfully obtained, even for dynamic and enriched environments. Supervised machine learning techniques, in particular Support Vector Machines and Convolutional Neural Networks, were applied for automatically recognizing animal behavioral patterns. This methodology revealed important properties on how machine learning methods should be constructed and tuned to efficiently learn animal’s behavioral events. Namely, particular attention should be given to the integration of temporal information along with spatial representations when designing a model’s architecture. To effectively correlate behavioral expressions with environmental changes, a versatile closed-loop framework was developed to provide real-time recognition of the animal’s behavior and to deliver this information for feedback control of behavioral experiments. The trainable system for automatic and markerless tracking and classification of the animal’s behavior was integrated into this closed-loop control system to trigger any external hardware device, such as behavioral mazes or real-time drug delivery optogenetics modules.

Overall, this thesis provides novel, automated, multi-purpose algorithms and methods for a complete analysis and quantification of behavior in laboratory animals, integrated in user-friendly software implementations, contributing to faster and high-throughput investigation in the computational ethology field.

Resumo

A análise de comportamento é uma das estratégias para a compreensão de sistemas biológicos em condições fisiológicas e patológicas. O comportamento expõe o funcionamento dos organismos desde o nível sistémico, tecidular, celular, até ao nível molecular. A análise do comportamento animal desempenha, portanto, um papel fundamental não apenas em investigação (universidades, empresas farmacêuticas), mas também na indústria (em bem-estar animal na indústria alimentar). No caso particular da investigação em neurociência, a caracterização do comportamento animal é uma ferramenta central no estudo de doenças como o *Parkinson*, *Alzheimer*, autismo, em reabilitação após acidentes vasculares cerebrais, entre outras. Infelizmente, o comportamento animal é complexo de analisar e é muitas vezes dependente da intervenção humana para anotação e quantificação manual, o que acarreta problemas de subjetividade e baixa reprodutibilidade, com custos elevados para a sociedade. Esta é a razão pela qual, nos últimos 50 anos, a análise do comportamento tem evoluído no sentido de se tornar cada vez mais quantitativa. As experiências convencionais para análise de comportamento foram gradualmente substituídas por métodos quantitativos que recorrem a gravações de vídeo, técnicas de visão computacional e de aprendizagem automática para uma análise objetiva, sem intervenção direta humana. No entanto, as soluções computacionais existentes apresentam ainda importantes limitações: gravações em ambientes estáticos ou não enriquecidos, que comprometem o comportamento natural; ausência de análises no espaço tridimensional, que limitam a deteção de comportamentos mais complexos; ausência de aplicações computacionais que sejam completas, versáteis e fáceis de utilizar. Essas limitações desafiam a criação de novas ferramentas computacionais que potenciem experiências laboratoriais mais reprodutíveis e mais fáceis de implementar, e que possam trazer mais conhecimento no estudo das funções dos circuitos neuronais.

Esta tese aborda estes desafios através do desenvolvimento de ferramentas computacionais para a quantificação e classificação automática do comportamento em animais de laboratório. A integração em ambiente laboratorial de sensores de profundidade e câmaras térmicas foi explorada como uma alternativa a câmaras convencionais. Estas tecnologias, através da utilização de radiação infravermelha, permitem a gravação na ausência de radiação visível, sem afetar o comportamento natural dos animais. Além disso, ao não serem afetadas pela cor dos animais, estas câmaras permitem melhorar o desempenho das técnicas de segmentação e rastreamento. Em particular, as câmaras com sensores de profundidade permitem uma análise tridimensional, essencial para descrever com precisão padrões comportamentais mais complexos. Quando combinadas com técnicas de visão computacional e de aprendizagem automática, a segmentação e rastreamento do animal foram obtidos com sucesso, mesmo em ambientes dinâmicos e enriquecidos. Diferentes técnicas de aprendizagem automática, em particular *Support Vector Machines* e *Convolutional Neural Networks*, foram utilizadas para reconhecer automaticamente os padrões comportamentais dos animais. Esta metodologia revelou características importantes que devem ser tidas em conta aquando da construção e adaptação de métodos de aprendizagem automática. Mais especificamente, a integração de informação temporal na arquitetura dos modelos, juntamente com informação espacial, mostrou ser relevante no que diz respeito ao desempenho dos métodos. Para correlacionar eficazmente expressões comportamentais com alterações no ambiente envolvente, foi desenvolvido um sistema de controlo em malha fechada que permite o reconhecimento em tempo real do comportamento animal e fornece estas informações para controlo de experiências laboratoriais. Os métodos desenvolvidos para rastrear e classificar automaticamente comportamento animal foram integrados nesta plataforma de controlo em malha fechada. O sistema pode assim ser usado com o objetivo de ativar dispositivos externos, por exemplo em arenas de comportamento operante ou em módulos de administração controlada de fármacos.

No geral, esta tese explora algoritmos e métodos inovadores para a análise automática, quantitativa e multifuncional do comportamento de animais de laboratório, integrados em aplicações computacionais que se esperam contribuir para acelerar estudos experimentais na área de etologia computacional.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

1D	One-Dimension
2D	Two-Dimensions
3D	Three-Dimensions
BCE	Binary Cross-Entropy
CNN	Convolutional Neural Network
CNS	Central Nervous System
COM	Communication Port
CSV	Comma-Separated Values
ConvLSTM	Convolutional Long Short-Term Memory
DMM	Depth Motion Map
EPM	Elevated Plus Maze
FCN	Fully Convolutional Network
fps	Frames per second
GMM	Gaussian Mixture model

GPU	Graphics Processing Unit
GUI	Graphical User Interface
GUIDE	Graphical Interfaces Development Environment
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HPA	Hypothalamic-Pituitary-Adrenal
IDE	Integrated Development Environment
IR	Infrared
IRT	Infrared Thermography
IoU	Intersection-over-Union
KDE	Kernel Density Estimation
kNN	k-Nearest Neighbor
LCD	Landscape-Change detection
LED	Light-Emitting Diode
LPS	Lipopolysaccharide
LSTM	Long Short-Term Memory
MBST	Mean Body Surface Temperature
MLP	Multi-Layer Perceptron
NN	Neural Networks
OCR	Object-Contextual Representation

OF	Open-Field
PIT	Passive Integrated Transponders
PCA	Principal Component Analysis
PIT	Passive Integrated Transponder
RBF	Radial-Basis Function
ReLU	Rectified Linear Unit
RFID	Radiofrequency Identification
RNN	Recurrent Neural Networks
RPN	Region Proposal Network
ROI	Region-of-Interest
SAP	Stretch-Attend Posture
SDK	Software Development Kit
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
SD	Standard Deviation
ToF	Time-of-Flight
t-SNE	t-Distributed Stochastic Neighbor Embedding
USB	Universal Serial Bus
WKY	Wistar Kyoto
YOLO	You Only Look Once

Author's contributions

The research work developed throughout this thesis directly resulted in the following contributions:

First Author Publications in International Peer-Reviewed Journals:

1. Gerós, A., Cruz, R., de Chaumont, F., Cardoso, J. S., & Aguiar, P. Deep learning-based system for real-time behavior recognition and automated closed-loop control of behavioral mazes using depth sensing (under review).

2. Gerós, A., Magalhães, A., & Aguiar, P. (2020). Improved 3D tracking and automated classification of rodents' behavioral activity using depth-sensing cameras. Behavior research methods, 52(5), 2156-2167, doi: 10.3758/s13428-020-01381-9 (IF = 6.242; CI$^{\#}$ = 8).

3. Franco, N. H.*****, Gerós, A.*****, Oliveira, L., Olsson, I. A. S., & Aguiar, P. (2019). *ThermoLabAnimal* – A high-throughput analysis software for non-invasive thermal assessment of laboratory mice. Physiology & behavior, 207, 113-121, doi: 10.1016/j.physbeh.2019.05.004, ***equal contribution** (IF = 3.244; CI$^{\#}$ = 9).

Presentations in Scientific Meetings:

Oral communications

1. Gerós, A., Magalhães, A., & Aguiar, P. (2019). Integrated Solution for 3D Tracking and automatic Classification of Rodents' Behavioral Activity based on RGB-D sensing. XVI

Meeting of the Portuguese society for Neuroscience 2019, 30 May-1 Jun, Lisboa, Portugal

2. Gerós, A., Magalhães, A., & Aguiar, P. (2018). Automatic Classification of Rodents’ Behavioral Activity based on Depth Cameras. Measuring Behavior 2018 - 11th International Conference on Methods and Techniques in Behavioral Research, 6-8 June 2018, Manchester, UK.

Poster communications

1. Gerós, A., Cruz, R., de Chaumont, F., Cardoso, J. S., & Aguiar, P. (2021). Deep Learning System for Online Rodent Behavioral Recognition and Automated Control of Behavioral Mazes, XVII Meeting of the Portuguese Society for Neuroscience 2021, 1-3 December, Coimbra, Portugal

2. Gerós, A., Cruz, R., de Chaumont, F., Cardoso, J. S., & Aguiar, P. (2021). Deep Learning System for Online Rodent Behavioral Recognition and Automated Control of Behavioral Mazes, Champalimaud Research Symposium 2021 - Dialogues on Neural and Machine Intelligence, 13-15 October, Lisboa, Portugal

3. Gerós, A., Magalhães, A., & Aguiar, P. (2018). Tracking and Automatic Classification of Rodents’ Behavioral Activity based on Depth Cameras. Champalimaud Research Symposium 2018 - Quantitative approaches to behavior & Neural Systems, 23-26 October, Lisboa, Portugal

4. Geros, A., & Aguiar, P. (2017). Automatic Quantification of Laboratory Animal Behaviour using 3D Video Recording. XV Meeting of the Portuguese Society for Neuroscience 2017, 25-26 May, Braga, Portugal

Awards:

1. Best student presentation of the work: Gerós, A., Magalhães, A., & Aguiar, P. (2018). Automatic Classification of Rodents’ Behavioral Activity based on Depth Cameras. Measuring Behavior 2018 - 11th International Conference on Methods and Techniques in Behavioral Research, 6-8 June 2018, Manchester, UK.

Other contributions in related research topics during the course of the thesis:

Other Publications in International Peer-Reviewed Journals:

1. Blenkuš, U.; Gerós, A.; Aguiar, P.; Carpinteiro, C.; Olsson, I.A.S.; Franco, N.H. (2021) Non-Invasive Assessment of Mild Stress-Induced Hyperthermia by Infrared Thermography in Laboratory Mice. Animals. (accepted for publication in *Animals*).

2. Mateus, J., Lopes, C., Aroso, M., Costa, A., Geros, A., Meneses, J., . & Aguiar, P. (2021). Bidirectional flow of action potentials in axons drives activity dynamics in neuronal cultures. Journal of Neural Engineering, doi: 10.1088/1741-2552/ac41db.

3. Horta, R., Nascimento, R., Gerós, A., Aguiar, P., Silva, A., & Amarante, J. (2018). A novel system for assessing facial muscle movements: the facegram 3D. Surgical Innovation, 25(1), 90-92, doi: 10.1177/1553350617753227 (IF = 2.058; CI$^{\#}$ = 2).

Other Awards:

1. RESOLVE Toolbox: *bringing ideas to the health market* (2017). Award with financial support for “FACEGRAM: Facial Movement Analysis in Reconstructive Plastic Surgery” project.

RESOLVE (*Respostas Específicas para Superar Obstáculos que Limitam a Valorização Eficaz*) was funded by Norte 2020 Programme (NORTE-01-0246-FEDER-000018), designed to provide solutions and management tools to early-stage projects and spin-offs in Health Sciences, and to transform innovative knowledge into business ventures and value creation. The RESOLVE Toolbox was awarded to selected teams to financially support the validation of prototypes and proofs of concept, and to establish communication routes between researchers/entrepreneurs and end-users. The “FACEGRAM: Facial Movement Analysis in Reconstructive Plastic Surgery” project proposed a novel system (hardware+software) capable of quantitatively and objectively assessing facial muscle movements. The system automatically describes a set of morphological measurements (static and dynamic) to support reconstructive plastic surgery. The system uses state-of-the-art, low-cost depth-sensing cameras, together with advanced computer vision techniques, to perform detailed tri-dimensional characterization of facial movements.

Organization of scientific training courses:

1. “Introduction to Python and Machine Learning for the Biosciences” courses (editions: 2020 and 2021). i3S – Instituto de Investigação e Inovação em Saúde, Porto, Portugal

2. “Introduction to data analysis and image processing with MATLAB” courses (editions: 2017 and 2018). i3S – Instituto de Investigação e Inovação em Saúde, Porto, Portugal

# Google Scholar citations as of 3rd of January, 2022

CHAPTER 1

Motivation, Objectives, and Thesis Structure

1. Motivation

A fundamental question in basic neuroscience is how brain functions are affected by new environmental conditions and disorders, such as addictive behaviors, depression, and personality disorders (Baker, 2011; Nestler & Hyman, 2010). As the brain is responsible for creating memories, emotions, and behavioral patterns, by studying the responses to environmental changes, it is possible to ultimately understand complex central nervous system (CNS) processes (Dickinson et al., 2000; Hong et al., 2015). By definition, animal behavior is the bridge between the physiological (molecular and cellular) and the ecological. It plays a critical role in biological adaptations and defines what animals do to interact with, respond to and control their environment (Mench, 1998; Snowdon, 2017). Therefore, the integration of animal behavior assessment in the neurosciences can provide important outlines for theorizing cognitive, social, and other mechanisms, which are deeply studied in the neuroethology context. In fact, research experiments are increasingly turning to animal social behavioral studies as a framework to understand human social-related symptoms witnessed in neurological disorders, such as Parkinson’s, Alzheimer’s, and autism. Besides that, animal behavior analysis outcomes can have the potential to support the development of new rehabilitation protocols and therapeutics (Brooks & Dunnett, 2009; Kabra, Robie, Rivera-Alba, Branson, & Branson, 2013; Weissbrod et al., 2013). Finally, it is important to emphasize that, in parallel with the crucial role that animal behavior analysis plays in research, it is also becoming an important tool in the industry, particularly in food production. Commercial pressure for competitiveness and concerns about animal welfare are leading the food production industry to the use of systems for automatic analysis of animal behavior (Ahrendt, Gregersen, & Karstoft, 2011; Hong et al., 2015; Stavrakakis et al., 2015).

The complexity of cognitive processes that characterize animal behavior, and the panoply of tests available for assessing these behaviors, generate vast amounts of data that are typically scored manually. Indeed, quantitative, precise, and long-term measurements that are user-independent, replicable, and standardized are not possible using video analysis based on visual inspection (Anderson & Perona, 2014; Hong et al., 2015; Jhuang et al., 2010; Kabra et al., 2013; Weissbrod et al., 2013). Therefore, the automation of animal behavioral analysis triggered new methodologies and technologic alliances in behavioral studies, having obvious advantages such as repeatability and reliability, with lower labor costs (Ou-Yang, Tsai, Yen, & Lin, 2011). Several studies have already addressed the behavioral analysis challenge by applying automatic systems based on machine vision and machine learning techniques to automatically track and characterize the behavior of different animals, such as flies, fishes, rodents, and larvae (Bohnslav et al., 2021; Kabra et al., 2013; A. Mathis et al., 2018; Romero-Ferrero, Bergomi, Hinz, Heras, & de Polavieja, 2019; Wiltschko et al., 2015). The relevance of this field has even led to dedicated denomination: Computational Ethology.

Although the behavioral research field is continuously evolving and experiencing fast innovation thanks to advances in machine vision and machine learning, the leading experts are unanimous in stating that there are still many important unsolved challenges in automatic classification/quantification of behavior (Egnor & Branson, 2016; M. W. Mathis & Mathis, 2020; Robie, Seagraves, Egnor, & Branson, 2017; von Ziegler, Sturman, & Bohacek, 2021; Zilkha, Sofer, Beny, & Kimchi, 2016). Acknowledging that ethologically relevant information is essential to study behavior and that this behavioral information should be acquired while the animal is performing natural and unconstrained behaviors (Zilkha et al., 2016), robust tracking algorithms in naturalistic (enriched) environments must be further developed. Besides that, the robustness of trackers and action/behavior systems must be improved to allow generalization when changing setup configurations (backgrounds, lighting conditions, cameras, etc.) or objects' appearance (different animal strains). Currently proposed systems are specifically designed for a given environment and a given set of predefined actions. Conventional systems based on machine learning techniques rely on annotated labels for training the classifiers in automatic tracking or classifying animal behavior. Such annotations need to be manually obtained, and most systems using deep learning techniques need large annotated datasets to obtain high performance, which is unfeasible in laboratory experiments. Thus, innovative techniques or fine-tuned networks that can work with small training datasets must be designed to increase the productivity and applicability of such computational tools. In addition, instead of relying on in-house training datasets created every time an automatic analysis needs to be performed, setting up benchmark datasets of animal behavior that can be explored and curated by the community is crucial to push the neuroscience field toward more in-depth studies. To try to reduce the human factor and analyze behavioral information from a data perspective, improved systems that do not rely on annotated datasets (unsupervised learning), or that can learn from both unlabeled and labeled data (semi-supervised or self-supervised learning), need to be further developed. In this sense, such techniques may not be subject to miss behavioral patterns that often arise from genotyping. Animal pose estimation and behavior recognition can also be improved by including 3-dimensional (3D) analyses, which allow a more complete description of the animals' movement or behaviors to be obtained. If animal behavior analysis aims at revealing underlying neural or genetic mechanisms or correlating behavioral expressions with the surrounding environment, real-time feedback systems are crucial to understanding cause-effect phenomena of specific neural circuits or environmental interactions. In this sense, new software tools that can achieve low-latency or real-time closed-loop feedback based on animal movement or behavior have the potential to change the paradigm in high-throughput animal experiments. Finally, sophisticated tools can be only integrated into experimental research or industry environments if they allow for low-cost setup equipment, easy to install and to use by behavioral researchers. In addition, they should preferably be as versatile as possible, combining tracking, behavior classification, and real-time/low-latency closed-loop analyses in a single application.

Altogether, these points highlight the need to create an integrated approach for the automated measurement of animal behavior that can efficiently solve the challenges that still persist in the behavioral neuroscience field, and pave the way to more reproducible and high-throughput experiments.

2. Objectives

The advances in technology, mathematics, and engineering, allowing scientists to automatically measure and analyze animals’ behavior, have led to the advancement of the computational ethology research. Aligned with this field, the aim of this thesis concerns the development of novel computational tools for the automatic quantification of behavior in laboratory animals. Envisaging the advance of the computational ethology field, the adopted strategy focused on exploring the opportunities and long-term directions of research in this area, to create different tools that address the previously mentioned challenges in the automated analysis of animal behavior. In particular, four main specific objectives were addressed in this work:

- **Exploration of the potential of depth sensing and thermal cameras to extract valuable behavioral and welfare information**

Acknowledging the importance of analyzing animal behavior in non-invasive and stress-free conditions, depth and thermal cameras are promising alternatives to conventional methods. By using infrared technologies, videos can be recorded even in dark rooms, without disturbing animals' natural cycle. Besides that, and since 3D analysis is necessary to adequately and objectively describe the highly complex rodents' behavioral patterns, the use of depth cameras in 3D systems is a low-cost and high-performance strategy. Considering these advantages, depth and thermal cameras were chosen to acquire video sequences, which were then analyzed for the extraction of behavioral patterns of interest.

- **Improvement of single-animal segmentation and tracking using machine vision and machine learning techniques**

Segmentation of the animal's whole body and tracking of the animal's centroid and body parts are fundamental tasks for the analysis of animal movements. Several algorithms and corresponding advantages and disadvantages were discussed in the literature review and later implemented to address this challenge. In particular, background modeling and deep learning methods for foreground segmentation (i.e., whole-body segmentation of single animals) were explored, with emphasis on algorithms for dynamic and naturalistic environments.

- **Development of a trainable system for automatic classification of behavioral primitives and automatic behavioral phenotyping**

The challenge of automatically recognizing animal behavior using depth-sensing data was further explored, with a focus on supervised machine learning techniques. Traditional methods, such as Support Vector Machine (SVM), were initially introduced and applied in this context. Deep learning approaches, such as Convolutional Neural Networks (CNNs), were later investigated, eliminating the need for manual feature-engineering and human bias. Mechanisms for integrating temporal information hidden in contiguous frames in the video sequences were also explored to understand the impact of different temporal scales in the classifier's performance.

- **Integration of a machine learning-based algorithm for real-time behavior recognition with closed-loop control systems for operant mazes**

Creating an integrated framework that allows automatic recognition of animal's position/behavior, and sending such signals for feedback control of sensors and actuators in a behavioral maze, is the ultimate goal of a high-throughput, multi-purpose and robust tool in the neuroscience field. A novel software solution was therefore developed for automated, markerless, non-invasive, and real-time 3D tracking and behavior recognition, that is integrated into a control platform, providing interfaces to trigger external hardware devices based on behavioral detection.

3. Thesis Structure

This thesis is divided into six chapters, starting with a literature review on the thesis subject, followed by three chapters containing the main experimental work, and the final chapter comprising the concluding remarks and future directions.

Chapter 1 highlights the motivation that drove the development of this work, the four main objectives, and how this thesis is structured.

Chapter 2 provides a general introduction on the definition of animal behavior and why its study is so relevant either in research or in industry. Given the recent developments in machine vision and machine learning methods, the analysis of animal behavior has become increasingly quantitative, reducing the need for time-consuming and non-standardized annotations. Computational methods for analyzing behavioral patterns are also reviewed in this chapter, highlighting the potential of these recent developments in advancing behavioral analysis but also reinforcing the need for further reproducible, human-free, and robust systems in computational ethology.

Chapter 3 presents a computational system for thermal assessment of laboratory mice using infrared thermography technology. Infrared thermography is a non-invasive alternative to classical stressful methods (such as rectal or infrared thermometers), to measure body temperature changes as a way to understand animals' physiological health and well-being. Using machine vision techniques applied to thermal imaging, a dedicated computational tool was developed for automatic segmentation of animal's whole body and analysis of mean body temperature of regions-of-interest (ROIs), which was integrated with a graphical user interface (GUI) for increased usability and transferability. This chapter is based on the following published original article: "*Franco, N. H., Gerós, A., Oliveira, L., Olsson, I. A. S., & Aguiar, P. (2019). ThermoLabAnimal–A high-throughput analysis software for non-invasive thermal assessment of laboratory mice. Physiology & behavior, 207, 113-121.*".

Chapter 4 explores the combination of depth-sensing technology, machine vision, and machine learning techniques for automated analysis of animal behavior. Depth cameras work with infrared sensors to produce range images where each pixel contains information regarding camera distance to the objects in the field-of-view. The use of cameras with infrared technology in the laboratory context was further explored, bringing added benefits in terms of improved background-foreground contrast and naturalistic lighting conditions. Segmentation of animal's whole body was improved using such technology, and different algorithms to solve the dynamic background challenge were reviewed and tested. Automatic classification of behavior was reported for both 4- and 7-classes tasks and the proposed methodology was validated for behavioral phenotyping of Wistar Kyoto and Wistar rats. This chapter is based on the following published original article: "*Gerós, A., Magalhães, A., & Aguiar, P. (2020). Improved 3D tracking and automated classification of rodents' behavioral activity using depth-sensing cameras. Behavior research methods, 52(5), 2156-2167"*.

Chapter 5 investigates the potential of modern machine learning methods, in particular deep learning, in extracting information directly from raw range video sequences. The objectives of the previous experimental work were extended by training CNNs in automatically tracking animal movements and classifying animal behavior. Deep networks were designed to learn not only spatial but also temporal features, which proved to be essential in accurately recognizing 4 different classes of behavior. The deep learning-based algorithm was adapted for real-time recognition of animal's position and behavior, and integrated into a platform for closed-loop control of sensors and actuators in any behavioral experiment. This chapter is based on the following original article that has been recently submitted for publication: "*Gerós, A., Cruz, R., de Chaumont, F., Cardoso, J. S., & Aguiar, P. Deep learning-based system for real-time behavior recognition and automated closed-loop control of behavioral mazes*".

Chapter 6 finishes this thesis with the general conclusions, along with the directions for future research in the computational ethology field.

CHAPTER 2

Quantifying and Understanding Animal Behavior

1. Studying behavior to understand biological systems' internal mechanisms

Behavior, the macroscopic expression of neural activity, translates to any sequence of movements performed by the animal, and it is the object of study of a comprehensive field called ethology. Depending on the goals of each experiment, the definition of behavior may change, to encompass different patterns and different types of behavior. In this sense, the definition of behavior may relate to social behavior (in relationships with other individuals) or non-social behavior. The latter can still be categorized into instinctive behavior (e.g., the inheritable tendency of an organism to respond to environmental stimuli), or learning (such as habituation or imitation behaviors) (Egnor & Branson, 2016). Nevertheless, within any particular animal species, some particular behaviors are shared among all members and others are more specific to certain individuals, regarding the environmental location. Besides that, and since all life forms exhibit behavioral activity, some attributes are common to all species. The relational component of all behaviors defends that the relationship between the animal and the environment influences the behavior, and the specification of the context or location is crucial to fully explain it. Also, the dynamic character of all behaviors imposes its analysis using frameworks for time series. Finally, since it is characterized by high dimensionality and complexity, its study must be performed taking into account all variables and considering overlapping of sub-behaviors, relational or social behaviors. In fact, each simple movement, as a single behavior, can be split into even smaller behaviors – sub-behaviors, which can be specific to a particular behavior or shared among many of them (Figure 2.1) (Anderson & Perona, 2014; Egnor & Branson, 2016; A. Gomez-Marin, Paton, Kampff, Costa, & Mainen, 2014).

In the neuroscience research, the readout of many laboratory experiments using animal models is at the level of animal behavior. Traditionally, the researchers have two distinct ways of analyzing behavior: detailed manual descriptions, in which the observers provide comprehensive and exhaustive reports on each animal behavior during a period of time, or specific assays, intended to assess particular patterns of behavior. Although ensuring that the specific pattern of interest is fully recorded, in detail, manual annotations are time-consuming, bring strong subjectivity and errors, and are difficult to standardize across experiments and observers. On the other hand, behavioral tests are easier to implement and faster to screen; however, in some cases, may not measure the pattern of interest in detail, being a more comprehensive technique (Egnor & Branson, 2016; Hong et al., 2015; Kabra et al., 2013; Weissbrod et al., 2013).

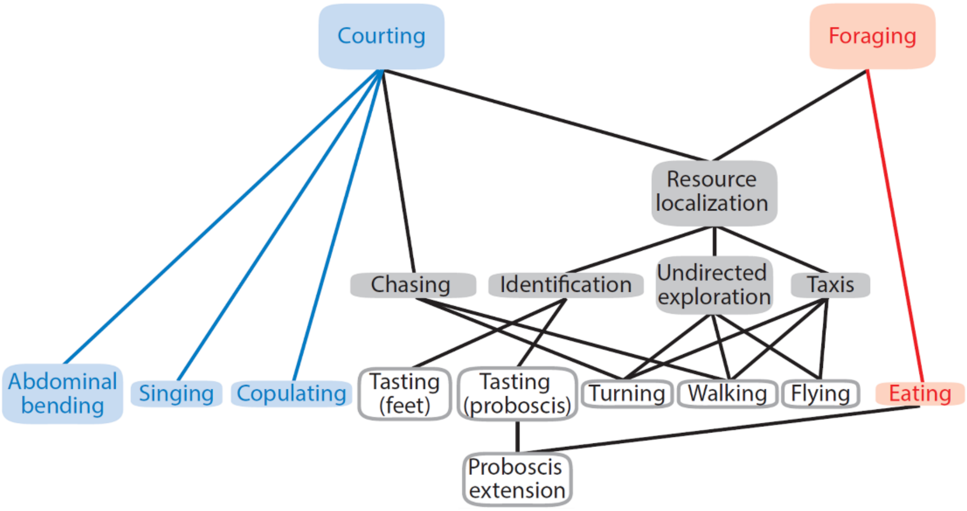


Figure 2.1 High-level specific behaviors and their sub-behaviors. Adapted with permission from Egnor and Branson (2016).

Nonetheless, in neuroscience research, behavioral assays are the most commonly used to study animal behavior, in particular, for motor analysis. Since it does not require expensive equipment and can be easily analyzed by visual inspection, the motor phenotype is the behavioral science function most frequently investigated (Brooks & Dunnett, 2009; Zörner et al., 2010). A variety of tests have been developed to describe the motor function (Figure 2.2), and they can be categorized according to the specific function intended to be evaluated, taking into consideration that some of them may be used in several different contexts. To assess locomotor activity, the open-field test is the most used apparatus in laboratory environments, and it involves placing the animal in a circular or square arena, and observing animal's movements, typically by visual inspection. Behavioral features like habituation (i.e., time the animal takes to explore the environment and venture out towards all regions) or moving times, and *rearing*, *grooming*, *freezing*, or defecation periods, are manually recorded to further assess, for example, locomotive impairment in animal models of neuromuscular disease or the efficacy of therapeutic drugs that may improve locomotion function (Brooks & Dunnett, 2009; Tatem et al., 2014).

With modern computational analysis methods, several approaches have been developed to observe and quantify the behavior of interest automatically, instead of relying on a visually behavioral assessment or time-consuming manual annotations. In fact, the research on ethology has benefited from advances in this field, allowing a collection and analysis of vast amounts of data and behavioral patterns that may go unnoticed to a human observer (subtle changes or modification at long time scales), the reduction in human bias and subjectivity, and standardization of measurements across labs (Egnor & Branson, 2016; Robie et al., 2017).

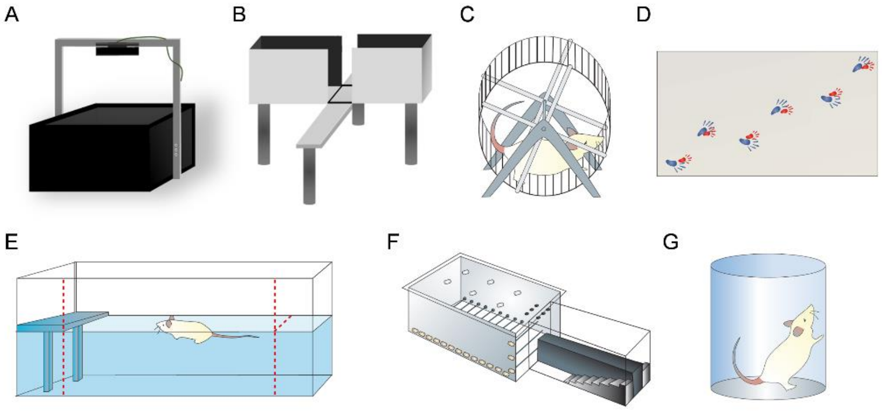


Figure 2.2 Behavioral tests commonly used to assess motor function in rodents. A. Open-field test. **B.** Elevated plus maze test. **C.** Running wheel. **D.** Footprint analysis. **E.** Swimming test. **F.** Staircase reaching test. **G.** Cylinder test. Images in C - D adapted with permission from Brooks and Dunnett (2009).

Importantly, automating the analysis of animal behavior allows for replacing or complementing human effort in two different ways: decision and observation. Computational methods appeared to support human decision-making, either through automatic methods for analyzing the animal's position or for automatic classification of individual or social behavior (offline or in real-time). This automated analysis can only be possible with automated video recordings for the observation of animals' behavior, and optimizations in the quality of video recording hardware have emerged hand in hand with advances in computer vision fields.

2. Automatic quantification of animal behavioral patterns using machine vision and machine learning approaches

The first automated methods that emerged used sensory devices, such as radiofrequency identification (RFID) transponders, for collecting spatial data of each individual animal at a given point in time (Macrì et al., 2015; Spink, Tegelenbosch, Buma, & Noldus, 2001; Tang & Sanford, 2005; Weissbrod et al., 2013). Recent methods still rely on this technique to analyze individual animal activity, arguing that these systems allow for a collection of longitudinal data without the need to remove the animals from their home-cage environments (Bains et al., 2016; Redfern et al., 2017). However, besides being an invasive technique where RFID microchips need to be implanted in animals' abdomen, they proved to be ineffective in analyzing more complex movements and behaviors.

To overcome these limitations, computerized video-analysis systems have emerged as potential non-invasive tools to assess animal behavior, combining two-dimensional (2D) video recordings with image analysis/processing techniques (Egnor & Branson, 2016; Robie et al., 2017). Pioneering studies and available commercial software share a methodology with three major steps to automate the recognition of animal behavior itself (Dell et al., 2014). First, video-based segmentation algorithms are initially applied to recorded videos for detecting the animal in each frame. Second, the whole-body and body parts positions are estimated across time, for a complete pose estimation and tracking. Finally, these pose estimates can be used for the automatic classification of actions and translate them into behaviors of interest.

Each of these steps will be separately addressed in the following sections, and they will be categorized based on algorithms' computational nature: machine vision-based or machine learning-based algorithms. Machine vision concerns the classical methods where rule-based algorithms are implicitly programmed for enabling machines to extract relevant information from images/videos. On the other hand, when using machine learning-based algorithms (Figure 2.3), the machine learns relevant information (patterns) within data representations (features), without being explicitly programmed (Bishop, 2006). Traditional machine learning methods follow the conventional paradigm of pattern recognition (Figure 2.3A and B). First, complex handcrafted features need to be carefully computed, transforming raw input data into a suitable internal representation. These features will be then fed to models during the learning process to detect or classify patterns in the input. In real-world scenarios, choosing which features are most important or relevant for learning a specific task at hand is a challenge itself, since the choice is highly problem- and user-dependent, and the power of these features significantly affects the performance of learned models. In this sense, instead of relying on handcrafted feature extractors, when using deep learning techniques (Figure 2.3C) the features themselves are automatically learned as part of the training step, in a process called representation learning. Deep learning architectures are obtained by composing simple non-linear modules (most of them subject to learning) that iteratively transform the representation at one level into a representation at a higher, more abstract level. By stacking multiple modules, and creating multilayer Neural Networks (NN), higher layers of representation can learn complex functions for the discrimination of relevant features (Bishop, 2006; Y. LeCun, Bengio, & Hinton, 2015).

Given the increased availability of large amounts of data and hardware computing power, deep learning methods have recently brought a breakthrough in artificial intelligence research. In particular, for image/video recognition, Convolutional Neural Networks (CNNs) are a dominant approach for many recognition and detection tasks and have been demonstrated to approach human performance. Similar to multilayer NN architecture, CNNs are comprised of multiple interconnected layers, in which each layer contains simple neurons that are connected to neurons in the next layer. However, it’s the way these layers work and interact with each other that makes CNNs suitable for image processing problems. The key feature of CNNs architecture is the convolutional layer, where convolutional operations with different kernels (learnable filters) are performed to encode spatial information between neighboring pixels of an image into feature maps. By composing several convolutional layers in a deep architecture, higher-level features are obtained by composing lower-level ones, in a way that edges, motifs, and objects can be iteratively learned throughout the network. Finally, in the last layer, a non-linear combination of the features extracted in the previous convolutional layers will be learned for the final classification, depending on the task at hand (Y. LeCun et al., 2015; Yann LeCun, Kavukcuoglu, & Farabet, 2010; Voulodimos, Doulamis, Doulamis, & Protopapadakis, 2018). Further theoretical concepts on NN and CNNs are available in Appendix - Fundamentals of Neural Networks and Convolutional Neural Networks.

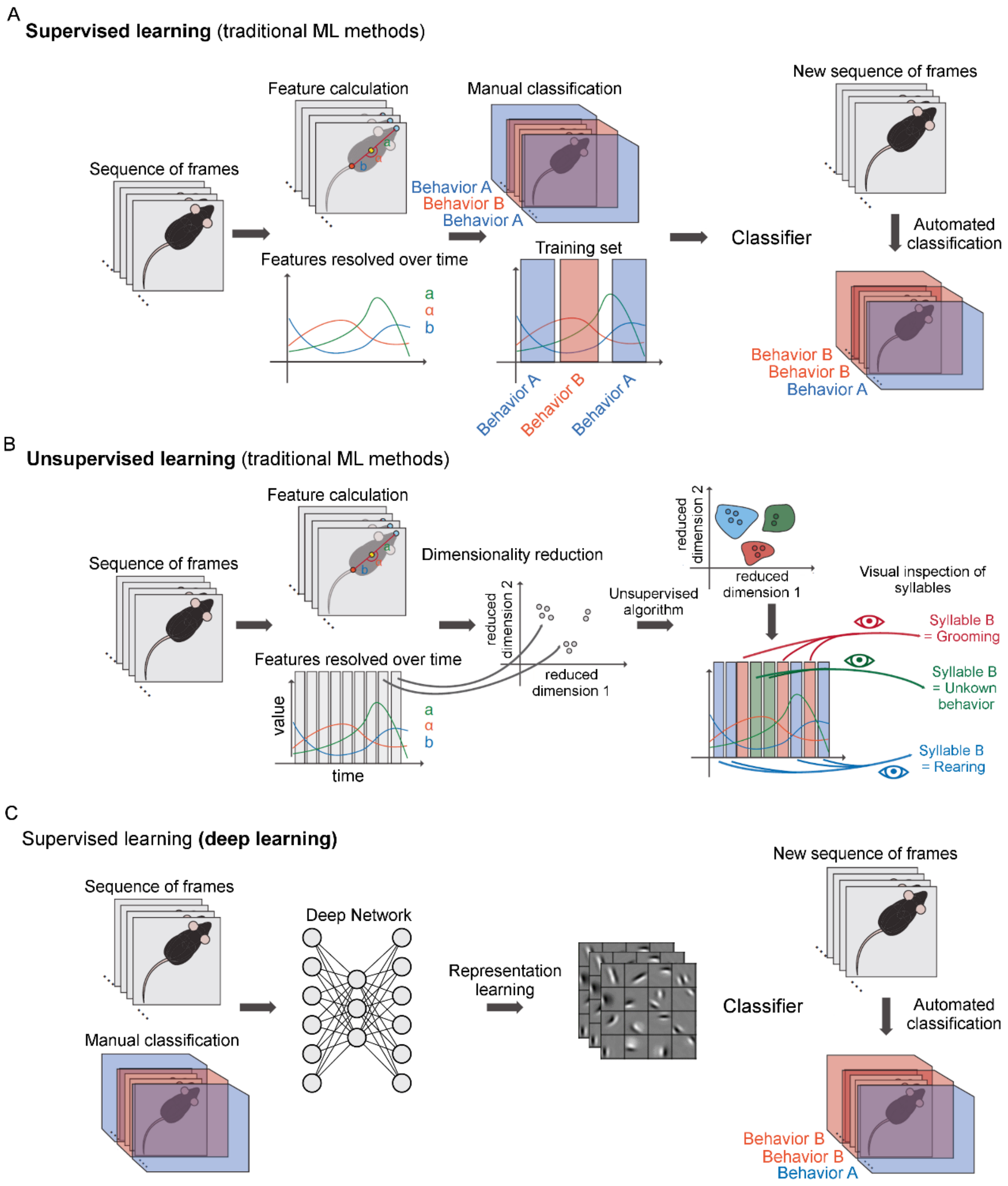


Figure 2.3 Categories of machine learning (ML) approaches used in behavioral research. Traditional ML methods are divided into: A. Supervised learning, where manually labeled data are required to first train the classifiers, and then to automatically recognize new videos. **B.** Unsupervised learning, where categories with similar features are automatically detected without manual annotations. Modern supervised ML methods rely on **C.** Deep learning, where the features themselves are automatically learned during the representation learning step. Image adapted with permission from von Ziegler et al. (2021).

Taking into account the availability of manual annotations and the intrinsic properties of the data, machine learning methods can be divided into two broad categories: supervised and unsupervised learning methods. In supervised approaches, manually labeled data (visually identified patterns) are required to train the classifiers (Figure 2.3A). Unsupervised methods rely on the selection of a feature representation of the raw data, without the need for manual annotations, and automatically detect any stereotyped patterns (e.g., automatically defines what it means for two behaviors to be similar, regarding available data) (Figure 2.3B). Some advantages come from these unsupervised properties, such as decreased subjectivity, the chance of finding different or rare behaviors, which may go unnoticed by human observers, and increased throughput and repeatability.

2.1 Automating video-based tracking in computational ethology

The animal tracking challenge is solved in most behavioral studies in a way to analyze the position and pose (body parts geometrical configuration) of animals over time. Initial solutions for video-based tracking relied on manually identifying the position of the animal in each frame, and disadvantages such as time-consuming, low spatiotemporal resolution, and observer subjectivity promoted recent advances in automated tracking systems (Dell et al., 2014; Robie et al., 2017).

Depending on the goal of each behavioral experiment, animal motion can be analyzed using descriptors that vary in terms of complexity, going from coarse representations such as centroid or ellipse tracking to finer representations, as three-dimensional (3D) animal pose estimation (Figure 2.4) (Talmo D Pereira, Shaevitz, & Murthy, 2020). Accordingly, computational tools for extracting animal movement over time will be discussed in the next subsections, addressing tracking techniques for progressively more detailed descriptions.

2.1.1 Segmentation of the animals from the background

The animal whole-body tracking task is closely related to the foreground (object(s) of interest) segmentation challenge, aiming at the detection of moving objects within a video stream, without prior knowledge about these objects (Sobral & Vacavant, 2014; Xu, Dong, Zhang, & Xu, 2016). If it is possible to distinguish foreground and background pixels in each frame, then estimating the centroid of a single animal is simply averaging the location of all foreground pixels, for each frame over time (Figure 2.4A).

One of the most common techniques to perform foreground detection is the implementation of background modeling methods. These approaches have evolved during the past years but they share a common pipeline: 1) background initialization, where a background model is constructed using a fixed number of frames; 2) foreground detection, in which a comparison (usually, background subtraction) between the current frame and the background model is performed to detect foreground objects; and optionally, 3) background maintenance, updating the background model, which was learned at the initialization step using the analyzed frames. The various background modeling algorithms can be categorized regarding the nature of the model (basic, parametric, and non-parametric models) or the area of the image under study (pixel-based, region-based and hybrid methods). The basic models for background subtraction include creating a background model using the median of the pixels over all frames (Static Median/Average methods) or just subtracting a manually-acquired static image of the background to each new recorded image (Static Frame Difference method).

Interestingly, the gold standard for animal segmentation still relies on such machine vision-based methods (Paulo Aguiar, Mendonça, & Galhardo, 2007; Alex Gomez-Marin, Partoune, Stephens, & Louis, 2012; Jhuang et al., 2010; Kabra et al., 2013; Noldus_Information_Technology_BV; Ohayon, Avni, Taylor, Perona, & Egnor, 2013; Pérez-Escudero, Vicente-Page, Hinz, Arganda, & De Polavieja, 2014; Rodriguez et al., 2018; Spink et al., 2001; Sridhar, Roche, & Gingins, 2019; TSEsystems; van Dam et al., 2013). Background subtraction methods are performed to segment the animal using separation by intensity thresholding, which is highly dependent on the environment's conditions (high contrast or uniform background are needed to ensure a good performance).

A potential methodology to address the laboratory animal tracking challenge in naturalistic (enriched) environments should be robust to lighting changes, objects overlapping and moving background elements. In fact, the tendency on animal models and behavior quantification is to continuously increase the complexity towards more natural environments. However, traditional segmentation methods are not capable of dealing with dynamic environments' situations and, therefore, have proved insufficient for the task of animal segmentation in enriched environments (Sobral & Vacavant, 2014; Xu et al., 2016).

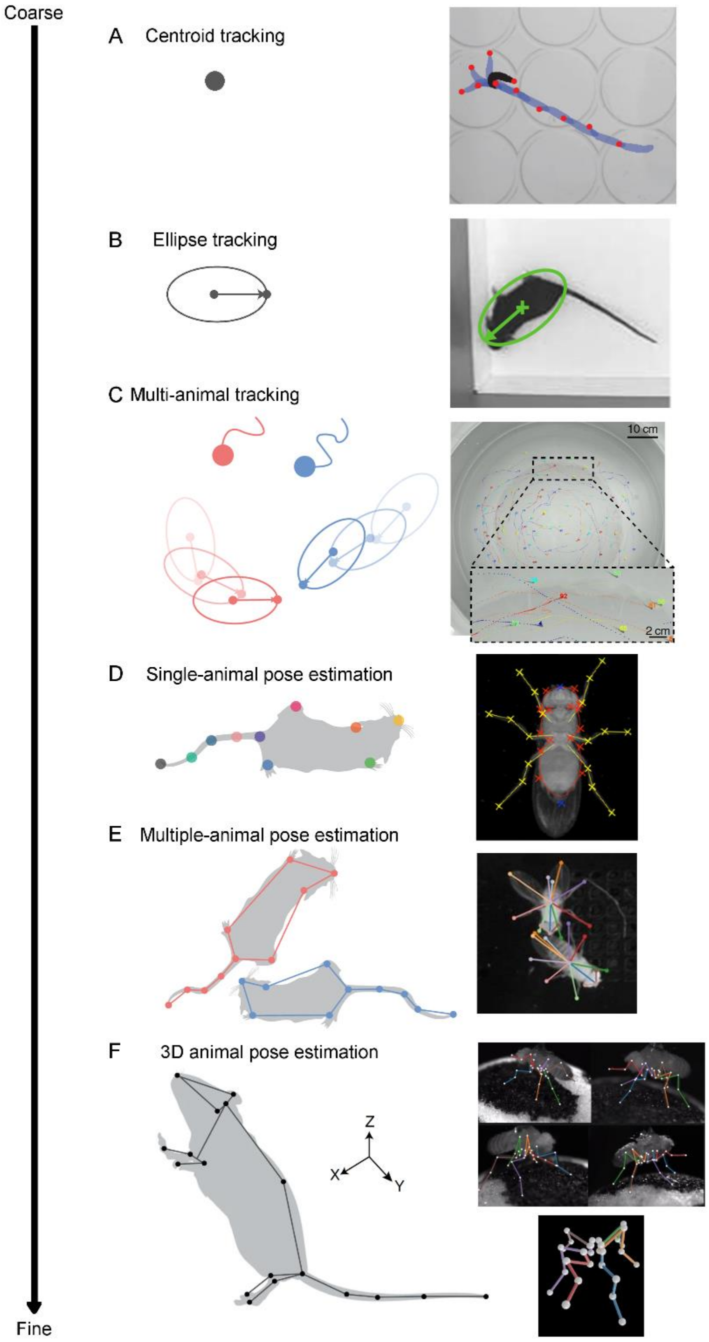

Coarse
A Centroid tracking
B Ellipse tracking
C Multi-animal tracking
10 cm
2 cm
D Single-animal pose estimation
E Multiple-animal pose estimation
F 3D animal pose estimation
Z
X
Y
Fine

Figure 2.4 Tracking representations ranging from a single-point tracking (coarse tracking representation) to full three-dimensional (3D) pose (fine tracking representation). A. Tracking of single animal's centroid. **B.** Single animal's tracking by fitting an ellipse to animal's shape. **C.** Multi-animals' tracking, with identity assignment over time. **D.** Single-animal pose estimation, with detection of multiple body parts. **E.** Multi-animal pose estimation and tracking, by detecting multiple body parts of different animal over time.

F. 3D pose estimation, with detection of multiple body parts in the three-coordinates system. Image representation adapted with permission from Talmo D Pereira, Shaevitz, et al. (2020). Images adapted with permission from: A. Alex Gomez-Marin et al. (2012), B. Geuther et al. (2019), C. Romero-Ferrero et al. (2019), D. Uhlmann, Ramdya, Delgado-Gonzalo, Benton, and Unser (2017), E. Talmo D Pereira, Tabris, et al. (2020), F. Karashchuk et al. (2021).

In this sense, a dynamic background landscape estimator algorithm is a solution for whole-body segmentation in changing environments. One of the pioneering methods described in the literature to deal with the dynamic background challenge (Sobral & Vacavant, 2014) is based on a parametric probabilistic background model proposed by Stauffer and Grimson - Gaussian Mixture Model (GMM) (Stauffer & Grimson, 1999). This machine learning-based method for unsupervised clustering consists of representing each pixel as a sum of weighted Gaussian distributions defined in a given color space. This technique determines which intensities are most probably belonging to the background and the remaining pixels are associated with the foreground. The distributions are updated using an online Expectation-Minimization algorithm. Particularly, each pixel is modeled by a mixture of K Gaussian distributions (usually, K = 2 or 3):

$$\mathrm{P}(\mathrm{x_t}) = \sum_{\mathrm{i}=1}^{\mathrm{K}} \omega_{\mathrm{i,t}}\, \mathrm{N}\,(\mathrm{x_t}|\mu_{\mathrm{i,t}}, \Sigma_{\mathrm{i,t}}), \tag{2.1}$$

with $\mathrm{N}\,(x_{\mathrm{t}}|\mu_{\mathrm{i,t}}, \Sigma_{\mathrm{i,t}}) = \frac{1}{(2\,\pi)^{\frac{\mathrm{D}}{2}}} \frac{1}{|\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(x_{\mathrm{t}} - \mu_{\mathrm{i,t}}\right)^{\mathrm{T}} \Sigma_{\mathrm{i,t}}^{-1} \left(x_{\mathrm{t}} - \mu_{\mathrm{i,t}}\right)\right)$, D is the dimension of the color space, $\omega_{\mathrm{i,t}}$ is an estimate of the weight and each Gaussian is described by its mean $\mu_{\mathrm{i,t}}$ and covariance matrix $\Sigma_{\mathrm{i,t}}$, and |·| denotes the matrix determinant. The Gaussian distributions that may correspond to background colors are determined based on the persistence and the variance of each one, and the pixel values that do not fit the background distributions are considered foreground until there is a Gaussian that includes them with consistent evidence. To improve the performance of this method in conditions of rapid variations of illumination, noisy videos, and shadows, many authors have proposed other alternatives, improving the original method. A detailed description of other GMM-based methods is presented in Sobral and Vacavant (2014); (Xu et al., 2016).

To solve the drawbacks of manually selecting the parameters in each environment, non-parametric methods can be used to perform background segmentation. The Kernel Density Estimation (KDE), a widely used non-parametric and region-based model, was introduced by Elgammal, Harwood, and Davis (2000). These region-based methods take advantage of inter-pixel relations to segment the images into regions and identify foreground objects from image regions. In order to model the background distribution in KDE methods, the probability density function that a particular pixel will have intensity value x_{t} at time t can be estimated using the kernel estimator (Elgammal et al., 2000):

$$\mathrm{P}(x_t) = \frac{1}{N}\sum_{i=1}^{N} K(x_t - x_i) \tag{2.2}$$

If $\mathrm{P}(x_{\mathrm{i}}) < \mathrm{T}$, then x_{i} will be classified as foreground pixel, where T is a global threshold over all the images. The main advantages of this method are its ability to deal with multimodal backgrounds (with fast changes), and to avoid parameter estimation steps. However, since KDE needs to save, in memory, N frames for the foreground detection process, this method can be computationally expensive.

Modern approaches have begun to use instead deep learning methods to solve animal segmentation challenge in more complex backgrounds, using encoder-decoder or adapted Mask-RCNN architectures for semantic segmentation of animal's body (Francisco, Nührenberg, & Jordan, 2020; Geuther et al., 2019; M. Marks et al., 2020). However, little progress has been made to solve this problem, and efforts have been directed towards methods that calculate the position of the animal and its body parts, skipping the background segmentation step.

2.1.2 Tracking multiple animals

One computer vision challenge that has been intensively explored is the tracking of multiple interacting and indistinguishable animals in a given environment (Figure 2.4C). This is particularly relevant because animals are social in the presence of other conspecifics and exhibit specialized social behaviors, most of them different from single-animal behaviors. Monitoring the behavior of multiple individuals is, however, computationally hard since some animals move particularly rapidly when interacting, and these interactions, as they involve physical contact, are prone to occlusions. The pipeline shared by pioneering solutions for the multiple-animal challenge can be divided into two sub-problems: identification of the position/pose of all animals in each frame (multi-animal detection problem), and connection of detected positions across frames into trajectories for each animal (identity problem).

A common methodology to detect multiple identity-less positions consists in, firstly, segmenting the foreground and, then, clustering the pixels into spatially connected groups using clustering algorithms, such as watershed segmentation (Fiaschi et al., 2014; Giancardo et al., 2013) or Expectation-Maximization algorithm for GMMs (Ohayon et al., 2013; Pérez-Escudero et al., 2014). Other methods take into account the similarity of size and shape of the organisms to perform multi-target detection (Branson, Robie, Bender, Perona, & Dickinson, 2009; CleverSys_Inc.; de Chaumont et al., 2012; Noldus_Information_Technology_BV; TSEsystems). However, both approaches alone do not allow detecting multiple organisms in cases of occlusion, since the shape and size information from foreground detections are not enough to individualize the animals (Robie et al., 2017).

Some solutions were found for tracking during occlusions, and one of them is to complement the previous approaches with prior knowledge about the shape of the organisms and appearance of the foreground pixels (de Chaumont et al., 2012; Dell et al., 2014; Kabra et al., 2013; Pérez-Escudero et al., 2014; Robie et al., 2017). However, these approaches are computationally slow, especially for multi-tracking with more than 2 animals. Another alternative is to use temporal context or kinematic information to solve the occlusion problem. Thus, it is possible to infer where the animals' pixels are in the current frame taking into account information of both past and future frames (Fiaschi et al., 2014; Kabra et al., 2013). These techniques are also computationally expensive when the future temporal context is taken into consideration, and some alternatives were suggested to improve the efficiency of these methods by reducing the size of the problem (Lenz, Geiger, & Urtasun, 2015).

To solve the second problem of maintaining the identity of multiple animals across frames, one strategy is to use, once again, temporal context and kinematic information of previous and future frames (assuming animals move short distances at a constant speed, for example) (de Chaumont et al., 2012; Gershow et al., 2012; Matsumoto et al., 2013; Rodriguez et al., 2018). Alternatively, visual differences/fingerprints between individuals can be detected using either machine vision-based (Pérez-Escudero et al., 2014), or machine learning-based methods. The latter can learn models of each animal's appearance from training frames and, then, use these learned models to predict the identity of each detected animal in a particular frame (Robie et al., 2017; Sridhar et al., 2019). Recently, deep learning techniques have led the way to improve tracking of large groups of unmarked animals, using state-of-the-art CNN architectures for animals' detection and identification over time (Arac, Zhao, Dobkin, Carmichael, & Golshani, 2019; Z. Chen et al., 2020; Francisco et al., 2020; Romero-Ferrero et al., 2019).

Naturally, the task of tracking animal identities can be facilitated using surface markers. In fact, biologists have long used dye, bleach, or even different kinds of shavings to mark animals for identification (Stonehouse, 1978). Such markings are still currently used to more easily assign pixels to organisms identities, either in academic research (Boenisch et al., 2018; Crall, Gravish, Mountcastle, & Combes, 2015; Hong et al., 2015; Ohayon et al., 2013) and commercial software (CleverSys_Inc.; Noldus_Information_Technology_BV; TSEsystems). Even so, these markings can be a technical limitation since marker imposition can change animals' natural behavior (de Chaumont et al., 2012).

2.1.3 From tracking information to pose estimation

Trajectory data from single or multiple animals, represented by the animal's center of mass over time, can be useful to answer some specific scientific questions. However, to allow the classification of subtler and more complex behaviors, particle-like trajectory information may not be enough (Berman, Choi, Bialek, & Shaevitz, 2014; Hong et al., 2015; Wiltschko et al., 2015). In fact, over the last years, proposed methods for analyzing the movement of animals have evolved towards capturing progressively more detailed descriptions of the geometrical configuration of multiple body parts, to estimate a complete pose of the animal (Figure 2.4D, E).

Most of the published work and available software rely on 3 different approaches to track animal body parts: simple morphological techniques, physical model-based, and machine learning-based methods. The first approach implements basic techniques by combining morphological operations and geometric considerations on skeleton detections to calculate a small number of body parts. This methodology assumes that the end-points of the skeleton can be used as a proxy for animal's head, nose, and/or tail positions (Ben-Shaul, 2017; Alex Gomez-Marin et al., 2012; Tong et al., 2020; Unger et al., 2017; Z. Wang, Mirbozorgi, & Ghovanloo, 2018).

Physical model-based methods can also be used to identify the position of individual body parts, and the existing methodologies can be divided according to their complexity. Simple shaped models, such as ellipses (Figure 2.4B), can be computed by fitting a shape contour to the organism's pixels (CleverSys_Inc.; Hong et al., 2015; Ohayon et al., 2013; TSEsystems), or by using encoder-decoder deep architectures to learn ellipse parameters (Geuther et al., 2019). The orientation of that shape provides information about a simple animal's pose. Physical model-based tracking algorithms, initially proposed by de Chaumont et al. (2012), and further extended to a 3D space (Matsumoto et al., 2013; Nakamura et al., 2016), can also be applied to obtain more complex ruled-based information on animal's shape. They are composed of a skeleton model using geometrical primitives, linked by physical constraints. The relative position (and orientation) of key anatomical structures of the animal (nose, head, belly, and tail) can be estimated by fitting the animal model into the animal's pixels, in a frame-based approach. Other complex fittings can be achieved, using, for example, an active shape models' approach, that combines deformable models of shape and local gray-level appearance, to provide a compact description of the shape of the animal and identification of several body parts (Thanos, Restif, O'Rourke, Lam, & Metaxas, 2017; Twining, Taylor, & Courtney, 2001; Uhlmann et al., 2017). Although fast and with good performance, they require sophisticated and rigid skeleton models that are difficult to construct and fit to animals' body and limit flexibility and applicability beyond species-specific experiments.

In contrast to machine vision methods, modern machine learning methods and, in particular, deep learning, can be used for directly learning and computing animal's poses. Unlike previous approaches, that explicitly model animal's body, state-of-the-art CNNs learn associations between image patterns and pose parameters directly from the images. Body-parts/landmarks positions are represented as confidence maps, which encode the location of each landmark as a density function of a 2D probabilistic distribution centered on the ground-truth image coordinates (the brightest pixel of the confidence map is at the location of the landmark). These confidence maps are learned by CNNs using raw input images and corresponding ground truth landmarks' coordinates, and, after training, the body-parts' coordinates on unlabeled images can be decoded from the predicted confidence maps via peak detection techniques. Using this methodology, state-of-the-art deep learning-based approaches to human pose estimation were firstly adapted for animal pose estimation by A. Mathis et al. (2018) and T. D. Pereira et al. (2019), which lead the way through the creation of several computational tools with human-level accuracy performance. One of the most important challenges in computational neuroscience is designing robust and generalizable deep learning networks with little training data, because, in contrast to human benchmark datasets, manually annotated datasets for animal experiments are smaller and sparse (M. W. Mathis & Mathis, 2020). These pioneering studies rely on two distinct approaches to obtaining high performances with little annotated data. The first approach, transfer learning, allows reducing the need for large datasets by reusing pre-trained network parameters (previously learned on a broader set of natural images, typically ImageNet dataset (Deng et al., 2009)). Ideally, the isolated learning paradigm can be overcome, to reduce the need for intensive learning on the dataset under study, just by transferring knowledge learned from one task to solve related ones. This approach has proven to be effective in animal pose estimation with reduced training sets (100-200 training images) (Arac et al., 2019; Francisco et al., 2020; Günel et al., 2019; Karashchuk et al., 2021; A. Mathis et al., 2018).

The second approach consists in designing efficient NN, where CNNs contain fewer parameters to tune, being faster to train and predict (Z. Chen et al., 2020; Ebbesen & Froemke, 2020; Graving et al., 2019; T. D. Pereira et al., 2019; Talmo D Pereira, Tabris, et al., 2020). In theory, imaging conditions in animal experiments do not suffer from significant variability, and, for that reason, high networks' representational capacity is not necessary. Although more efficient than transfer learning approaches, low-weight networks' architectures may not be well suited for changing environmental conditions or naturalistic environments.

Some recent progress has been achieved towards further reducing the need for annotated data and this is currently an active area of research. Semi-supervised learning and domain adaptation techniques have been explored (Li et al., 2020; Suwajanakorn, Snavely, Tompson, & Norouzi, 2018; Lauer et al., 2020), as well as incorporating temporal information for a precise location of landmarks without the need for additional annotations (X. Liu et al., 2020; Wu et al., 2020).

2.2 Automating animal tracking in three dimensions

Recent studies have tried to address the animal's tracking challenge by applying automatic systems, with focus on machine learning-based techniques. However, constraints such as manual interventions, single-animal detection, complex setup, and output of few informative data make them ineffective and inappropriate in animal behavior research experiments. Besides that, some animals' movements/behaviors are extremely complex (e.g., *grooming*, *rearing*), and can only be adequately and objectively described using a 3D spatiotemporal analysis (i.e., a combination of space and time). Consequently, the precise estimation of animals' poses in these 2D video-analysis systems is very limited, impairing detailed pose characterization.

In order to perform 3D tracking of laboratory animals using video-based methods (Figure 2.4F), range images are now one of the best options. These images, also referred to as depth images or depth maps, correspond to frames whose pixels express the distance between a known reference frame and a visible and specific point in the scene. In fact, the use of methods based on the imaging range concept has steadily increased, as well as the development and production of dedicated sensors' hardware (Cantzler, 1997). Early researches, aiming at animal's tracking in 3D, combine machine vision-based methods with multiple cameras to produce range images (Ardekani et al., 2013; Attanasi et al., 2013; Cachat et al., 2011; Straw, Branson, Neumann, & Dickinson, 2010; Veeraraghavan, Srinivasan, Chellappa, Baird, & Lamont, 2006). They share a standard three-step methodology (Figure 2.5A): 2D tracking or pose estimation, triangulation, and post-processing. Triangulation is used to determine the depth information of points in the scene, by collectively capturing all regions-of-interest (ROIs) across different viewpoints. To improve 3D reconstruction (eliminate false detections or resolve inconsistencies between different views), post-processing techniques are usually applied (Attanasi et al., 2013; Ebbesen & Froemke, 2020; Günel et al., 2019; Karashchuk et al., 2021). One-time camera calibration precedes these steps, where an object with distinct features or patterns is used to compute camera calibration parameters that allow mapping 2D points into 3D world coordinates. More recent studies still apply this multiple-views' approach to reconstruct and track animals' movements and pose, using machine learning-based methods (in particular, deep learning). Deep NN are trained to learn 2D confidence maps from multiple 2D views, which are later used for extracting 3D pose by triangulation (Arac et al., 2019; Francisco et al., 2020; Günel et al., 2019; Karashchuk et al., 2021). With the recent advances in the deep learning field, an alternative approach has been explored that allows directly extracting and learning the 3D representation of the pose to combine information from different views, without the need for the triangulation step (Figure 2.5B) (Dunn et al., 2021; Zimmermann, Schneider, Alyahyay, Brox, & Diester, 2020).

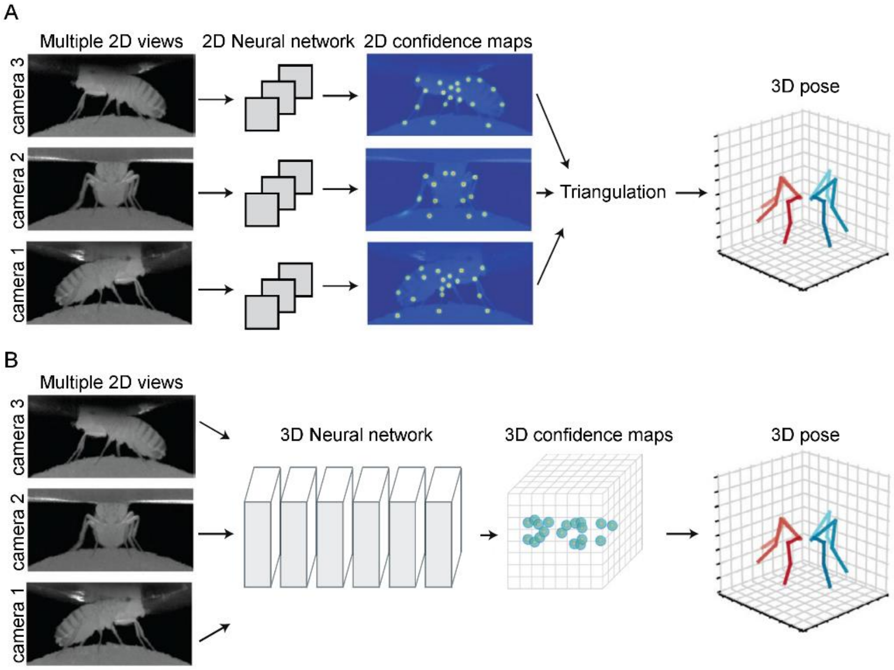


Figure 2.5 Deep learning approaches for 3D pose detection using multiple cameras: A. 2D tracking of landmarks is obtained using a 2D pose-detection network for each camera view, to create 2D confidence maps which are then triangulated to obtained the final 3D pose. **B.** Multi-camera views are processed by a 3D network to directly predict 3D landmark positions. Images adapted with permission from Dunn et al. (2021); Talmo D Pereira, Shaevitz, et al. (2020).

When using multiple cameras, apart from requiring calibration, expensive setups, or additional hardware to synchronize different equipment, the correspondence problem between different images must be solved, and depth determination depends on surface features (Cantzler, 1997). Furthermore, while only two cameras are required for triangulation, additional cameras are often used to increase the precision and quality of point clouds' reconstruction and to avoid occlusions (Dell et al., 2014). To overcome the limitations of the multiple cameras' scenario, some technologies have been created that allow the acquisition of range images from a single imaging device (Dell et al., 2014). Range cameras, concerning the sensor device that is used to produce this type of images, may have different operating modes, for example, stereo triangulation, structured light, and time-of-flight (ToF). If the sensing system combines RGB color information with per-pixel depth information, it is also called RGB-D camera. The most promising developed equipment, providing a good trade-off between performance and cost, use structured light or ToF operating modes to acquire depth maps. A detailed description of the latest researches in animal tracking and behavior using these technologies can be found in Section 3 in the current chapter.

2.3 Automating animal behavior recognition

The information about animals' or body parts' trajectories can be used to analyze position-based interactions, such as the amount of time animals spend at different locations, or near one another. And although there has been a surge in machine learning tools for segmentation, identification, and pose estimation, a key element for the computational ethology field is the recognition of behavior itself, knowing what were the animals doing at a specific time point, and the social interactions between multiple animals (Figure 2.6A). The task of categorizing behavior at a particular time point into distinct classes is called behavior analysis or classification (Egnor & Branson, 2016; Robie et al., 2017). Automatic behavior analysis seeks, indeed, to overcome the limitations of manual approaches, and to create more efficient methods to classify animal behavior. In fact, unlike human annotators, automatic systems do not change their definition of behavior over time, and, in this sense, the automatic analysis allows the reduction of biases and the production of annotations that are more repeatable over time and also across different laboratories and experiments (Dell et al., 2014; Egnor & Branson, 2016; Robie et al., 2017).

2.3.1 From low-level representations to automatic behavior classification

One simple approach to automate behavior classification is to explore rule-based classifiers, in which a set of rules are manually defined to describe the presence of behavior in a given sequence (de Chaumont et al., 2012; Alex Gomez-Marin et al., 2012). Although simple, this type of classification has disadvantages regarding generalization between different populations and pathologies, it can only be used for simple behaviors, and it depends on the quality of the features (Egnor & Branson, 2016).

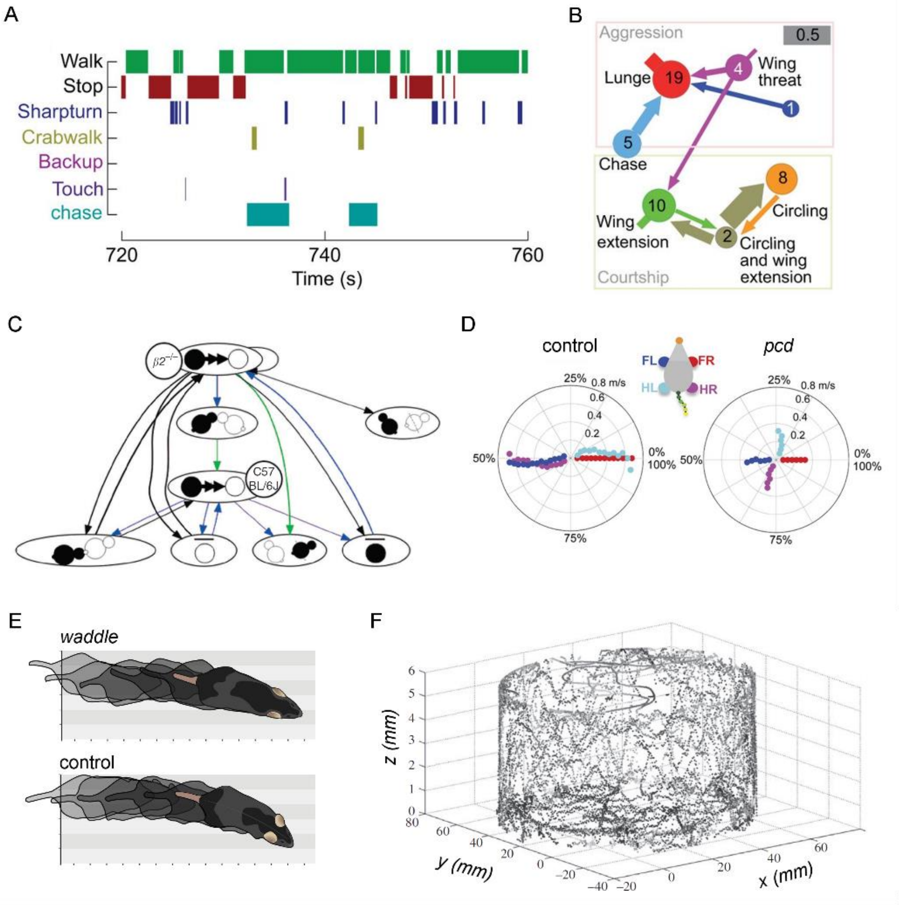


Figure 2.6 Examples of graphical representations of animal behavior with biological meaning. A. Ethogram representation: social behaviors of Drosophila Melanogaster as function of time. **B.** and **C.** Transitional behavioral graphs of different populations: **B.** transition probabilities of male Canton-S flies engaging in both aggressive (top) and courtship (bottom) behaviors. Circle diameters (scaled logarithmically) and numbers represent the average frequency of each action; **C.** cross-linked transitional graphs for the contact event between C57BL/6J and β2-/- mice. Colored arrows represent events that occur only in the first 4 minutes (blue) and the last 4 minutes (green). For both B. and C., the thickness of the arrows is proportional to the probability of the event transition. **D.** Polar plot representation: front-hind limb coordination represented as the phase of the step cycle in which each limp enters stance, aligned to stance onset of the front-right (FR) paw (red), for control (left) and Purkinje cell degeneration (pcd) (right) mice. Distance from the origin represents walking speed. FL: front-left; HR: hind-right; HL: hind-left. **E.** Stroboscopic representation: consecutive animal poses during a walking event for two different sub-behaviors/modules: waddle and control. **F.** 3D trajectory representation: tracking coordinates (x, y, and z) of a fly in millimeters (mm). Images adapted with permission from:

A. and B. Robie et al. (2017), C. de Chaumont et al. (2012), D. Machado, Darmohray, Fayad, Marques, and Carey (2015), E. Wiltschko et al. (2015), F. Ardekani et al. (2013).

More complex approaches can be applied for automatically classifying behavior using ML methods. In traditional machine learning methods, it is first necessary to represent the input video sequence into useful information that will be interpreted by the classification algorithm (Egnor & Branson, 2016; Robie et al., 2017; von Ziegler et al., 2021). In this sense, low-level representations of behavior can be extracted from the video sequences and they can be further divided into two categories: trajectory-based features and pixel-based features. Trajectory-based features are extracted from the tracked positions of the animals and/or their body parts over time. In this sense, trajectory-based features may carry useful and interpretable information of dynamic and continuous nature, which can be used for behavior state characterization. Published works employ this type of features to feed classifiers using different methodologies. From the trajectory, simple measurements can be extracted, such as position-, speed- and acceleration-based features, to identify some basic behaviors (*rest*, *walking*, *running*) (Jhuang et al., 2010; Nilsson et al., 2020; Segalin et al., 2020; van Dam et al., 2013; Weissbrod et al., 2013). If whole-body segmentation was performed using ellipse-fit tracking, for example, or simple model-based methods, some features extracted from the tracked shape can be derived (Hong et al., 2015; Kabra et al., 2013; van den Boom, Pavlidi, Wolf, Mooij, & Willuhn, 2017). More complex behaviors can also be identified using features extracted from multiple animals (Burgos-Artizzu, Dollár, Lin, Anderson, & Perona, 2012), multiple body parts (de Chaumont et al., 2012), or using motion features estimated by calculating the optical flow of adjacent frames with, for example, *Lucas-Kanade* algorithm (van Dam et al., 2013). As expected, the quality of the trajectory-based features, which are a key determinant of classification methods' performance, depends on the accuracy of the trajectories' tracking and body-parts segmentation previously calculated (Egnor & Branson, 2016).

On the other hand, pixel-based features can be derived directly from the raw pixel values of the video sequences. There are several described techniques, which include extracting information from local patches (de Chaumont et al., 2019; Jhuang et al., 2010) using feature descriptors such as Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005), combining multiple patches within the sequence using interest-point detectors (Dollár, Rabaud, Cottrell, & Belongie, 2005), or exhaustively (H. Wang, Ullah, Klaser, Laptev, & Schmid, 2009).

The selection of the type of features largely depends on the tracking information that was obtained. If the body parts were successfully segmented, trajectory-based features can be extracted for classification. If, on the other hand, tracking parts' data are not available, pixel-based features may be more effective (Egnor & Branson, 2016).

Given the complexity of animal movements and behaviors, the number of low-level representations of behavior tends to be very high, and in fact, feature vectors easily exceed the thousands. In this sense, dimensionality reduction techniques are a common approach to overcome this problem. These methods are frequently used in machine learning frameworks to reduce the number of variables under study, retaining the relevant information from the original space. In fact, the dimensionality of a dataset is closely related to the amount of required training data and, in turn, associated with data overfitting problems (the classifier learns exceptions that are specific to the training data and, when tested with new data, the performance is drastically reduced). Besides improving classifier generalization capability, dimensionality reduction techniques have advantages in reducing overall computational time and storage and, when applied to low dimensions, the visualization of data is improved. These techniques can be divided into two major categories: feature selection, which is the process of selecting subsets of relevant features to the predictive modeling problem, and feature extraction or reduction, mapping the original high-dimensional data onto lower-dimensional space. Both approaches can be used, independently, to improve model's classification, and there are already some published works that explore these techniques, such as Principal Component Analysis (PCA) or linear and non-linear discriminant analysis, in the context of animal's behavior (Berman et al., 2014; Hsu & Yttri, 2021; Ravbar, Branson, & Simpson, 2019; van Dam et al., 2013).

Afterwards, these behavior representative features are used for automatic behavioral phenotyping with machine learning approaches. Here, the main goal is searching for a classifier or a set of classifier functions that best reproduce the data, in a way to transform complex trajectories and features into biological and systematical data (Dell et al., 2014; Egnor & Branson, 2016). To achieve this, a mathematical model (classifier) is trained by interactively optimizing weights/parameters and increasing its accuracy in predicting behavioral patterns. The first experiments to detect specific behaviors in rodents started with Rousseau, Van Lochem, Gispen, and Spruijt (2000), using a NN approach to recognize rat behaviors with trajectory-based features. In 63.7% of all frames, the behavior was successfully recognized, when compared to the human-annotated ground truth. Supervised approaches based on sparse spatiotemporal and trajectory features were also described by Dollár et al. (2005), Burgos-Artizzu et al. (2012) and Giancardo et al. (2013), reaching 72%, 61.2% and 74.44%–82.67% of overall accuracy between automatic system and human graders, respectively.

General-purpose machine learning algorithms were also used to address rodent’s behavior analysis, including simple quadratic classifiers based on normal densities (van Dam et al., 2013), Support Vector Machines (SVM) (Hong et al., 2015; Zheyuan Wang, Mirbozorgi, & Ghovanloo, 2015), Random Forests (de Chaumont et al., 2019; Giancardo et al., 2013; Hong et al., 2015; Hsu & Yttri, 2021), and Boosting techniques (Kabra et al., 2013; Segalin et al., 2020). Sequential behavioral data can also be extracted to train more complex classifiers, such as Hidden Markov Models (HMM), in which the state transition probabilities are learned from the training sequence data. Jhuang et al. (2010) predicted mouse strain type based on detected behaviors, with an accuracy of 77.3%, by applying a modified HMM (SVMHMM). However, in general, training a behavioral system with HMMs may be disadvantageous, since these approaches require larger training sets and longer time to train. Besides that, when dealing with specific animal experiments, drug-treated or genetically modified animals, the behavior transition probabilities may be altered and may be different from those initially considered in the model (van Dam et al., 2013).

Contrary to these supervised techniques, some studies have explored unsupervised techniques to automatically identify animal behavior. Most of the published work clustered feature representations into behavior categories using different unsupervised clustering algorithms, such as hierarchical clustering (Z. Chen et al., 2020; Wiltschko et al., 2015), PCA (Marques, Lackner, Felix, & Orger, 2018; Wiltschko et al., 2015), t-distributed stochastic neighbor embedding (t-SNE) (Berman et al., 2014; Günel et al., 2019; Klibaite, Berman, Cande, Stern, & Shaevitz, 2017; Marques et al., 2018) or k-means (Braun, Geurten, & Egelhaaf, 2010; Schwarz, Branicky, Grundy, Schafer, & Brown, 2015).

Alternatively, to combine the strengths of both unsupervised and supervised methods, semi- , weakly- and self-supervised learning may be introduced in this context (Egnor & Branson, 2016; Robie et al., 2017). Semi-supervised methods learn from both unlabeled and labeled data. In weakly-supervised techniques, labeled data or a previously trained model is used as a prior and further replaced with more reliable signals. Finally, in self-supervised learning, useful representations from the unlabeled pool of data are learned and then fine-tuned with fewer labels. Although having potential in reducing annotation effort, the use of these methods for automatic classification of animal behavior is still in its infancy and need to be further explored (Egnor & Branson, 2016; Robie et al., 2017; Tanha et al., 2012; Wutke, Schmitt, Traulsen, & Gültas, 2020).

2.3.2 Direct behavior classification from raw inputs

With the new advances in machine vision and machine learning, recognizing the behavior itself can be performed automatically with deep learning techniques, without resorting to the first step of extracting positional or shape-based features. In fact, given the complexity and subtlety of some animal behaviors, it is expected that more informative features may be generated by deep learning, leading to better performances. Although a leading technique for both 2D and 3D human pose estimation and object recognition, CNNs for animal behavior analysis has recently started to be studied.

A pioneering study in this field applied standard 2D CNNs to recognize, per-frame and with low error rate (0.072%), two basic behaviors of *Drosophila* (regarding the position on the environment substrate) using 2D video sequences (Stern, He, & Yang, 2015). State-of-the-art pre-trained neural network models (e.g., *You Only Look Once* (YOLO) v3 network) were also used to recognize different postures in videos with multiple animals (Jin & Duan, 2019), where transfer learning has proven to be an interesting approach to increase performance while reducing the quantity of annotated data. Another study used simple feed-forward NN combined with features extracted from the *DeepLabCut* tool (A. Mathis et al., 2018) to prove that, by using deep learning techniques, it is possible to detect and quantify behavioral data as well or better than commercial solutions (Ethovision XT14 from Noldus, and TSE Multi Conditioning System from TSE systems). However, a small number of behaviors were analyzed, questioning its transferability to broader applications (Sturman et al., 2020).

Acknowledging the fact that temporal information may be an important piece in analyzing animal behavior, recent studies have tried to construct and adapt CNN-based networks to include both temporal and spatial information. A simple approach is to merge a sequence of adjacent frames into a single image, and then apply a 2D CNN to extract features from both spatial and temporal dimensions. Zhou and Xu (2018) applied this technique, and the pre-trained *CaffeNet* network was fine-tuned to automatically recognize 7 behaviors. Another approach is to combine optical-flow information (using a flow-generator as the *MotionNet* architecture) with RGB frames for further classification of behavior (Bohnslav et al., 2021). To avoid extra pre-processing steps, temporal sequence modeling techniques, such as Recurrent Neural Networks (RNNs), are an alternative for learning complex temporal dynamics. In fact, spatial feature extraction using 2D CNNs can be followed by recurrent layers for temporal integration (M. Marks et al., 2020; Nguyen et al., 2019). Finally, 3D CNNs can be constructed by substituting original 2D by 3D kernels, allowing for an extension of spatial dimensions along with the time domain. In this way, feature extraction and temporal-spatial information encoding of animal behavior can be achieved on an end-to-end basis (Jiang, Chazot, Celebi, Crookes, & Jiang, 2019; Murari, 2019; Nguyen et al., 2019; van Dam, Noldus, & van Gerven, 2020). More detailed background information on using CNNs for spatiotemporal encoding can be found in Appendix – Sequence Models for Extracting Spatiotemporal Features of Depth Sequences.

2.4 Representing behavioral dynamics

In order to make computational data easily analyzable by ethologists or researchers, it is necessary to summarize them in graphical representations with biological meaning (Figure 2.6). Trajectory-based data can be organized into simple ethograms, geometrical representations of movement patterns (Figure 2.6D) or 2D or 3D tracking representations inside behavioral mazes (Figure 2.6E and F). Behavioral data is usually represented by ethograms as well, to describe the fraction of time spent on each behavior (Figure 2.6A), or using kinematic diagrams to show the transition probabilities between behaviors (Figure 2.6B and C) (Egnor & Branson, 2016; Robie et al., 2017).

3. Depth cameras in animal behavioral analysis

The process of acquiring a machine-readable sequence of images or video data that can be efficiently analyzed, and that accurately describes the real world, depends on the final purpose, species, and type of environment to be studied. Nevertheless, in addition to automating the recognition itself, optimizing video quality must also be taken into consideration to improve the quality of that analysis. In this sense, to ensure high-quality video for computer vision, video equipment must share some particular features: guarantee uniform and sufficient lighting, high contrast between objects and background, and ensure standard conditions between trials (Robie et al., 2017).

Also, the complexity of animal behavior together with the lack of precise estimation of their poses in the available 2D systems impairs a detailed and complete behavior characterization. In fact, and given the setup constraints of 3D recording systems, recent technologies for 3D analysis from a single imaging device have appeared as an alternative to multi-camera systems. As mentioned in Section 2.2, these 3D imaging technologies use structured light or ToF operating modes to acquire range in addition to color images, in what is called an RGB-D camera (Figure 2.7A-D).

The structured light general principle reflects the process of projecting a known pattern of pixels onto a scene and inferring depth from the deformation of that pattern. The variation of the projected pattern against the reference one for a fixed distance provides a method to reconstruct the depth map (Litomisky, 2012). An example of this technology initially launched on the market is the *Microsoft Kinect v1* sensor (first generation of the *Kinect*-based sensors) (Figure 2.7B). This motion-sensing input device combines an RGB camera along with a depth sensor (an infrared (IR) laser-based projector and an IR camera) (Figure 2.7A). Based on the previous principle, the IR projector sends out a fixed pattern of light and dark speckles, which is captured by the IR camera and compared part-by-part to reference patterns. Commercial depth cameras typically have 8-bit RGB video stream's resolutions, up to 1920x1080 pixels, while the monochrome depth sensor has 16-bit resolution up to 1280x720 pixels, and a maximum depth range from 0.2 to 10.0 meters. Both video outputs typically work at 30 frames per seconds (fps) (or up to 90 fps for lower resolutions and optimal lighting conditions) (Intel®RealSenseTM, 2020; Z. Y. Zhang, 2012). *Kinect*-based cameras were originally created in 2010, for game purposes on *Xbox 360* video game consoles and *Windows* computers, but quickly spread to technological and scientific applications. In fact, some studies have shown that RGB-D cameras are even an accurate device for clinical purposes (Geros, Horta, & Aguiar, 2016; Khoshelham & Elberink, 2012; Muhammad et al., 2021; Thevenot, López, & Hadid, 2017). Interestingly, sensor systems that provide depth data have been custom-built for years, however at an extremely high price. The RGD-D cameras are now available at a low cost due to the development of *Kinect*-based models.

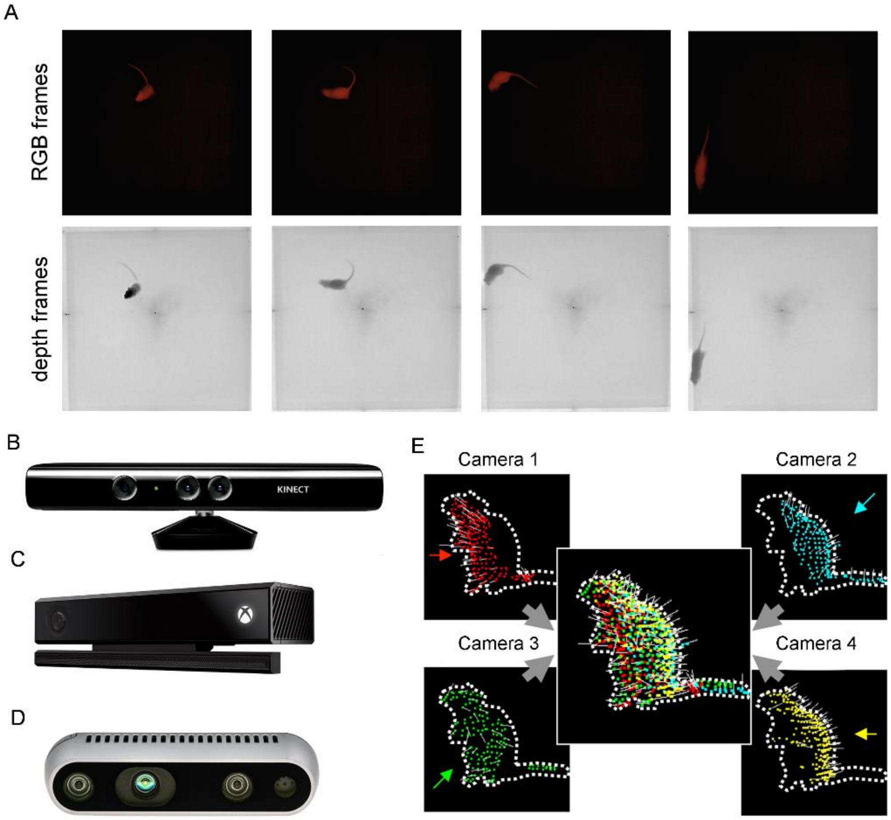


Figure 2.7 Depth-sensing technology. A. RGB and depth frames acquired from an RGB-D camera, under dim red light conditions. RGB-D cameras: **B.** *Microsoft Kinect v1* – 1st generation of Kinect-based cameras, operating with the structured light principle. **C.** *Microsoft Kinect* v2, operating with the time-of-flight principle. **D.** *Intel RealSense* D435 – new generation of low-cost RGB-D cameras. **E.** 3D reconstruction of animal's shape (central image) by merging point clouds captured by 4 depth cameras. Images in E adapted with permission from Matsumoto et al. (2013).

To circumvent the limitations of the pioneer low-cost range cameras, different sensor technology was created - ToF - to integrate the acquisition of accurate intensity data and range information into a single device at a low cost (Figure 2.7C). This technology uses a laser or light pulse to calculate the distance by measuring the signal's time of flight between the camera and the object, for each point of the image. Additional information on ToF functioning principles is available in Appendix – Fundamentals of Time-of-Flight operation system. Several advantages over the first generation can be highlighted, such as the reduction in shadow areas, and efficiency under different lighting conditions and speeds. However, interferences and multiple reflections can influence the performance of this technology and should be carefully addressed (Ganapathi, Plagemann, Koller, & Thrun, 2010b). The first experiments with ToF low-cost cameras (E Lachat, Macher, Mittet, Landes, & Grussenmeyer, 2015), namely the study of geometric and depth calibration, as well as system properties (outdoor efficiency, influence of materials and colors, pre-heating time, influence of frames averaging), revealed promising results for computer vision tasks.

Importantly, the overall advantages of RGB-D cameras fit perfectly into the context of animal behavior characterization: by working with an infrared sensor, contrast is independent of animals' coat color and environment light, and videos can be recorded in dark rooms, which will not affect animals' biological cycle (Figure 2.7A). Some studies have therefore addressed the problem of animal tracking and behavior analysis in 3D using RGB-D cameras. Tracking and pose reconstruction of rodents were initially studied by Ou-Yang et al. (2011) and Monteiro, Oliveira, Aguiar, and Cardoso (2012), using a single *Microsoft Kinect v1* camera. The former applied threshold methods to perform animal segmentation, and rat's mass center, size, and solid shape were extracted by geometric considerations. Rest and movement positions were also detected, as well as velocity in each time instant. Limitations such as the poor reconstruction of shaded parts of the rat and basic behaviors' identification were pointed out by the authors. Monteiro et al. (2012) addressed point cloud representation of mice using different segmentation methods. Results showed better performances for a fixed background method, achieving a true positive rate of 68%. Animal tracking using a single range camera was further explored by Kulikov et al. (2014) and later by Saberioon and Cisar (2016) for 3D tracking of multiple animals. Background segmentation was achieved using threshold techniques, and animal detection and identification were possible thanks to Pérez-Escudero et al. algorithm (Pérez-Escudero et al., 2014). Point cloud segmentation of rodents was further studied using a single *Kinect*-based sensor (*Creative 3D SenZ* camera) by Paulino Fernandez, van Dam, Noldus, and Veltkamp (2014). To segment animals' whole-body from the background, *RANdom Sample Consensus* (RANSAC) algorithm was applied, and rodents' point clouds were reconstructed. The segmentation task using a basic background scenario was facilitated, which is the major limitation pointed out by the authors.

The use of multiple depth cameras for 3D tracking was introduced later, as a way to cover the entire surface of the object. The 3D points are captured by multiple depth cameras and different viewpoints are integrated into one single 3D point cloud. Nakamura et al. (2016) used this approach for markerless motion capture of monkeys by fitting a physical skeleton model. Taking advantage of the recent progress in the machine learning field, Ebbesen and

Froemke (2020) used a deep CNN to detect key points (nose, ears, base of tail, and neural implant) in the RGB images and extracted the 3D point cloud from depth images of multiple cameras. Because these methods do not depend on feature engineering techniques or physical models for tracking multiple animals, they can capture movement dynamics of unmarked mice with high temporal (~60fps) and spatial (~2mm) precisions.

Behavior in rats, including social and sexual interactions, was firstly studied by Matsumoto et al. (2013), using multiple *Microsoft Kinect v1* cameras. In here, de Chaumont et al. (2012) tracking system was expanded to RGB-D streams, and body parts' tracking was performed by fitting a skeleton model of the animal to the 3D images (Figure 2.7E). Although identifying, for the first time, several behaviors associated with sex interactions using 3D information on ruled-based models, manual interventions, unsynchronized cameras, and incapability in detecting new user-defined behaviors can be pointed out as the main limitations of this work. Two-animal interactions were analyzed by Hong et al. (2015), using two different range cameras (for a comparative study), along with a side-view RGB video camera. The tracking algorithm was based on background subtraction techniques, in which the background model was constructed using unoccupied regions of the cage, and behavior analysis were tested using different machine learning classifiers (including SVM, adaptive boosting, and random decision forest). The need for animals with distinct coating colors, and the absence of body-parts segmentation are some of the limitations of this system. In a recent study, depth-sensing cameras were combined with RFID for long-term real-time analysis of mice's social behavior (de Chaumont et al., 2019). Animals' segmentation was performed by dynamic background subtraction, machine learning methods were applied for animal identity recovery, and social interactions were identified using the same approach as de Chaumont et al. (2012). Finally, *Microsoft Kinect v1* was used to construct a behavior recognition and analysis system, developed by Zheyuan Wang et al. (2015). The segmentation and behavior analysis was performed using depth information in the following pipeline: reference image-static difference technique was used to track animal's whole-body; pre-processing methods were applied to reduce noise level; geometric features were extracted using morphological operations, and, finally, an SVM classifier was trained, to identify 5 basic rodents' behaviors (*standstill*, *walking*, *grooming*, *rearing*, *rotating*). Although having higher accuracy rates, the background technique chosen to perform animal tracking is insufficient when applied in more complex and naturalist environments.

To the best of the author’s knowledge, combining deep learning methods with data from depth-sensing technology has been poorly explored for animal behavior analysis, and only a few studies have recently started to use both technologies to classify very basic behaviors (Nourizonoz et al., 2020).

4. Infrared thermal imaging for assessing animal health and welfare

Thermal sensors also play an important role in the computational ethology field (Giancardo et al., 2013; Magdalena Mazur-Milecka & Ruminski, 2020; Xudong, Xi, Ningning, & Gang, 2020) (Figure 2.8). Thermal or IR imaging cameras work by detecting and measuring the infrared radiation emanating from objects (their heat signature). They contain an optical system fitted with a lens that allows IR frequencies to pass through and focuses them onto a detector chip or sensor array containing multiple detector pixels arranged in a grid. Each pixel, also called microbolometer pixel, reacts to the infrared wavelengths hitting it, causing a change in its electrical resistance, which can be probed by passing a current through the device. In this sense, temperature changes can be read out as electronic signals, that are sent to a processor to produce an image as a color map of different temperature values. The sensor array may have different pixel resolutions, usually lower than visible light sensors. Since thermal detectors need to sense energy of much larger wavelengths than visible light, each sensor pixel needs to be significantly larger, resulting in a lower resolution (Lloyd, 2013; Ostrower, 2006).

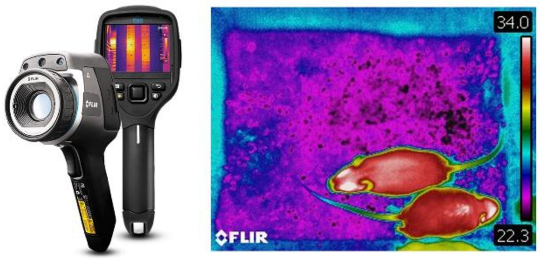


Figure 2.8 Thermal imaging technology. FLIR E60 camera (left), from FLIR Systems (USA) (one of the most broadly used thermal camera brand), and acquired thermal image (right), with temperature values ranging from 22.3 to 34.0 degrees.

4. Infrared thermal imaging for assessing animal health and welfare

With the recent availability of low-cost thermal imaging technologies, infrared thermography has increasingly been incorporated into animal research and veterinary medicine. In particular, the assessment of animal welfare has become an important concern in laboratory environments, as to promote the search for different methods that prevent, minimize and relieve any discomfort experienced by research animals (Mota-Rojas et al., 2021). Such contactless technologies are a promising alternative for invasive methods, which, by stressing/perturbing the animals, may affect the temperature readout. Besides allowing for non-invasive recordings in more naturalistic environments (dark rooms, dynamic backgrounds), thermography can itself be a useful method for the analysis of the overall physiopathological state of the animals, since modifications in animal's body temperature can be associated with physiological changes (such as the presence of infections, increased metabolic activity, lesions, or stress) (Całkosiński et al., 2015; David, Chatziioannou, Taschereau, Wang, & Stout, 2013; Harshaw & Alberts, 2012; Ludwig, Gargano, Luzi, Carenzi, & Verga, 2010).

Despite the recent advances in machine vision and machine learning areas, most studies still rely on cameras' proprietary software for direct pixel readouts or to extract mean temperature in manually defined ROIs (Bautista et al., 2017; Całkosiński et al., 2015; David et al., 2013; Gabbi et al., 2021; Joy et al., 2021; Sutherland et al., 2020; H. Yuan, Liu, Wang, Wang, & Sun, 2022). In addition to reproducibility and standardization problems, proprietary software has a small number of features that limit a complete quantitative analysis. Conventional machine vision techniques have been poorly explored for automatic segmentation of animals' bodies in thermal images, using thresholding methods for background subtraction (Lecorps, Rödel, & Féron, 2016; Manzano-Szalai et al., 2016; Martinez, Ghamari-Langroudi, Gifford, Cone, & Welch, 2015; Magdalena Mazur-Milecka & Rumiński, 2017), and watershed algorithms for individualizing multiple animals (Giancardo et al., 2013). More recently, deep learning methods have started to be applied, using, for example, U-Net-based networks for direct segmentation of multiple animals' bodies (Magdalena Mazur-Milecka & Ruminski, 2020), or Mask R-CNN model in RGB images for the segmentation of ROIs in thermal images (S. Kim & Hidaka, 2021). It is important to highlight that the vast majority of published studies apply thermal imaging technologies to assess animal health and welfare in the food industry environment, which is currently an active area of research. Hopefully, the knowledge from this field can help accelerate future studies in the laboratory context, in parallel with advances in the computational ethology field.

5. Control of operant mazes based on real-time behavioral analysis

Improvements in the way researchers record and analyze animal behavior have been rapidly emerging over the last years in parallel with breakthroughs in computer vision and imaging technologies. These advances not only enable high throughput and fully automated analysis but also increase the quantity and quality of extracted behavioral data. Most of the previously described techniques focus on offline quantification across several species and different patterns of behavior. However, automatically recording and measuring behavior, and further distilling it down into meaningful metrics, are not sufficient to effectively correlate behavioral expressions with the environment or neuronal activity. The behavior should be detected in real-time, allowing for online closed-loop (environment or organism) manipulations based on the current identified behavioral expression. In this sense, besides accurate quantification, real-time detection of behavioral dynamics is essential to construct movement-triggered feedback systems and brain-machines interfaces, and to further allow online reinforcement of user-defined patterns of interest.

Real-time measurements are already possible thanks to advances in imaging and computing technologies, being performed as images are acquired and removing the need for storing huge amounts of video data (Alex Gomez-Marin et al., 2012). In fact, several studies have tried to address feedback control in real-time, combining machine vision (P. Aguiar, Mendonca, & Galhardo, 2007; Lopes et al., 2015; G.-W. Zhang, Shen, Li, Tao, & Zhang, 2019) or machine learning (de Chaumont et al., 2019; Forys, Xiao, Gupta, & Murphy, 2020; Kane, Lopes, Saunders, Mathis, & Mathis, 2020; Nourizonoz et al., 2020; Schweihoff et al., 2021; Sehara, Zimmer-Harwood, Larkum, & Sachdev, 2021) techniques for extracting animal behavioral patterns, with hardware devices to close the loop in behavioral experiments (Figure 2.9).

Feedback control signals can be generated by simply detecting spatial coordinates of the animals, using thresholding techniques for background subtraction and geometrical considerations for centroid's calculations (P. Aguiar et al., 2007; G.-W. Zhang et al., 2019) (Figure 2.9A). The control system can also be constructed to allow closed-loop feedback with basic body parts' position estimation, using traditional machine learning methods (de Chaumont et al., 2019). Deep learning techniques can be applied to further improve pose estimation by tracking multiple points (Forys et al., 2020; Sehara et al., 2021) or animal's body parts in 2D (Kane et al., 2020; Schweihoff et al., 2021) (Figure 2.9B and C) and 3D (Nourizonoz et al., 2020). The combination of these computational methods with hardware devices makes real-time or low-latency feedback control a promising framework for behavioral experiments.

To allow the study of animals' social interactions in controlled and customizable environments, some studies have integrated robotics in animal behavior studies (Gribovskiy, Halloy, Deneubourg, Bleuler, & Mondada, 2010; Halloy et al., 2007; Changsu Kim, Ruberto, Phamduy, & Porfiri, 2018; Q. Shi et al., 2013). In these robotic-based platforms, stimuli are adapted based on the automatic detection of behavioral patterns of interest, and biologically-inspired replicas interact with the animal under study for a dynamic interplay in closed-loop control systems.

Although these high-throughput approaches have already acknowledged the potential of combining strong computational techniques with cutting-edge hardware devices, some limitations, such as the need for manual interventions, complicated and costly setup, and absence of direct recognition of behavior, need to be addressed for complete integration in laboratory environments. Nevertheless, important steps towards more robust, rich, and reproducible animal experiments have been taken over the last years, and future improvements have an important position in revolutionizing the field of behavioral neuroscience.

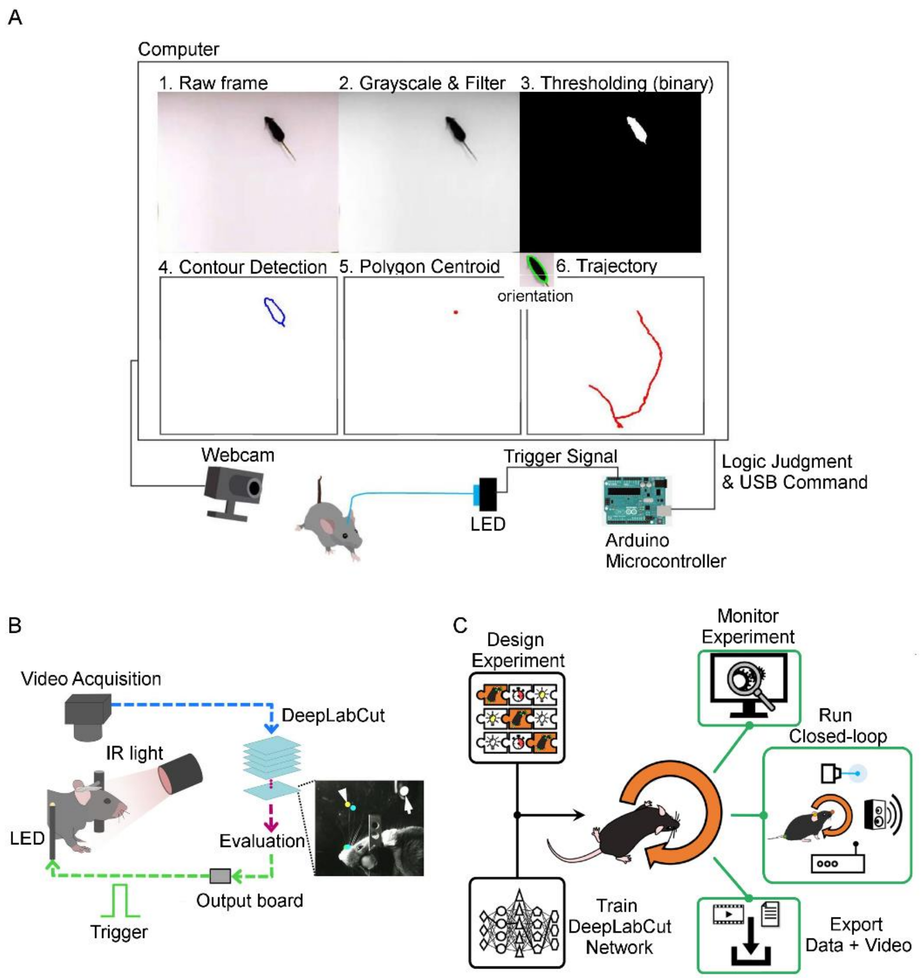


Figure 2.9 Closed-loop approaches for controlling behavioral experiments. A. *Track-Control* system: images from a webcam are streamed into the computer, where each frame is initially converted to a grayscale image. After gaussian filtering and binarization, animal contour is detected by polygon fitting, and the centroid of that polygon is determined. Logic computations are then performed to determine whether a command signal will be sent via the universal serial bus (USB) port, and finally, the signal is used to trigger hardware output (e.g. light-emitting diode (LED) light) via an Arduino microcontroller. **B.** Closed-loop setup using *DeepLabCut* network for body-parts' estimation. high-speed video of a head-fixed mouse is acquired under infrared (IR) illumination. Whisker positions are estimated for each frame, and a digital output, turning on an LED, is generated based on estimated positions using *DeepLabCut* network. **C.** *DeepLabStream* workflow: an experimental protocol is initially designed using a sequence of modules (puzzle pieces), and a trained *DeepLabCut* network is integrated into the *DeepLabStream*, providing three different outputs for every experiment. Experiments can be monitored on a live stream and the experimental protocol

is run based on the automatic posture detection. Finally, the recorded video and experimental data are exported for further analysis. Images adapted with permission from: A. (G.-W. Zhang et al., 2019), B. (Sehara et al., 2021), and C. (Schweihoff et al., 2021).

CHAPTER 3

A High-Throughput Analysis Software for Non-Invasive Thermal Assessment of Laboratory Mice

This chapter was based on the following original research paper:

Franco, N. H., Gerós, A.*, Oliveira, L., Olsson, I. A. S., & Aguiar, P. (2019). ThermoLabAnimal – A high-throughput analysis software for non-invasive thermal assessment of laboratory mice. Physiology & behavior, 207, 113-121.*

** equal contribution*

1. Abstract

Body temperature changes in laboratory mice are often assessed by invasive and stressful methods, which may confound the measurement. Infrared thermography is a possible non-invasive alternative, but the cost of standard thermal cameras, lack of dedicated software for biomedical purposes, and labor-intensiveness of thermal image analysis have limited their use. An additional limitation lies on the scarcity of research on the causing factors of differences between body surface and core body temperature. We propose a method for automatic processing of non-invasive mean body surface temperature in freely-moving mice, using dedicated software for thermal image analysis. While skin surface temperature may not necessarily be linearly correlated with core body temperature (in itself an imprecise concept), under standardized environmental conditions, such as those in which laboratory animals are kept, mean body surface temperature can provide useful information on their thermal status (i.e. deviations from normothermia, namely hypo- and hyperthermia). We developed a publicly available software (*ThermoLabAnimal*) that includes an imaging analysis workflow/algorithm for automatic segmentation of the pixels associated with the animal from the pixels associated with the background, removing the need for manually defining the area of analysis. A batch analysis mode is also available, for automatic and high-throughput analysis of all image files located in a folder. The software is compatible with the most widespread thermal camera manufacturer, *FLIR Systems*, as well as with the low-cost *Thermal Expert TE-Q1* miniaturized high-resolution thermal camera used for this study. Furthermore, the software has been validated in a mouse model expressing non-transient hypothermia, where the thermal analysis results were compared with readings from implanted thermo-sensitive passive integrated transponders tags. Thermography allows for thermal assessment of laboratory animals without the effect of handling stress on their physiology or behavior. Our automatic image analysis software also removes observer errors and bias, while speeding up the data processing.

Keywords: Temperature Variation • Mice • Analysis Software • Infrared Thermography • LPS-Induced Hypothermia • Mean Body Surface Temperature

2. Highlights

- Body temperature variation gives valuable information on animal health and welfare
- Thermal assessment of laboratory animals raises technical and welfare challenges
- IR thermography is an option, but analysis is laborious and prone to variability
- We propose a novel user-friendly software for analysis of thermal images of mice
- The new analysis methodology was validated in LPS-injected mice showing hypothermia

3. Introduction

Body temperature variation can provide valuable information in animal-based biomedical research. An increasingly used method for monitoring thermal changes in laboratory animals is infrared thermography (IRT), which allows contactless estimation of body surface temperature variations, with several applications in research (Bautista et al., 2017; David et al., 2013; Gjendal, Franco, Ottesen, Sørensen, & Olsson, 2018; Lecorps et al., 2016; Meyer, Ootsuka, & Romanovsky, 2017; Mufford et al., 2016; Tattersall, 2016). This approach prevents the scientific and welfare impact of more invasive methods for thermal assessment. The use of rectal or infrared thermometers directly affects the temperature readout, due to a core temperature rise resulting from a hyperthermic stress response mediated by the sympathetic-adrenal and the hypothalamic-pituitary-adrenal (HPA) axes (Adriaan Bouwknecht, Olivier, & Paylor, 2007; C. Gordon, 2012). This is accompanied by a sympathetically-mediated vasoconstriction, resulting in a transient drop in the extremities (particularly in the tail), which rebounds by the warming from vasodilation for core heat dissipation (A. Marks, Vianna, & Carrive, 2009; D. M. Vianna & Carrive, 2005).

This response is moreover heightened by repeated handling (C. J. Gordon et al., 2008; Hartinger, Külbs, Volkers, & Cussler, 2002) and can even be elicited prior to handling, in response to alarm calls from handled animals (C. J. Gordon et al., 2008; Hartinger et al., 2002; Zethof, Van Der Heyden, Tolboom, & Olivier, 1994). Surgically implanted sensors can be read from a distance (Sanford, Yang, & Wellman, 2011); however, these methods are not non-invasive in themselves. Sensor implantation requires surgery under general anesthesia, and adds considerably to workload, as well as animal welfare impact (Helwig, Ward, Blaha, & Leon, 2012; Morton et al., 2003; Tang & D. Sanford, 2002).

The use of IRT is not new in veterinary research and practice (Church, Cook, & Schaefer, 2009; Fabio Luzi, Malcolm Mitchell, Leonardo Nanni Costa, & Redaelli, 2013; Herbut & Walczak, 2013) but its use with laboratory animals has been limited, probably as a result of the cost (in the thousands of *Euros* range) and bulkiness of most IRT cameras, which are incompatible with measurements in small rodent cages, in combination with the lack of dedicated analysis software. To the best of our knowledge, there are no infrared thermography imaging software dedicated/tailored for laboratory mice, and researchers often rely on general purpose and proprietary software for direct pixel (temperature) readouts (e.g. *Fluke SmartView*, *FLIR Tools*, or *Testo IrSoft*). Despite these difficulties, IRT technology has been proven useful in laboratory animal science in identifying housing problems (David et al., 2013), following neonatal development (Harshaw & Alberts, 2012), identifying stress (Ludwig et al., 2010) and monitoring infections (Całkosiński et al., 2015; Vadlejcha et al., 2010), among others.

In a previous study, we found IRT readout of mean body surface temperature (MBST) to be a more reliable parameter for non-invasive assessment of surface temperature variation in freely-moving mice than either maximum eye or tail temperature (Gjendal et al., 2018). This parameter has also been successfully used in previous studies in mice (Lecorps et al., 2016; Manzano-Szalai et al., 2016; Martinez et al., 2015; Mufford et al., 2016), as well as in other species, including rabbit pups (Bautista et al., 2017; Gilbert, McCafferty, Giroud, Ancel, & Blanc, 2012), chickens (Herborn, Jerem, Nager, McKeegan, & McCafferty, 2018) and animals in the wild (McCafferty, Gallon, & Nord, 2015; Powers et al., 2017). The inherently limited accuracy and noise of thermal cameras, the inhomogeneity in the animals' surface emissivity, along with the presence of other interfering heat sources, render temperature measurements based on a single spot error-prone. A better approach may be to average larger body surface areas, which are relatively isothermal if the ambient temperature remains unchanged, as is the case in environmentally controlled laboratory animal facilities. In a previous study (Gjendal et al., 2018), we assessed MBST by manually defining regions-of-interest (ROIs) corresponding to the contour of each animal, in each image. Image analysis was then performed on the ROIs to extract, for example, mean thermal values. This process was found to be laborious, time-consuming, user-dependent, error-prone and difficult to perform consistently.

In response to these challenges, we hereby present a dedicated computational tool developed for automatic assessment of MBST from a virtually unlimited number of thermal images (*ThermoLabAnimal*). This new software greatly facilitates surface temperature quantification of freely-moving mice. The software uses image segmentation algorithms that remove background heat and focus the analysis in specific ROIs. Further, it allows both individual thermal image screening and batch analysis of thermal images. The type of image analysis to perform, single or batch, as well as other options and features, are chosen using a simple and intuitive graphical user interface (GUI). The software reads thermal images saved as comma-separated values (.CSV) files, a format to which most thermal cameras can save images directly, or convert to, using the camera's software.

We validated our computational tool and analysis protocol in a mouse model of acute septic shock by intraperitoneal injection of a high dose of lipopolysaccharide (LPS), of which a pathophysiological hallmark is quick-onset, pronounced hypothermia. The aim was to assess whether MBST could reliably inform on body temperature changes in this model, as compared with data from the well-established method of implanted thermo-sensitive passive integrated transponders (PIT) tags. Subcutaneous PIT tags are a widely used method for non-invasive measurements but have many limitations, as presented before. The objective is not to compare the IRT methodology itself with PIT tag measurements, but instead to emphasize the advantages that *ThermoLabAnimal* brings when using IRT. In compliance with the 3Rs principle of Reduction, measurements were obtained as additional data from an already scheduled experiment, in an ongoing project in our institution.

The software can be downloaded from https://github.com/ThermoLabAnimal.

4. Materials and Methods

4.1 Thermal Cameras

A miniaturized (47 mm x 25 mm x 25 mm) *Thermal Expert TE-Q1* (i3 Systems, Korea) thermal camera was used in all animal thermal imaging experiments. According to the manufacturer, it has a 384x288 pixel's resolution, thermal sensitivity below 50 mK and ± 3℃ or ± 3% accuracy (depending on environmental conditions). To assess camera's accuracy, we tested the accuracy of the low-cost using a blackbody radiation source (see Supplementary Information).

The camera was placed 30 cm above the cage, alongside an RGB *Microsoft LifeCam HD*-3000 camera (Redmond, Washington, USA) to allow visual identification of the animals, with the camera's optical axis perpendicular to the cage floor. Both cameras were connected via OTG universal serial bus (USB) to an *ASUS* T101Ha computer (running OS *Windows* 10) and operated via the *Thermal Expert* proprietary software (version 1.7.0). Emissivity was set to 0.95.

All animal thermal measurements were carried out after an equipment warm-up period of 30 minutes (mins) (i.e. the camera was turned on 30 mins before the first animal measurement, and continued to operate continuously until completing the last measurement). With the purpose of widening and validating the compatibility of this computational tool, we made it capable of reading and analyzing files generated by *FLIR* cameras (the most broadly used thermal camera brand). Compatibility was assessed using images of freely-moving mice taken by a *FLIR E60* (320×240 pixels' resolution, < 50 mK thermal sensitivity and ± 2°C or ± 2% accuracy; *FLIR Systems*, USA), after being converted to .CSV using the *FLIR* proprietary software (*FLIR Tools*).

4.2 Software development

A dedicated thermal imaging analysis software, named *ThermoLabAnimal*, was developed in *MATLAB* R2018a (*The MathWorks* Inc., USA) and took advantage of the advanced image analysis algorithms available in *MATLAB*'s Image Processing Toolbox. In order to automatically segment the animals' bodies, and separate the background from foreground pixels, a global thresholding method was used, where the temperature threshold value was calculated in order to minimize the intra-class variance (*Otsu*'s method) (Otsu, 1979). The segmentation mask was corrected with the removal of small patches (associated in general with urine spots), by eliminating all the components (using an eight neighbor's connectivity) with an area smaller than a defined (but user modifiable) value. This corrected mask was then used to perform the thermal analysis statistics exclusively in pixels associated with the animal(s). The analysis workflow also included an optional step for the identification of multiple animals in the mask. In situations where animals are not juxtaposed, a connected-components analysis can be performed and the mask is segregated into individual components (associated with the individual animals).

The GUI for the software was developed using *MATLAB*'s graphical interfaces development environment (GUIDE). Import capabilities were included providing the ability to load files with thermal data from both *Thermal Experts* and *FLIR Systems*' cameras. *ThermoLabAnimal* was developed for compatibility with both *Microsoft Windows* and *Apple* operating systems.

4.3 Animals

To validate the *ThermoLabAnimal* software, we obtained thermal images from C57BL/6 mice undergoing procedures in a different project (DGAV license 009951), which involved large-dose LPS injection via the intraperitoneal route. No additional intervention, other than the thermal imaging, was imposed on the animals for the purpose of the present study. We present pooled data from nine mice from three of the experimental groups, the control group and two of the experimentally treated groups for which surface temperature variation did not differ significantly from that of control mice (one-way ANOVA, *df* = 2, *ssq* = 2,120, *F* = 1,426, $p = 0.244$). To preserve confidentiality of yet unpublished results, treatments are not disclosed. Four control mice were housed with one mouse from an experimental group in a Type II (dimensions: 268 mm x 215 mm x 141 mm, floor area: 370 cm^2) cage, whereas the other seven mice were housed together in a Type III (dimensions: 425 mm x 276 mm x 153 mm, floor area: 820 cm^2) cage (*Tecniplast*, Italy). All cages contained corncob bedding (LBS serving Biotechnology, United Kingdom), absorbent paper (*Renova*, Portugal) for nesting material, and a cardboard tube (*LBS* serving Biotechnology, United Kingdom). Mice had access to *Teklad Harlan* 2014S (*Envigo*, United Kingdom) chow and tap water *ad libitum*. Room temperature was maintained at 20-24°C with a relative humidity of 45-65%. Mice were housed under a 12:12 h dark/light cycle with lights on between 08:00 h and 20:00 h.

4.4 Experimental Protocol

The mice were subcutaneously implanted with a *Biotherm* thermo-sensitive PIT tag, using a 12-gauge needle, under short-term (<5 mins) isoflurane anesthesia (administered with 1 bar of oxygen, at 5% concentration for induction and 2% for maintenance). PIT tags were read by a *Destron Fearing* GPR+ handheld reader (with a reading range between 33-43°C). The puncture site was sealed with surgical glue, but in two of the animals the tags were nonetheless exteriorized and found the following day on the cage bedding. These animals, as well as a third which maintained normothermia throughout the experiment (possibly due to error in administration of LPS), were removed from the study. Hence, subcutaneous and mean body surface temperature of a total of N = 9 animals (three per each of the treatment groups) was monitored. One week after PIT tag implantation, animals were marked in the tail (by colored marker) for fast visual identification, and injected intraperitoneally with 12.5 mg/kg of LPS from Escherichia coli O111:B4 (Sigma Aldrich). Thermal images were collected during a period of 6 hours: every 10-20 mins post-injection during the first hour and a half (namely at (at 0:20, 0:32, 0:45, 0:58, 1:12, 1:22, 1:36, 1:48 hours post-injection), and then at more spaced intervals (namely at 2:28, 3:00, 4:42, 5:15, 5:44, 6:13 hours post injection). At each time-point, the cage lid was open, the nesting material and cardboard tube were gently removed and three images were taken of the group, when all animals had all four paws on the cage floor. At each time-point, image collection of the group of animals took about 1 min to perform. At t = 4.67 h, mean subcutaneous temperature reached 33.3 °C, and cages were placed over a warm heating pad.

4.5 Statistics

Significant changes between time-points were assessed by paired-samples *t-test*, with a *Holm-Bonferroni* correction for multiple comparisons. The threshold for significance was set at $p <$ 0.01. Temperature decrease was expected to be very pronounced, following LPS challenge. Through a power calculation (using *G-Power* software) for a one-sided matched-pairs test with $\alpha =$ 0.01, a power of 80%, and a standardized effect size *Cohen*'s *d* = 2 (mean difference at least as large as two standard deviations) a total sample size of six mice was deemed sufficient. Our sample size of N = 9 would allow identifying smaller differences, with the same statistical power, for the same significance level. The *IBM SPSS* Statistical package (version 25) was used.

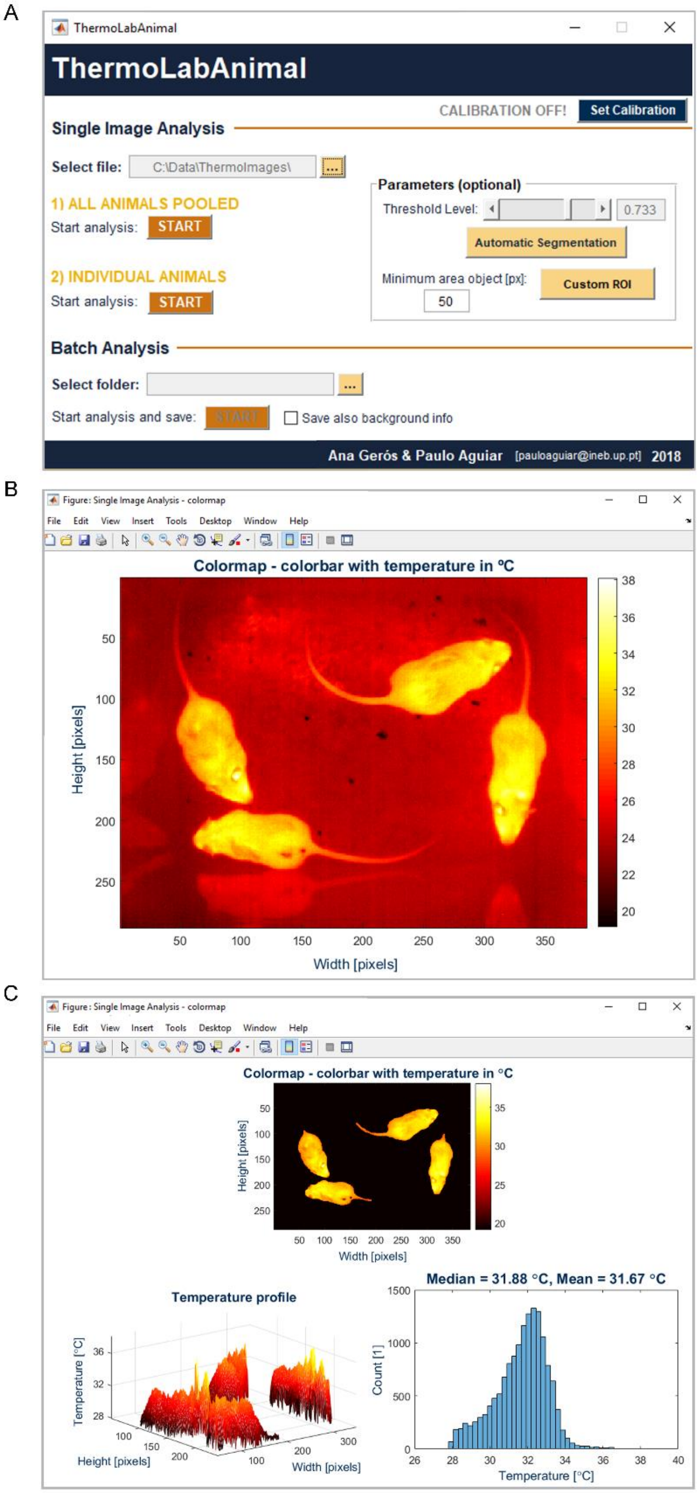
5. Results

5.1 *ThermoLabAnimal* graphical user interface

The computational tool *ThermoLabAnimal* runs on both *Microsoft Windows* and *Apple*'s operating systems, and has a user-friendly GUI allowing an easy workflow of importing the thermal images, analyzing the data and saving the results (Figure 3.1). Two modes of analysis are available in the main GUI (Figure 3.1A): single image and batch analysis (performed on all files in a specified folder).

In the single image analysis mode, the user can select a thermal image in .CSV format. The segmentation algorithm automatically creates a mask for the animals, separating foreground pixels (animal) from background pixels. From the single image analysis, two figures are generated. The first figure shows an unaltered false-color thermal image reconstructed from the .CSV file, along with the thermal color legend (Figure 3.1B). The second figure is composed of three elements (Figure 3.1C): i) a reconstructed image with the automatic mask applied to eliminate the background; ii) a 3-dimensional (3D) temperature map (x, y, °T), which can be freely rotated; and iii) a histogram of temperatures in the mask (animal surface). All animals in a cage can be considered an experimental unit or, alternatively, each can be analyzed individually (Figure 3.2). If in the thermal image the animals are not overlapping, *ThermoLabAnimal* can automatically segment each one individually (Figure 3.2A) and present separate thermal analysis for each animal (Figure 3.2B). All output figures are interactive and can be exported.

While the automatic segmentation algorithm works in most cases, mask fine-tuning may be necessary in some conditions (Figure 3.3). For this purpose, *ThermoLabAnimal* provides manual adjustment of the threshold with immediate visualization feedback. Furthermore, in order to eliminate odd detections (e.g. urine spots), the software allows automatic removal of all blobs smaller than a user-defined number of pixels, for both single and batch image analyses (Figure 3.3A). To keep *ThermoLabAnimal* as versatile as possible, users can also perform the thermal analysis of a user-defined ROI (Figure 3.3B). This feature can be used to analyze local temperatures at specific anatomical regions (e.g. eye, tail, wounded or inflamed region, etc.).



Select file: C:\Data\ThermoImages\

Threshold Level: 0.733

Minimum area object [px]: 50

□ Save also background info

Figure 3.1 Graphical user interface (GUI) of the *ThermoLabAnimal* software. A. The main GUI provides simple access to the software tools/features, namely the option between single image analysis and batch analysis. **B.** After loading a thermal image in single image

analysis, the user can inspect the image using standard visualization tools (zoom, read specific pixel values, etc.). **C.** The automatic segmentation and thermal background elimination algorithms allow isolation of the pixels associated with the mice. A histogram for surface temperature distribution of all mice in the frame, and a three-dimensional representation of the temperature map, integrate the analysis outputs.

The batch analysis mode allows automatic and high-throughput thermal data analysis from all the image files located in a specified folder. The batch analysis produces two outputs. The first is a single spreadsheet document, listing the mean and median MBST of all animals identified in each thermal picture. The second is, for each .CSV file, an automatically-generated composite image similar to the output of single image analysis with a false-color thermal image, a thermal histogram, and a 3D temperature map (x, y, °T). In the batch analysis mode, the user can choose to also export information about the background of each image in the folder. In parallel, a similar composite image is generated, containing information about the background, obtained using the counter-mask that separates the foreground and background pixels.

5.2 Software output

The software successfully recognized each individual animal from their thermal background (Figure 3.1), in every image, providing a histogram for the distribution of group mean and median body surface temperature, along with a fully-adjustable 3D map (pixel coordinates vs temperature). The thermal profile of specific areas could be obtained by defining a contour/ROI (Figure 3.3). Using the batch analysis function, a dataset of 1384 .CSV files were automatically analyzed, obtaining MBST in a standardized way, thus eliminating the possibility of unwanted variation and operator bias. The output was a single spreadsheet file with values for group mean and median temperature for every .CSV image, as well as an automatically generated composite image for each file analyzed. Even when surface temperatures differed considerably between animals in the same frame, the software was successful in highlighting each animal in the frame. However, in the extreme cases when the MBST of some animals decreased to the point where their thermogram blended with the background, the detection threshold had to be manually adjusted for those images.

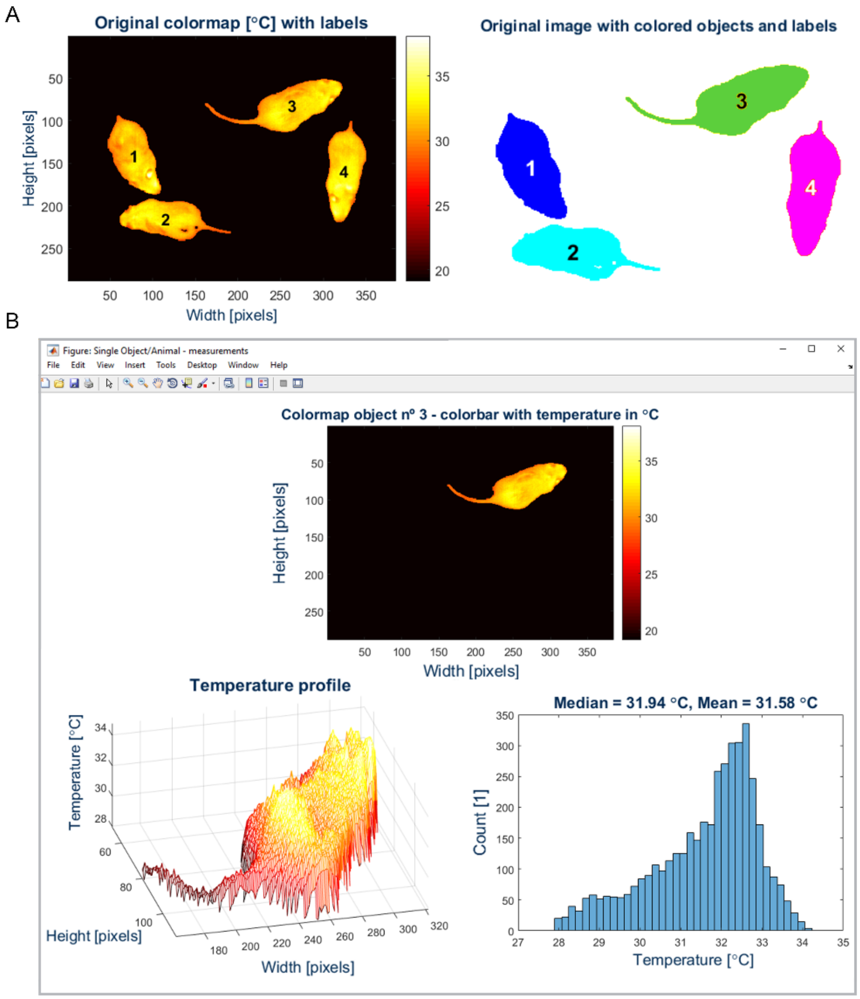


Figure 3.2. The software is also capable of individualized thermal analysis, as long as the animals are not overlapping. A. Using the “Individual Animals” analysis option, the software detects and labels each animal. However, animal identity is not maintained between images. **B.** For each separated animal in the image, the software outputs a dedicated thermal analysis window.

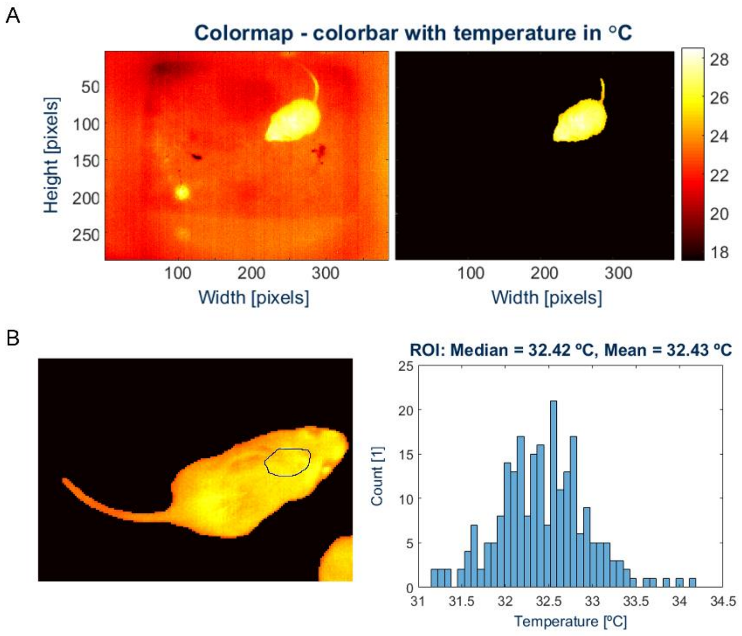


Figure 3.3 Additional tools in *ThermoLabAnimal*. A. To eliminate odd occurrences, such as urine spots, an added feature allows removing all blobs smaller than a user-defined number of pixels, in both single image and batch analyses. **B.** Defining regions-of-interest (ROIs) is also simple, allowing temperature analysis in specific user-defined areas (e.g. an inflammation area, the tail, or any other anatomical area of interest).

5.3 High-dose LPS challenge

Immediately following a high-dose LPS challenge, there was a small subcutaneous temperature rise in both the PIT tag readout and MBST, yet not found to be significant (Figure 3.4). This was followed by a steady decrease in subcutaneous temperature, along with a faster and more pronounced decrease in MBST, which remained low until decreasing further at t = 5.25 h. At t = 4.67 h mean subcutaneous temperature was estimated as 33.3°C, with most animals (six out of nine) falling below the reading range of the PIT tag reader of 33.0°C (and registered as 32.9°C). At t = 4.67 h cages were placed over a warm heating pad, in compliance with institutional animal welfare guidelines. This pattern was consistent for all three groups (Figure 3.4A). The aggregated information from all groups is presented in Figure 3.4B.

6. Discussion

We developed software for automatic analysis of thermal images of freely-moving mice, and tested it on a mouse model of sepsis-induced hypothermia, comparing its output with the readout from thermo-sensitive PIT-tags. The laborious and time-consuming nature of thermoimage analysis is presently a barrier to the wider use of thermography in research with laboratory animals. Researchers end up having to rely on general-purpose imaging software (typically associated with camera manufacturers) that is primarily focused on direct pixel (temperature) readout. While some allow manual definition of ROIs that can be used to calculate statistics, this generic software lack basic tools for efficient measurements of laboratory mice (e.g. segmentation and background subtraction, automatic identification of multiple animals, detection and removal of urine spots, automatic ROIs, batch analysis).

We thus developed *ThermoLabAnimal*, a computational tool capable of analyzing thermal images in .CSV file format, a standard output file used by several commercially available thermal cameras. The software addresses the analysis challenge in two ways. Firstly, through the segmentation algorithm, it automatically separates the pixels in the image that represent the animal from those that represent the background. This eliminates the need to manually defining the area of analysis, which is time-consuming and prone to inter-user variation. Importantly, instead of collapsing the thermal readout of the segmented pixels into a single statistical measure (such as a mean or median value), the method generates and uses the full histogram of the values to generate a more informative thermal fingerprint. Secondly, the batch analysis function allows automatic and high throughput analysis of all image files located in a folder. We tested the compatibility of the software with both the *Thermal Expert* camera used in this study (a miniaturized, low-cost thermal camera, a technology that is likely to improve and widen access to IRT technology (Clausing, 2016; Maillot et al., 2018) – and the most broadly used thermal camera manufacturer, *FLIR* (using .CSV files converted by its proprietary software). The software's compatibility can be extended to other manufacturers, but presently it already reads the standard .CSV format, with or without manufacturer-specific headers. A comparison between typical general-purpose IRT software and the proposed *ThermoLabAnimal* software is presented in Table 3.1.

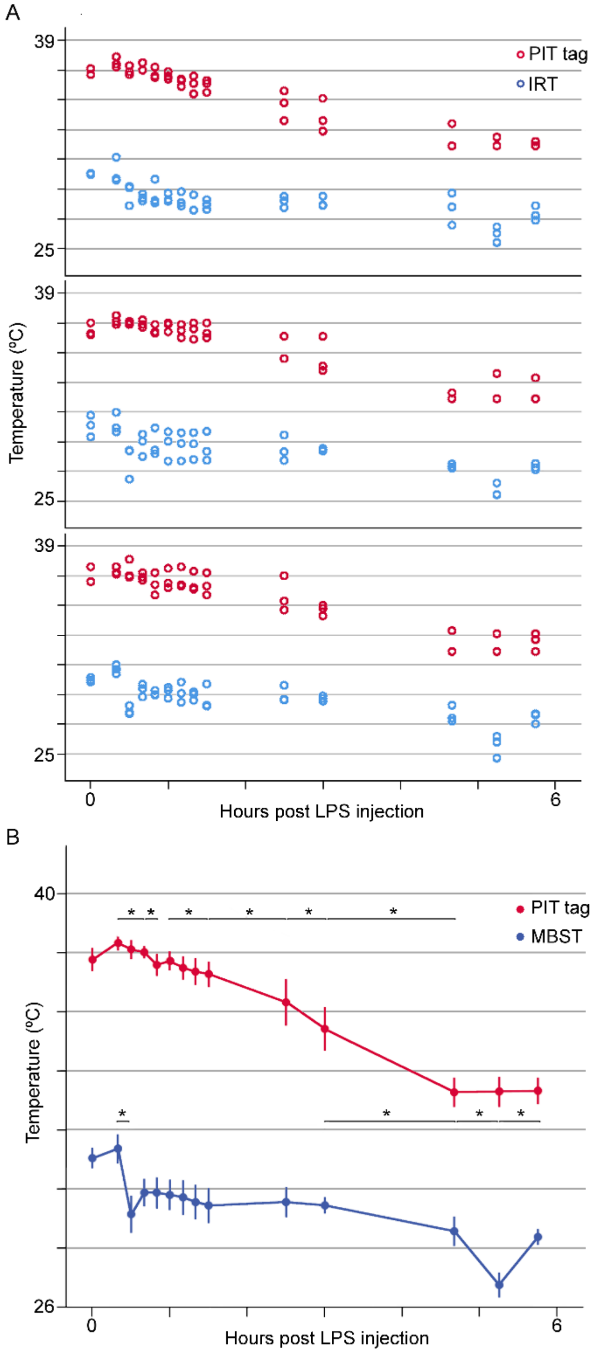


Figure 3.4 MBST and subcutaneous temperature following LPS injection. A. The temperature profiles following a high-dose LPS challenge were consistent in all three cages (treatment groups) and between PIT tags and MBST method. **B.** Temperature profiles combining (mean) the 3 groups of 3 mice (N = 9). The largest mean MBST variation (-4.6°C) was found between t = 0.33 h (20 min post-LPS injection) and t = 4.67 h (280 min post LPS-injection). The largest subcutaneous temperature variation (-5.4°C, between t = 0.33 h and t = 4.67 h) is an underestimate, since temperatures under 33.0°C were under the lower

reading range limit of the PIT tag reader, and thus scored as 32.9°C, for analysis. Asterisks (*) indicate significant changes ($\boldsymbol{p} < 0.01$) between consecutive time-points (paired samples t-test, with *Holm-Bonferroni* correction for multiple comparisons). Bars represent 95% confidence interval.

Our protocol and software for assessing MBST variation in laboratory animals have the potential to improve not only animal welfare but also the quality of the scientific results, by minimizing interference from handling stress (Bailoo, Reichlin, & Würbel, 2014; Gouveia & Hurst, 2013). While this is aligned with the 3Rs principle of Refinement, it can also help further the principle of Reduction, by reducing unwanted inter-individual variability – e.g. from handling stress, sensor position or operator bias – which warrants larger sample sizes for detecting a given effect (Bailoo et al., 2014; Parker & Browne, 2014). In this experiment, we present data from thermal images of animals placed in separate cages. We have now further developed the software to allow segmentation of each individual animal, so that future recordings can be done without having to remove an animal from its home cage and cage mates. The present version of the software is still unable to identify the same animal across multiple images, we are contemplating if it is feasible to add a feature to automatically identify animals, using the unique distribution of body surface temperature as an identifier (M. Mazur-Milecka, 2016), or adding a marker recognizable by the software.

Our results on a mouse model of sepsis-induced hypothermia show that the expected temperature decrease and hypothermia in mice in the first hours following a high-dose LPS injection (Blanqué, Meakin, Millet, & Gardner, 1996; Saito, Sherwood, Varma, & Evers, 2003) was identifiable by both readout of subcutaneous PIT tags and the automatic estimation of MBST. This decrease was, however, more immediately perceivable in the animals' body surface temperature, which to our knowledge has not previously been monitored for this model. This finding is consistent with the peripheral vasoconstriction expected to follow high LPS dose injection in mice (microcirculatory dysfunction is a central feature of sepsis pathogenesis (Bauer, 2002)), which would result in a subsequent temperature decrease in peripheral areas due to reduced blood flow. It might also be a physiological response mechanism to maintain core-body temperature (C. J. Gordon et al., 2008; Overton, 2010), or a combination of both.

Table 3.1 Comparison between general-purpose IRT software (e.g. *Fluke SmartView*, *FLIR Tools*, or *Testo IrSoft*) and *ThermoLabAnimal* software.

	Proprietary software	***ThermoLabAnimal***
Compatibility	Compatible with own manufacturer's export formats	*FLIR Systems* .CSV; *Thermo Expert* .CSV (expandable to other manufacturers' .CSVs)
Batch Analysis	No	Yes
Automatic identification of animals	No	Yes
Statistics over a region-of-interest	Yes (usually simple geometrical shapes)	Yes (freehand regions)
3D thermal map	No	Yes
Automatic animal body segmentation	No	Yes
Price	Variable	Free, for the version with all features except batch analysis

A transient small temperature rise was also observable by both methods (although not significant) immediately after intraperitoneal injection of LPS, which is consistent with a quick onset hyperthermic stress response, and in agreement with similar observations in rats (Almeida, Steiner, Branco, & Romanovsky, 2006). It could, however, also be attributed to an increase of activity following the disturbance of the animals and the removal of the nesting material, or to a putative short-lived fever response following the LPS challenge, immediately before the onset of hypothermia. We also observed a rise in skin temperature resulting from placing the cages over a warm heating pad, although how this affected subcutaneous temperature is uncertain, as PIT tags cannot register temperatures below 33°C.

It is worth mentioning that the easiness of image analysis brought about by this software allows for greater monitoring frequency, which can be an advantage for many animal studies, especially during short standardized behavioral tests (e.g. open-field (OF) test). For this particular study, however, in which more interventions other than collecting images – as this was coupled to another experiment – were carried out, this would mean making animals endure prolonged periods under a bright light (and warrant a cage with taller walls) or disturb their microenvironment more frequently.

We found it would not be appropriate to test for correlation between MBST and subcutaneous temperature, even though temperature decrease was observable by the two methods. Firstly, because the PIT tags have a hard lower bound and missed data below 33°C, which prevented identifying further temperature decreases from t = 4.67 h onward. Secondly, despite standardized implantation methods, we found great variation in PIT tag position (upon post-mortem examination, these could be found in some animals in the interscapular region, in others on the flank of the animal, and others on the backside), and such variation is a likely contributing factor to the unreliability of PIT tags to inform on core body temperature in vivo despite providing accurate readings in vitro (Hartinger et al., 2002). More accurate information on the relationship between this proxy measure of core body temperature and MBST could be sought in further studies, by fixing the position of the thermo-sensitive transponders, either on the dorsal or abdominal region (Kort, Hekking-Weijma, Tenkate, Sorm, & VanStrik, 1998).

Another possibility would be to compare MBST with maximum eye temperature, since it has been found to be a good proxy of other measures of core body temperature in mice (Vogel et al., 2016), as the eyes are supplied by blood from the ophthalmic artery from the brain (Vogel et al., 2016). However, the relationship between eye temperature and core temperature may be easily affected by peripheral vasoconstriction, either elicited exposure to cold (Piccione, Gianesella, Morgante, & Refinetti, 2013; Vannetti et al., 2014) or acute stressors (Herborn et al., 2015; Ludwig et al., 2010; D. M. Vianna & Carrive, 2005). Moreover, we have found it not to be well suited for measuring freely-moving mice, as constant shifts in head position will affect the thermal readout from the eyes. Furthermore, the eyes are often not visible, leading to up to 40% missing values, even when taking three images per time-point (Gjendal et al., 2018). It is, therefore, not surprising that others have recommended picking up and restraining mice from placing their eyes directly in front of the thermal camera for estimating absolute temperature by IRT (Vogel et al., 2016), losing the advantages of contactless measurement.

Perhaps most importantly, there is no ‘pure' measure of body temperature, just local temperatures in different parts of the body – internal or external – each of them with its particular bias (D. L. Vianna & Carrive, 2012). Therefore, although in small animals, skin temperature is more likely connected with core temperature than in larger species (McCafferty et al., 2015), we do not wish to make the claim that MBST can be used as a proxy of mice core body temperature. We are insteade proposing that MBST – obtained by means of our software – can be a reliable, non-invasive approach to identify and monitor temperature changes in freely-moving group-housed mice with minimal disturbance. It is, however, worth mentioning that gathering species-specific thermographic data could contribute to defining standard values for MBST that could be used as a reference value in and of itself.

ThermoLabAnimal is distributed as free software with all its features except for the "Batch Analysis" mode, which is available at a small fee. The paid full version does not require *MATLAB* on the user side, and the fee is intended to promote long-term support and improvement of the software. We encourage colleagues to use the software and provide feedback to promote improvements and bug fixes.

ThermoLabAnimal software is available on GitHub: https://github.com/ThermoLabAnimal.

7. Acknowledgments

This work was partially funded by National Funds through FCT – Fundação para a Ciência e a Tecnologia under the project SRecognite Infect-ERA/0003/2015 that also funds Liliana Oliveira. Equipment was partially funded by a UFAW Small Travel and Project Award for Nuno Franco (UFAW Grant 25-2016/2017). Funding also by FEDER - Fundo Europeu de Desenvolvimento Regional funds through the COMPETE 2020 - Operacional Programme for Competitiveness and Internationalisation (POCI), Portugal 2020, and by Portuguese funds through FCT - Fundação para a Ciência e a Tecnologia/Ministério da Ciência, Tecnologia e Ensino Superior in the framework of the project “Institute for Research and Innovation in Health Sciences” (POCI-01-0145-FEDER-007274). Ana Gerós is funded by FCT – Fundacão para a Ciência e a Tecnologia, grant contract SFRH/BD/137385/2018.

We would like to thank Joaquim Gabriel Mendes and Ricardo Vardasca, at the University of Porto Faculty of Engineering, for carrying out the tests with a blackbody radiation source.

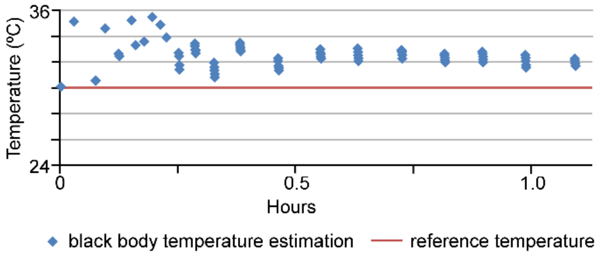
8. Supplementary Information

8.1 Extended methodology

8.1.1 Thermal reading validation experiments

The accuracy of the low-cost *Thermal Expert* TE-Q1 camera was tested using a blackbody radiation source of 1.00 emissivity (*Hyperion R Blackbody* Model 982, Blackbody Isotech, Southport, UK) at a stable controlled temperature of 30°C, in equally controlled room conditions. Thermal pictures were collected of the central circular body placed at a distance of 30 cm (with the circular body occupying 2/3 of the frame), at 1 min intervals for the first 15 mins and 5 min intervals for the following 45 min. Three to four images per time-point were taken. The *ThermoLabAnimal* software was used to segment and perform statistical analysis (full distribution, mean and median values of the readings) of the temperature at the blackbody. Camera measurements were compared to the specified blackbody temperature to perform an independent assessment of the quality (precision, accuracy and stability) of the *Thermal Expert* TE-Q1 measurements. This assessment showed that the camera requires a warm-up period of at least 30 mins for the thermal readings to begin stabilizing (Supplementary Figure S 3.1). The warm-up time is a common condition in thermal cameras (which require hardware stabilization before accurate measurements) and is critical to take into account in thermal imaging (Priego Quesada, Kunzler, & Carpes, 2017; Vogel et al., 2016). In the controlled blackbody experiments, after 30 min warm-up time, our particular camera registered values at +2.6°C (+/- 0.26 SD), and at 60 mins, it further stabilized at +1.96 °C (+/- 0.19 SD) above the controlled blackbody temperature. Differences in mean temperature between some consecutive time-points were found to be significant (Supplementary Table 3.1), albeit small (at most 0.35°C) (Supplementary Figure S 3.1).

8.2 Supplementary Figures



Supplementary Figure S 3.1 In thermal imaging, all cameras, both low-cost and high-end, typically require a warmup period. The low-cost *Thermal Expert* TE-Q1 thermal camera used in this study required about 30 minutes to reduce significantly its readings variability.

8.3 Supplementary Tables

Supplementary Table 3.1 Differences in mean temperature to blackbody reference temperature. Readings were taken in intervals of typically 5 minutes, after waiting for 30 minutes of warmup (m30).

	Mean	Std.	95% Confidence Interval of the Difference				
	difference	**Deviation**	**Lower**	**Upper**	***t***	***DF***	***p***
m30 - m35	0.064	0.108	-0.070	0.198	1.324	4	0.256
m35 - m40	-0.132	0.156	-0.326	0.062	-1.889	4	0.132
m40 - m46	0.348	0.075	0.255	0.440	10.43	4	0.000
m46 - m50	-0.090	0.069	-0.176	-0.004	-2.905	4	0.044
m50 - m55	0.346	0.078	0.250	0.442	9.963	4	0.001
m55 - m61	0.092	0.198	-0.153	0.337	1.041	4	0.356

CHAPTER 4

Improved 3D Tracking and Automated Classification of Rodents’ Behavioral Activity using Depth-Sensing Cameras

This chapter was based on the following original research paper:

Gerós, A., Magalhães, A., & Aguiar, P. (2020). Improved 3D tracking and automated classification of rodents' behavioral activity using depth-sensing cameras. Behavior research methods, 52(5), 2156-2167

1. Abstract

Analysis of rodents' behavior/activity is of fundamental importance in many research fields. However, many behavioral experiments still rely on manual scoring, with obvious problems in reproducibility. Despite important advances in video-analysis systems and computational ethology, automatic behavior quantification is still a challenge. The need for large training datasets, background stability requirements, and reduction to 2-dimensional analysis (impairing full posture characterization), limit their use. Here we present a novel integrated solution for behavioral analysis of individual rats, combining video segmentation, tracking of body parts, and automatic classification of behaviors, using machine learning and computer vision methods. Low-cost depth cameras (RGB-D) are used to enable 3-dimensional tracking and classification in dark conditions and absence of color contrast. Natively our solution tracks five anatomical landmarks in dynamic environments and recognizes seven distinct behaviors, within the accuracy range of human annotations. The developed free software was validated in experiments where behavioral differences between Wistar Kyoto and Wistar rats were automatically quantified. The results reveal the capability for effective automatic phenotyping. An extended annotated RGB-D dataset is also made publicly available. The proposed solution is an easy-to-use tool, with low-cost setup and powerful 3D segmentation methods (in static/dynamic environments). The ability to work in dark conditions means that animal natural behavior is not affected by recording lights. Furthermore, it is capable of automatic classification with only ~30 minutes of annotated videos. By creating conditions for high-throughput analysis and reproducible quantitative measurements of animal behavior experiments, we believe this contribution can greatly improve behavioral analysis research.

Keywords: Animal Tracking in 3D • Automatic Behavior Classification • Automatic Phenotyping • Depth Sensors • Dynamic Background Segmentation • Free and User-Friendly Software • Public RGB-D Dataset • Wistar Kyoto Model

2. Introduction

Analysis of how animals interact with, respond to, and control their environment, is a fundamental methodological approach in many research fields (Anderson & Perona, 2014;

Berman, 2018). This is particularly relevant in behavioral neuroscience and in the challenge to understand brain function (Dickinson et al., 2000; Hong et al., 2015; Krakauer, Ghazanfar, Gomez-Marin, MacIver, & Poeppel, 2017). Besides being a pillar in the health sciences, supporting research translation to human clinical trials (Richardson, 2015; Unger et al., 2017), animal behavior analysis is an increasingly important tool in industry, namely in the essential animal welfare monitoring in food production (Ahrendt et al., 2011; Hong et al., 2015; Stavrakakis et al., 2015).

A full characterization of phenotypic domains in behavioral analysis asks for screening test batteries, with different degrees of coverage and validation, and implemented in a non-subjective and standardized way. Computerized video-analysis systems have thus emerged as potential tools to automatically assess behavior, combining 2-dimensional (2D) video recordings with image processing (Robie et al., 2017; Valletta, Torney, Kings, Thornton, & Madden, 2017) and machine learning methods (P. Aguiar et al., 2007; de Chaumont et al., 2012; Jhuang et al., 2010; Preisig et al., 2016). Most published solutions rely on standard background subtraction methods (P. Aguiar et al., 2007; Jhuang et al., 2010; Twining et al., 2001) for animal segmentation, with dynamic background conditions being still under active development. Body-parts classification can be addressed using algorithms for learning/computing the individual's pose (A. Mathis et al., 2018; T. D. Pereira et al., 2019). In turn, trajectory-based features (Burgos-Artizzu et al., 2012; Kabra et al., 2013) can be extracted from video sequences (Dollár et al., 2005; Jhuang et al., 2010) to describe low-level representations of behavior. These features can then be used for automatic behavior classification by applying rule-based classifiers (de Chaumont et al., 2012), or supervised (Burgos-Artizzu et al., 2012; Kabra et al., 2013) and unsupervised (Berman et al., 2014; Schwarz et al., 2015) machine learning methods to train classifiers. Alternatively, semi- and weakly-supervised learning may be introduced in this context although modest progress has been made here (Egnor & Branson, 2016; Malte Lorbach, Poppe, & Veltkamp, 2019; Robie et al., 2017).

Nevertheless, as expected, the estimation of animals’ pose in 2D is unsatisfactory in most cases. Some studies have therefore started to address the problem in 3-dimensions (3D), using multiple conventional cameras, or cameras capable of combining color and depth sensing (RGB-D cameras) (Hong et al., 2015; Matsumoto et al., 2013; Z. Wang et al., 2018).

The present study describes a novel computational solution for automated, markerless 3D segmentation and tracking (in static and dynamic environments), of both whole-body and body parts, in experiments with a single freely behaving rodent. This tool uses low-cost RGB-D sensors and machine learning/computer vision techniques, to precisely quantify behavioral features in 3D space. Given its focus on automatic Classification and Tracking in depth (z-axis), our computational tool is named *CaT-z*. The tool is tested and validated in controlled experiments to assess its performance and precision. It is made freely available to the research community, as to foster reproducible and reliable quantitative behavioral analysis in labs with limited resources.

The *CaT-z* software is publically available for download at GitHub: https://github.com/CaT-zTools/CaT-z_Software.The open access dataset (41 GB) is also publicly available for download at *Zenodo*: https://zenodo.org/record/3636136#.YbysrGjP2Uk.

3. Materials and Methods

3.1 Behavioral Protocol

Behavioral experiments for dataset construction and system validation were conducted during 3 consecutive weeks for each animal (N = 2). Inside the experimental environment (an opaque acrylic open-field (OF) cage, 1 m × 1 m × 0.5 m, made in-house), 3 types of light conditions where alternatively used: dim red light, dim white light, and in total darkness (Figure 4.1). Animals were recorded using *CaT-z* software, while freely moving, for 15 minutes (mins). For behavioral phenotyping studies, Wistar Kyoto rats (WKY; N = 10) and wild type rats (N = 10) were subjected to the Elevated Plus Maze (EPM) test (standard apparatus). Animals were allowed to freely explore the maze for 5 mins. The following measurements were taken: percentage time spent in the open arms, percentage time spent in center arena and total distance, as well as automatic classification of seven behaviors (see below).

3.2 Video Acquisition

RGB-D videos were recorded using a *Microsoft Kinect v2* camera, with 1920x1080 color and 512x424 depth pixels' resolution, respectively. It records at a maximum of 30 frames per second (fps), but in low light conditions this value drops to 15 fps (typically). The operation range is from 0.5 to 4.5 m, with a spatial resolution of ≈2 mm. The camera was placed centrally above the OF and the EPM (1.20 m high, to fully include setup dimensions) and connected to a computer. A pre-heating time of 30 mins for the camera was respected for the stabilization of the depth sensor (E. Lachat, Macher, Landes, & Grussenmeyer, 2015).

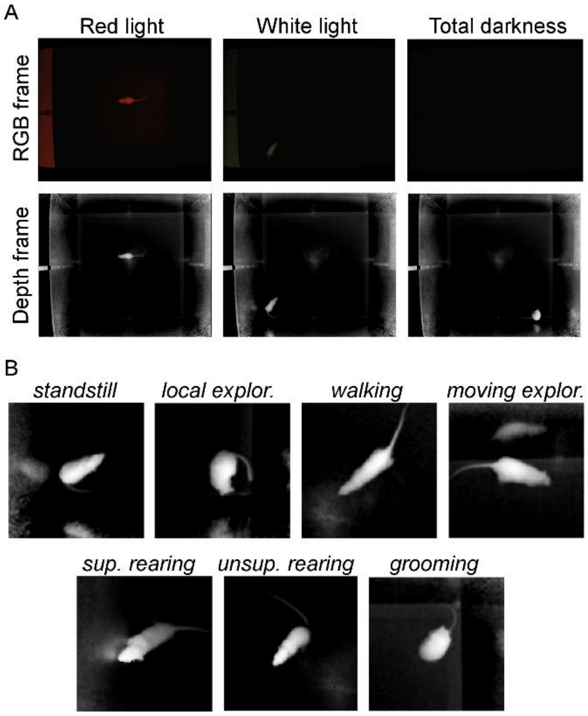


Figure 4.1 RGB-D behavioral dataset. A. RGB and depth frames under three different lighting conditions: dim red light, dim white light, and in total darkness. **B.** Depth frames for the seven types of rodent behaviors.

3.3 Manual annotation of rodents’ behaviors

The RGB-D dataset containing frames for supervised classification (ground-truth) was fully annotated by researchers with experience in ethology, with one of seven mutually exclusive behavioral labels: *standstill*, *local exploration*, *moving exploration*, *walking*, *supported* and *unsupported rearing*, and *grooming* (Supplementary Table 4.1; see Figure 4.1 for examples). An extended list of classes is sometimes not necessary, nor advisable (increase in subjectivity), and consequently a simplified list was also considered: *standstill*+ (*standstill* and *local exploration*), *walking*+ (*walking* and *moving exploration*), *rearing* (*unsupported* and *supported rearing*), and *grooming*. The *CaT-z* software also includes an interface for manual annotation, which was used for the manually annotated dataset for the supervised classification algorithms ("ground-truth"). Regarding the observation method, the annotation interface allows the construction of the animals ethogram based on focal-animal annotations, and all actions of one animal are annotated for a specified time period (all video frames are annotated).

The level of agreement between observers for the annotated dataset was calculated using two different metrics. In the frame-based approach, 1 frame tolerance was allowed in the transitions. In the quality-based approach, the number of matching (overlapping) behavior periods between observers was used.

For the WKY/Wistar EPM experiments, seven mutually exclusive behaviors were also defined: *standstill*+ (*local exploration* and *standstill*), *walking*+ (*walking* and *moving exploration*), *rearing* (*supported* and *unsupported rearing*), *head dipping* (snout sloping down from the EPM and body standing in the same place with the 4 legs in the open arms), *protective head dipping* (snout sloping down from the EPM and body standing in the same place with at least one limb in the closed arms), *SAP* (lower back, elongation of the body, and either standing still or moving forward very slowly), and *grooming* (see Supplementary Table 4.1 for definitions).

3.4 Tracking and Classification Algorithms

Four computational components are addressed in our method (Supplementary Figure S 4.1): animal segmentation, tracking, features detection and classification. All algorithms were implemented in *C*++ language, for computational performance, and using *Qt Creator* (The Qt Company Ltd.) environment to integrate the algorithms in the user-friendly *CaT-z* software. Three graphical user interfaces (GUIs) were developed to: support videos acquisition, annotation and processing (segmentation, tracking, and classification of behavioral data).

3.4.1 Animal detection and tracking.

Animal segmentation was performed using three different background modeling methods. The static Median-difference method sets a static background model using the median of the pixels over a set of initial frames. A 2D median filter (5x5 size) was also applied. Along the frames, the foreground detection was performed by computing the difference between the current frame and the background model.

In order to cope with dynamically changing environments (e.g. bedding material, small objects moving/(dis)appearing), two other algorithms were developed. Both methods are initialized with a background model similar to the static method. The landscape-change detection (LCD) method uses the background subtraction technique but constantly updates the background model. The updating algorithm uses the assumption that local environment modifications are smaller than the animal's area. The background model is updated using information from the current frame to incorporate possible objects that (dis)appeared/moved in the frame. Finally, the probabilistic Gaussian Mixture model (GMM), was adapted from Stauffer and Grimson (1999), to incorporate 16-bit depth images in the processing algorithm and improve background estimation.

The validation of these methods under dynamic environments was performed using a controlled synthetic dataset. This dataset consisted on 1000 depth frames, whose intensity values follow a normal distribution of mean 1000 mm and standard deviation 5 mm (experimental precision value of this depth sensor). A dynamic environment was simulated by synthetically creating well-defined dips or rises in the depth map. The validation was performed by comparing background models and ground truth.

3.4.2 Body parts' detection and tracking.

From the 3D segmented animal, five anatomical points were tracked: nose, head, body center (centroid), tail-base and tail-end. Importantly, these landmarks were estimated using scale-free geometrical constraints/properties (see Body parts' detection and tracking). For example, after finding the rodent body contours, the tail-end is defined as the furthest contour point from the centroid (independently of animal size). Simple heuristics were implemented to check the validity of the detected body parts location (for example, discrepancy between the positions in consecutive frames). Frames with uncertain body parts' detection are flagged and this information is later used for the frame classification (see Supplementary Table 4.2): not only this flag in important for signaling tracking anomalies, but also, interestingly, the absence of particular body parts (e.g. by occlusions) can in itself help detecting certain behaviors (for example, during *grooming* events frequently the nose is not detected).

The performance of the body parts' detection algorithm (which relies on scale-free geometrical rules) was evaluated by comparing the automated tracking results with manually annotated locations of body parts in a set of 600+ frames.

3.4.3 Features' extraction.

For the automatic classifiers, low-level representations of behavior were organized to describe trajectory-based aspects and shape-based information (Supplementary Table 4.2). In order to add information from previous frames (temporal memory) and to help distinguish between behaviors with different temporal dynamics, the features' set for each frame were combined with the features from ~1 second in the past, obtaining a final set of 22 features for each time point. The features were normalized using Z-score transformation.

3.4.4 Automatic behavior classification.

The Support Vector Machine (SVM) classifier was selected for supervised and multi-class behavior recognition (Boser, Guyon, & Vapnik, 1992). A nonlinear classifier with a radial basis function kernel was used. Further theoretical concepts on SVMs are available in Appendix – Fundamentals of Support Vector Machines. The best combination of SVM parameters was selected by grid search and the parameters with higher cross-validation accuracy were selected, using k-fold cross-validation approach (k = 5) on the training set. Performance was estimated using the leave-one-video-out technique, where all but one video of a pool of N videos were used to train the model, and the performance was evaluated on the remaining video. This procedure was repeated N times for all videos. Learning curves were constructed to show the classification performance as a function of the training dataset size, and to determine the minimum N size to construct this pool of videos.

Model predictions for all the testing frames were filtered (with a 5x5 median filter) to reduce erroneous classifications of isolated frames, and then concatenated to compute the overall accuracy (ratio of correct frames), and performance per class using confusion matrices and the F1-score.

F1-score is the harmonic average of the precision and recall, ranging from 0, with no correct predictions, to 1 for perfect precision and recall, calculated as follows:

$$F1\ score = 2 \times \frac{precision\ \times recall}{precision + recall} \tag{4.1}$$

where $precision = \frac{true\ positive}{(true\ positive + false\ positive)}$ and $recall = \frac{true\ positive}{true\ positive + false\ negative}$

This metric is better suited for datasets with behaviors that occur with different frequencies (M. Lorbach et al., 2018). This leave-one-video-out approach provides the best estimate of the future performance of a classifier, and was also applied to avoid testing bias due to the consecutive frames effect and “double-dipping” (Kriegeskorte, Simmons, Bellgowan, & Baker, 2009).

When studying the activity of WKY rats inside the EPM, only RGB-D data from Wistar rats was used to train the classifier, but both Wistar and WKY data was used as testing sets.

3.5 Behavioral phenotyping

The capability to detect behavioral differences (phenotyping) between different strains was assessed using a k-nearest neighbor algorithm (kNN). This choice served the purpose of demonstrating that even a simple classifier can be used for this step. Model’s accuracy and posterior probabilities of belonging to the control class were calculated for both Wistar and WKY strains in order to select a reduced set of metrics and to construct a behavioral profile for phenotyping strains.

Extended methodology is presented in Supplementary Information.

4. Results

4.1 An RGB-D behavioral dataset to promote advances in computational ethology

As in other fields, important contributions to computational ethology can potentially arise from machine learning researchers not directly engaged in behavioral experiments. The availability of large, public, annotated datasets is therefore of fundamental importance to empower these potential contributions. With this in mind, instead of producing a specific dataset for developing *CaT-z*, we have compiled a general-purpose dataset, which is made public to catalyze new developments in computational ethology and automatic classification of rat behavior activity.

The produced RGB-D dataset was compiled from videos and respective behavior annotations that capture freely-walking Wistar rats in an OF arena. The OF was chosen since it is a standard setup commonly used in ethology studies to measure behavioral and locomotor activity in animal models (Belzung, 1999; Cryan & Holmes, 2005; Overstreet, 2012). The dataset is composed of several ≈10/15 mins RGB-D video sequences of individual rat behavior, where the animal is allowed to move freely inside the OF cage (Figure 4.1). Three different lighting conditions were used (Figure 4.1A) to recreate the typical light setups used in behavioral recordings. Total darkness is the ideal lighting condition for the animals' active phase, but it is usually replaced by dim red light or dim white light due to limitations of the standard recording systems. The full dataset consists of 24 videos, with a total of 6 hours: 4 hours of fully annotated sequences (for supervised machine learning methods; ~180 000 annotated frames) and 2 additional hours of raw behavioral sequences (adding data for unsupervised machine learning methods).

Every RGB-D video frame in the annotated dataset was manually labeled by researchers with experience in ethology, with one of the seven mutually exclusive rat behavioral labels (Figure 4.1). These specific behaviors were selected as they are commonly used in manual scorings in neurobehavioral research. Information regarding the frequency of each behavioral event, within the annotated dataset, is described in Supplementary Table 4.1, which ranges from 2.5%, for *walking* events, to 37.9% for *local exploration* events.

In the manual annotation of animal behavior, reliability between human observers is typically limited to 70%-80% (Anderson & Perona, 2014; Spruijt & DeVisser, 2006). This limitation was, in fact, a core motivation for this work. In the annotated dataset the average level of agreement between the observers was 83.3% ± 5.7 in a frame-based approach (N = 21 988 frames), and 81% ± 0.8 in a quality-based approach (agreement on behavioral type; please see Materials and Methods). Taken together, these results reveal that both agreement scores for the annotation of this dataset are consistent with the reported range.

4.2 Depth information improves whole-body segmentation in both static and dynamic background conditions

Animal segmentation, a challenging problem in RGB video sequences, is considerably improved and facilitated using depth information combined with the implemented static/dynamic background algorithms (Figure 4.2). In the segmented images, it is possible to visually distinguish specific body parts such as tail, snout, upper and lower limbs (Figure 4.2A). For different lighting conditions, there were no differences in detection performance, which means that animal detection is independent of ambient lighting.

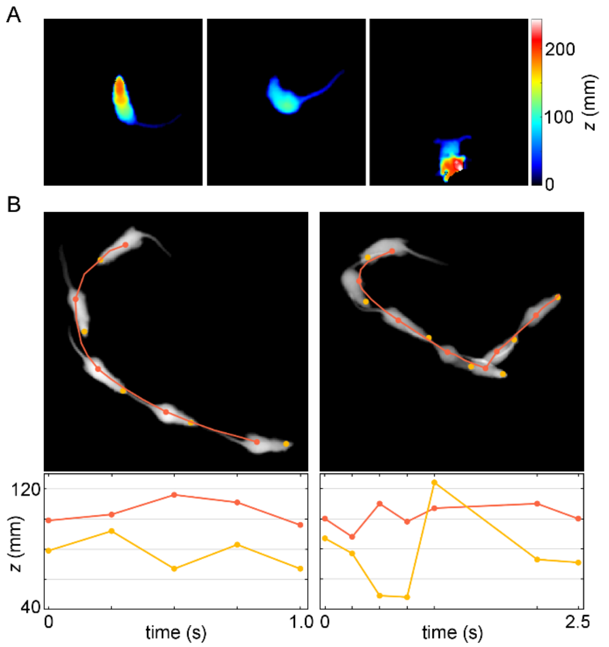


Figure 4.2 Depth information improves whole-body segmentation. Segmented depth frames, using the static Median-difference method for background removal, capturing: **A.** Three different behaviors: *unsupported rearing* (left), *local exploration* (middle), and *supported rearing* (right). Background pixels in black. Maximum depth values (240mm) in white. Depth colormap in mm. **B.** Body parts' tracking (centroid and nose) and their depth profile. Top: Two sequences of segmented depth frames with identification of some body parts: centroid (in orange) and head (in yellow). Bottom: Corresponding depth profile (in mm) for the centroid and head points in the depth frames sequences.

The performance of the three background segmentation algorithms (standard static, modified GMM, and the new LCD algorithm) was quantified in controlled dynamic background landscapes (Supplementary Figure S 4.2). The results showed that the LCD method is more effective at dealing with background changes, incorporating them quickly into its depth profile: as the background changes, the pixel depth values change instantly, allowing a more accurate estimate of the background. In turn, the modified GMMs method also incorporates pixel modifications in the estimated background but much slower than the LCD method, which is consistent with the defined learning rate. As expected, the widely used static Median-difference method has very limited performance in dynamic environments.

4.3 Tracking multiple anatomical landmarks in 3D

Geometric methods for the detection of body parts greatly benefit from depth information, enabling the detection of the 3D trajectories of each anatomical landmark. Using these representations, it is possible to identify subtle fluctuations in depth which could not be noticeable by visual inspection (Figure 4.2B).

Overall tracking performance was assessed by comparing automatically predicted coordinates with the manually labeled ones (Figure 4.3). In particular, automatically detected positions of the animal's body center are in very high agreement with the carefully manually-traced trajectories (Figure 4.3A). The trajectories overlap along the frames, with a 5% error in the estimation of the distance traveled by the animal. The errors in estimating the traveled distance can be driven by differences between the visual estimate of the animal's body center and the centroid mathematical estimate, which is affected by other segmented body parts (e.g., tail). For each labeled frame, the x- and y-coordinates' differences between predicted and manually defined body center coordinates were computed for error quantification (Figure 4.3B). The differences were, most of the cases (median), less than 2 pixels (Figure 4.3B). In fact, a difference of 2 pixels, between the predicted and manually labeled body center coordinates, is barely noticeable and within the variability of human annotation (Figure 4.3C).

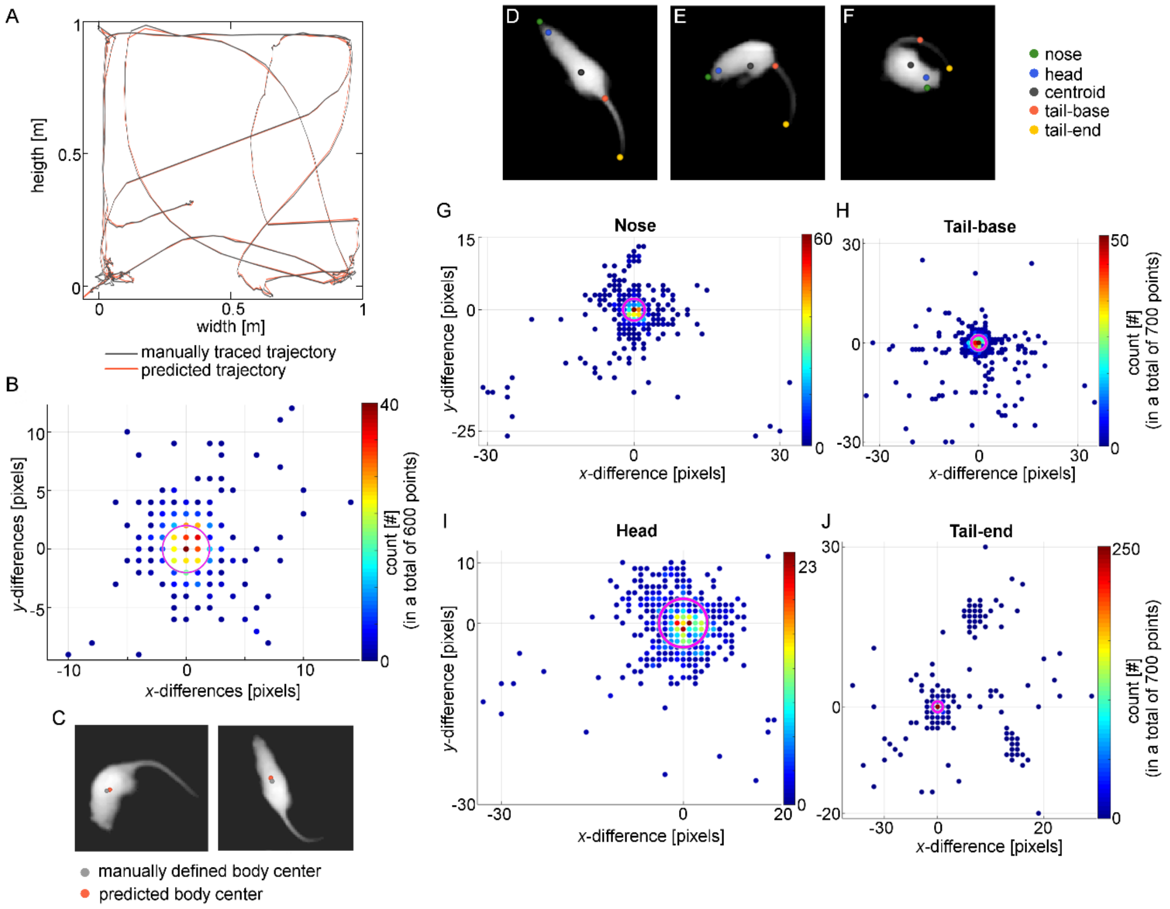


Figure 4.3 Multiple anatomical landmarks can be accurately tracked in 3D. A–C. Comparison between manually defined body center and automatically predicted coordinates, for a 40 seconds' frame sequence. **A.** Manually-traced (gray) and predicted (orange) trajectories inside the open-field cage. **B.** x- and y-coordinates differences, in pixels, between manually defined and predicted centroid's coordinates. Colorbar indicates x- and y-differences' occurrences. The circle in magenta (2 pixels radius) represents 50% of the results. **C.** Example images with manually defined body center (gray) and predicted (orange) coordinates, where the distance is equal to the median value (2 pixels). **D–F.** Examples of body parts' detection in several frames of a single video. **F.** shows an example of incorrect detection of tail-base and nose body parts. **G–J.** Histograms of coordinates' differences, in pixels, between manually defined and predicted body parts' coordinates, for a 46 seconds' frame sequence. Colorbar indicates x- and y-differences' occurrences. The circle, in magenta, represents 50% of the results, whose distance radius is 1.0, 2.1, 2.2 and 4.0 pixels, for G–J histograms, respectively. Scale factor calculated using open-field setup dimensions (scale factor = 3.2 mm/pixel).

The system is also able to automatically locate the position of landmarks for a variety of animal postures (Figure 4.3D and E). Nevertheless, when the animal is at ambiguous poses, the performance is reduced (Figure 4.3F). Globally, the performance of the system is very high, with the majority (median) of the landmarks detection errors being below 2 pixels for the nose and tail-base anatomical points, 4 pixels for the head estimate, and 1 pixel for the tail-end point detection (Figure 4.3G–J). The geometric algorithms defined to find the anatomical points are scale-free, making the tracking system robust to changes in animal sizes.

4.4 Automatic behavior classification using depth information

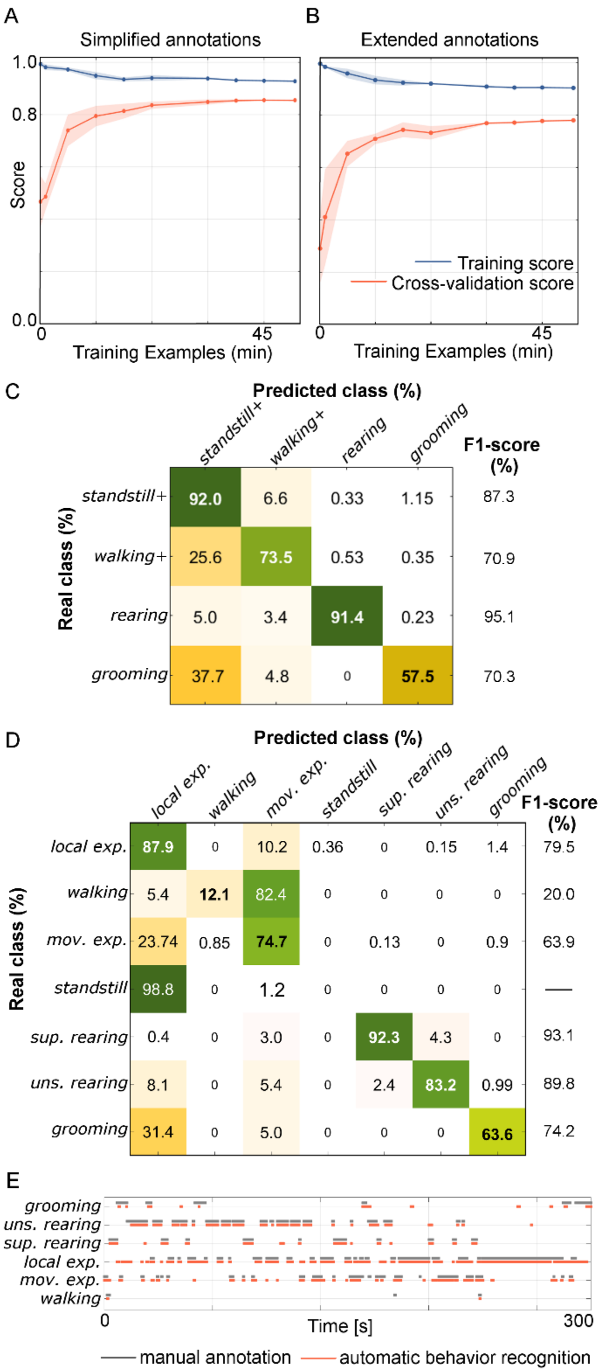
The proposed automatic classification system, based on multiclass SVMs, shows the capability to attain high-performance levels even if trained with only 30 mins of annotated video recordings (Figure 4.4). As the number of training examples increases, the mean gap between the validation and training scores gets narrower, and from a training set size of 30 000 examples (≈30 mins. video), both scores stabilize. This level of performance is observed using either simplified or extended annotations, corresponding to either 4 or 7 different types of behaviors (Figure 4.4A and B, respectively). The performance levels were assessed using a 5-fold cross-validation approach and avoiding testing bias problems (see Materials and Methods - Automatic behavior classification.). The 30 mins is an important figure, as compared with the very large training datasets required by other approaches, particularly conventional deep learning methods. It means that the manual annotation effort may be remarkably reduced in supervised training approaches. For consistency, the results presented from here on were all obtained with training datasets with roughly 30 mins of video.

Standard methods for automatic behavior analysis (Ethovision - Noldus, The Netherlands; Smart - Panlab, Spain; Kabra et al., 2013) are not fully functional under total dark conditions, which is an important limiting factor for recording natural rodent behavior. Our methods are independent of ambient light conditions (dim red, dim white, and total darkness) as shown by the automatic classification accuracy and F1-scores (Supplementary Table 4.3). Moreover, the system generalizes among different lighting conditions; for example, dim red light videos can be used for training and total darkness for testing (Supplementary Table 4.3).

For a detailed analysis of the classification errors for each type of behavior, we constructed confusion matrices, showing the combinations of predicted and real/annotated values (examples in Figure 4.4C and D). For the simplified annotations (4 classes), the average accuracy was 84.9%, with high F1-score values for all behaviors (Figure 4.4C), whereas in the extended annotations (7 classes) the average accuracy was 76.9%. In both conditions, the presently defined features for the SVM classifier allows the system to correctly recognize most behaviors (Supplementary Movie 1). In the extended annotations, the current system shows some limitations. *Walking* periods belong to the most misclassified behaviors, occasionally classified as *moving exploration*, leading to low F1-scores. Also, F1-scores for *standstill* are very low, or not possible to calculate due to lack of representativeness in the training set. The automatic classification methods presented here allow the direct generation of ethograms to describe the behavioral data, and the time spent on each behavior (Figure 4.4E).

4.5 *CaT-z*: a user-friendly computational solution for quantifying animal behavior

Acknowledging the paramount importance of encapsulating all algorithms in a user-friendly application suited for laboratory environments, an effort was made to create an integrated, easy-to-use, and freely available software that works out-of-the-shelf – *CaT-z*. This computational tool contains three different modules to support annotation and recording of RGB-D frames, and automatic tracking and classification of rodent’s behavior (Figure 4.5). The GUI for RGB-D data visualization and annotation (Figure 4.5A) allows the manual scoring of color and depth frames, simultaneously, into user-defined behaviors. Depth frames can be displayed in three different visualizations, and RGB-D videos can be played using media controls (in different velocities). During annotation, a behavioral ethogram is automatically updated to give color feedback on the behaviors previously identified. It is possible to resume an unfinished annotation and, finally, the data is saved in comma-separated values (.CSV) format to be later used for automatic behavior analysis. As far as we know, such RGB-D data annotation tools are not presently available.



Real class (%)	standstill+	walking+	rearing	grooming	F1-score (%)
standstill+	92.0	6.6	0.33	1.15	87.3
walking+	25.6	73.5	0.53	0.35	70.9
rearing	5.0	3.4	91.4	0.23	95.1
grooming	37.7	4.8	0	57.5	70.3

Real class (%)	local exp.	walking	mov. exp.	standstill	sup. rearing	uns. rearing	grooming	F1-score (%)
local exp.	87.9	0	10.2	0.36	0	0.15	1.4	79.5
walking	5.4	12.1	82.4	0	0	0	0	20.0
mov. exp.	23.74	0.85	74.7	0	0.13	0	0.9	63.9
standstill	98.8	0	1.2	0	0	0	0	—
sup. rearing	0.4	0	3.0	0	92.3	4.3	0	93.1
uns. rearing	8.1	0	5.4	0	2.4	83.2	0.99	89.8
grooming	31.4	0	5.0	0	0	0	63.6	74.2

Figure 4.4 Automatic behavioral recognition performance. A. and **B**. Learning curves of trained model for the recognition of 4 (simplified annotations) or 7 (extended annotations) behaviors, respectively. Results represented as mean (filled line) and SD (colored shadow) for training (blue) and cross-validation (orange) scores. **C.** and **D.** Examples of normalized confusion matrix of automatic behavioral recognition and corresponding F1-scores, for 4 or

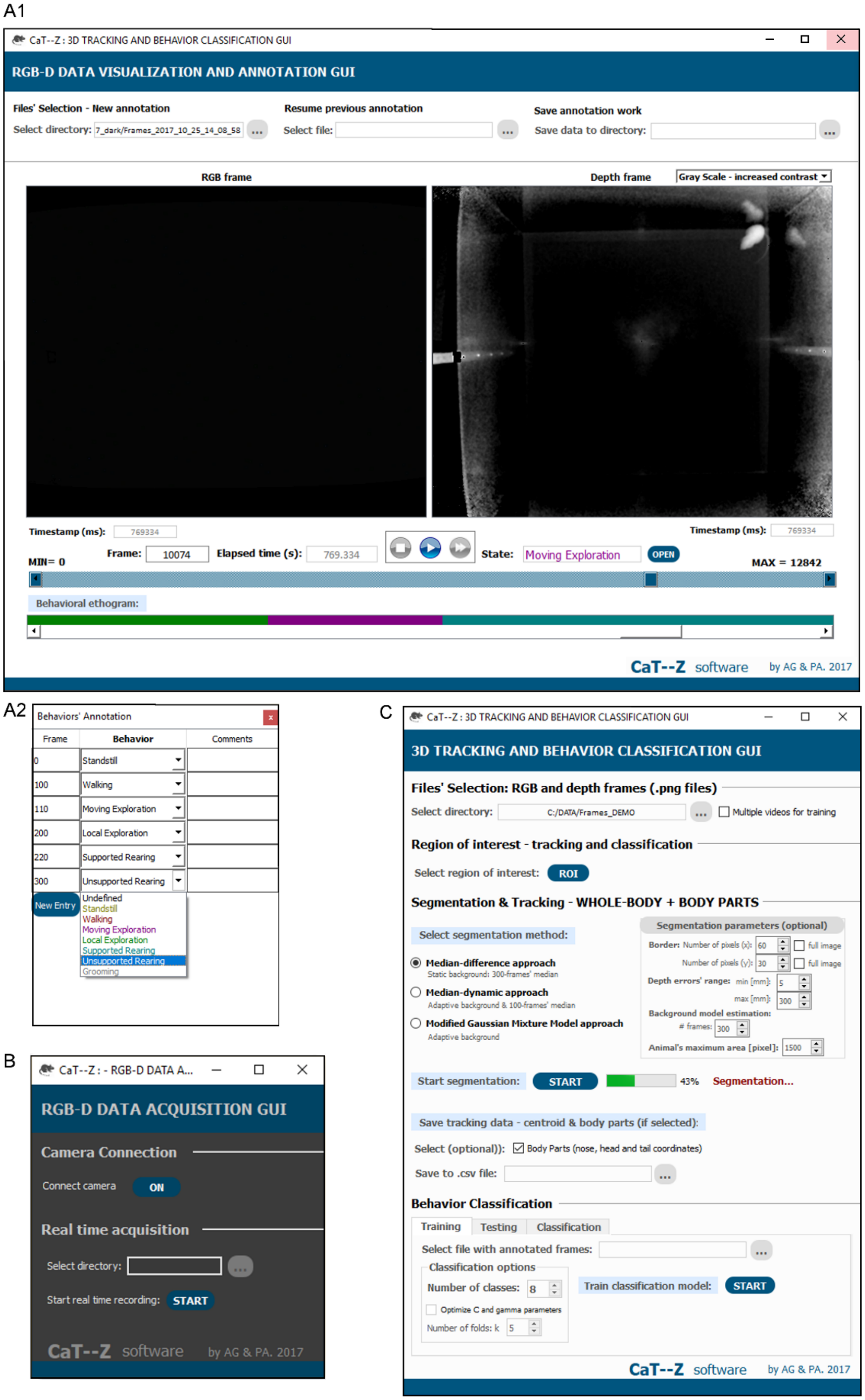
7 classes, respectively. **E.** Example of ethogram for manual annotations (gray) and automatic behavioral recognition labels (orange), over 300 seconds of testing video.

New RGB-D data can be acquired using the data acquisition GUI (Figure 4.5B), and later annotated or analyzed by the tracking/behavior classification GUI (Figure 4.5C). Segmentation and tracking are performed using different available methods, and a particular region-of-interest (ROI) can be select. Body part's tracking information (x, y and z coordinates) can be exported to a user-defined directory. Finally, using previous tracking information and annotated data, the classifier can be trained, tested, or applied for the recognition of new behavioral data. The GUI also allows training the classifier with multiple videos, simultaneously, without the need for multiple launches. Noteworthy, *CaT-z* is made available to the community with a detailed user manual and tutorial/walkthrough videos.

4.6 Ability to distinguish between strains – automatic behavioral phenotyping

The behavioral profile of WKY rats was quantitatively compared with Wistar rats using *CaT-z*. The system was capable of automatically detecting behavioral differences between strains (behavioral phenotyping) (Figure 4.6). Specific ethology metrics to assess the degree of activity within EPM were calculated from the tracking data: percentage of time in open arms, total distance traveled, and percentage of time in the EPM center. In most cases, no significant differences were found between genders within the same strain (Supplementary Figure S 4.3) and, as such, the variable gender was dropped.

As expected, WKYs generally spend less time on the open arms of the EPM ($p <$ 0.05), since they are a strain characterized by high levels of anxiety and depression, as well as less time in the center of the EPM ($p <$ 0.05) (Figure 4.6A). There also appears to be a decrease in the traveled distance in WKY, when compared to Wistar rats (but without statistical significance). These results are consistent with the fact that WKY animals are generally less exploratory (D'Souza & Sadananda, 2017; Langen & Dost, 2011).



CaT--Z : 3D TRACKING AND BEHAVIOR CLASSIFICATION GUI

RGB-D DATA VISUALIZATION AND ANNOTATION GUI

Files' Selection - New annotation
Select directory: 7_dark/Frames_2017_10_25_14_08_58 ...

Resume previous annotation
Select file: ...

Save annotation work
Save data to directory: ...

RGB frame

Depth frame — Gray Scale - increased contrast

Timestamp (ms): 769334

Timestamp (ms): 769334

MIN= 0 Frame: 10074 Elapsed time (s): 769.334 State: Moving Exploration OPEN MAX = 12842

Behavioral ethogram:

CaT--Z software by AG & PA. 2017

Behaviors' Annotation

Frame	Behavior	Comments
0	Standstill	
100	Walking	
110	Moving Exploration	
200	Local Exploration	
220	Supported Rearing	
300	Unsupported Rearing	

CaT--Z : - RGB-D DATA A...

RGB-D DATA ACQUISITION GUI

Camera Connection
Connect camera ON

Real time acquisition
Select directory: ...
Start real time recording: START

CaT--Z software by AG & PA. 2017

CaT--Z : 3D TRACKING AND BEHAVIOR CLASSIFICATION GUI

3D TRACKING AND BEHAVIOR CLASSIFICATION GUI

Files' Selection: RGB and depth frames (.png files)
Select directory: C:/DATA/Frames_DEMO ... □ Multiple videos for training

Region of interest - tracking and classification
Select region of interest: ROI

Segmentation & Tracking - WHOLE-BODY + BODY PARTS

Select segmentation method:
● Median-difference approach
Static background: 300-frames' median
○ Median-dynamic approach
Adaptive background & 100-frames' median
○ Modified Gaussian Mixture Model approach
Adaptive background

Segmentation parameters (optional)
Border: Number of pixels (x): 60 □ full image
Number of pixels (y): 30 □ full image
Depth errors' range: min [mm]: 5
max [mm]: 300
Background model estimation:
# frames: 300
Animal's maximum area [pixel]: 1500

Start segmentation: START 43% Segmentation...

Save tracking data - centroid & body parts (if selected):
Select (optional)): ☑ Body Parts (nose, head and tail coordinates)
Save to .csv file: ...

Behavior Classification
Training | Testing | Classification
Select file with annotated frames: ...
Classification options
Number of classes: 8
□ Optimize C and gamma parameters
Number of folds: k 5
Train classification model: START

CaT--Z software by AG & PA. 2017

Figure 4.5. CaT-z: a free computational solution for quantifying animal behavioral features, in depth (z). Graphical user interface (GUI) of the applications developed for: **A1.** RGB-D frames visualization and annotation (main window); **A2.** Dock window for the annotation; **B.** RGB-D data acquisition (dark mode for animal facility environments); **C.** 3D segmentation, tracking and behavior classification.

The specific set of types of behaviors for the EPM were quantified and compared between both strains, and, as before, no differences were found between genders, within the same strain (Supplementary Figure S 4.3). When comparing both strains (Figure 4.6B), WKY animals spend less time in *rearing* periods than the Wistar rats ($p < 0.01$), whereas there were no statistically significant differences between groups in the other behaviors.

The combination of the metrics %time *walking*, %time *rearing*, and %time in the open arms, allow a high discrimination power when comparing strains using a kNN classifier (Figure 4.6C): accuracy of 79% and average posterior probabilities of 96% ± 12.6 and 25% ± 15.4, of a control or WKY sample, respectively, belonging to the control class. In addition, and according to the confusion matrix using these metrics, 2 rats in every 10 WKY rats were misclassified as belonging to the control class (20% false positives rate), while 22% of the controls were misclassified as not belonging to the Wistar class (false negative rate).

Thus, the results show that, although statistically significant differences were not found in isolated metrics, when they are combined, it is possible to distinguish the two strains with an accuracy degree of 79%. Furthermore, it is possible to construct behavioral profiles, characteristic of each strain, with 20% of false positives.

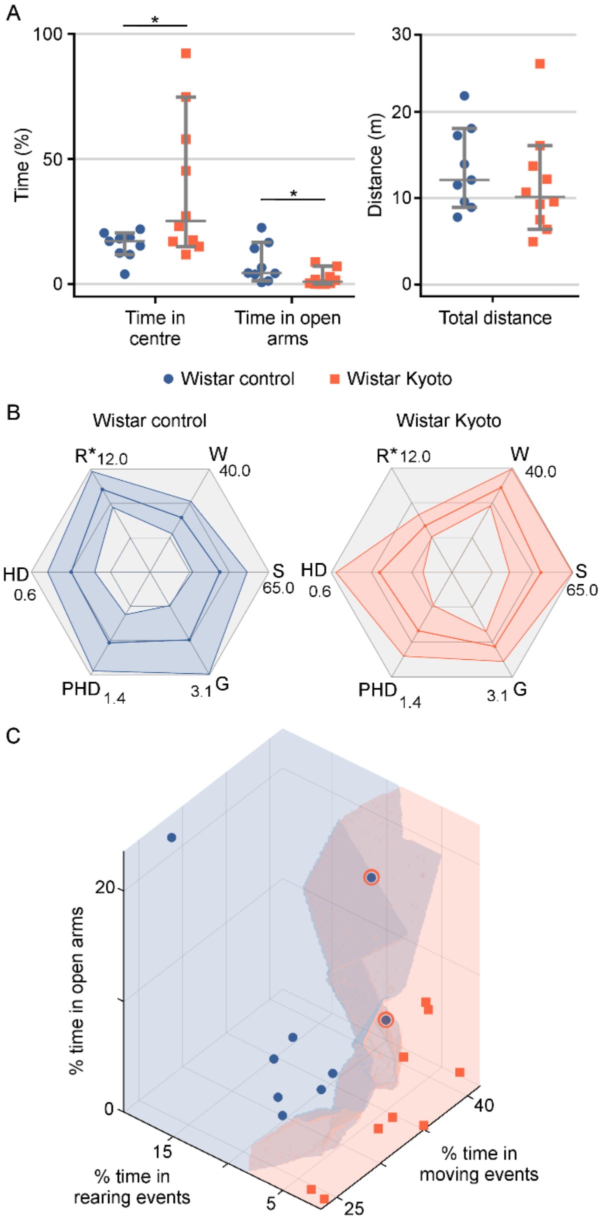


Figure 4.6 Distinction between Wistar Kyoto strains (behavioral phenotyping) made easy using RGB-D information. A. Motor activity measurements inside the elevated plus maze, for Wistar control (blue circles) and Wistar Kyoto (orange squares) rats. Data represented as median ± 95% confidence interval. $*p < 0.05$. **B.** Radar plot of automatic classification of behaviors for Wistar control (blue) and Wistar Kyoto (orange) rats. Solid lines (both blue and orange) represent median values. Shaded areas (both blue and orange) represent ±95% confidence interval. $*p < 0.05$. **C.** Three-dimensional representation of clustering results, for Wistar control (blue area) and Wistar Kyoto (orange area) rats,

regarding three features: % of time in moving and rearing, and % of time in open arms. Blue circles and orange squares represent well-classified points, for Wistar control and Wistar Kyoto, respectively. Blue circles with orange margin represent misclassified points, regarding the decision region of the clustering algorithm: both points should belong to Wistar control area but were misclassified as Wistar Kyoto points. *standstill* (S); *walking* (W); *rearing* (R); *head dipping* (HD); *protective head dipping* (PHD); *grooming* (G).

5. Discussion

The core goal of this work was to develop a free and fully-integrated system for 3D segmentation, tracking, and classification to automatically detect and quantify behaviors in rodents. With the developed algorithms, the *CaT-z* tool is capable of performing segmentation of a single animal's whole-body in complex backgrounds, tracking multiple body parts, and detecting different behaviors. These methods are embedded in a user-friendly software package, supported by a publicly available manual. The outputs of this tool are: 3D coordinates of body parts, automatically predicted behaviors, and, if applicable, corresponding performance metrics. From the 3D coordinates one can construct trajectories, and extract other motor parameters, such as distance traveled, average velocities, periods of active movement.

Importantly, this work also introduces the first publicly available RGB-D rat behavioral dataset that is suitable for training automatic behavior recognition in rodents, catalyzing new machine learning developments.

From the results, it was shown that 30 mins of annotated video of freely-walking movement is already sufficient to train our multiclass SVM classifier and attain accuracy levels which are comparable with the level of agreement in human observers (70-80%). The 30 mins figure is worth emphasizing since other methods, namely deep learning, typically require many hours of annotated videos for reaching high accuracy levels (but see (A. Mathis et al., 2018; T. D. Pereira et al., 2019). The ability to generalize is also fundamental in machine learning systems and, as demonstrated with the phenotyping experiments, *CaT-z* is able to cope not only with different setups but also with new types of behavior (without the need to redefine the features).

The use of depth sensors in analyzing animal behavior include advantages that go well beyond just adding a third dimension. Several research considered its potential application to segment and track rodents (Ou-Yang et al., 2011; Paulino Fernandez et al., 2014), as well as to estimate their pose, and social and non-social interactions (Hong et al., 2015; Matsumoto et al., 2013; Z. Wang et al., 2018). However, limitations as marker-imposition, basic poses/behaviors recognition, manual interventions, integration in a user-friendly public software, or insufficient classifier performance have limited their use. In addition to presenting important advantages over other approaches, *CaT-z* can be used to compare behavioral profiles ("behavioral fingerprints") of different strains. Previous studies have shown that WKY rats exhibit a combination of anxiety- and depressive-like behaviors, as well as hypoactivity and decrease in locomotion and social interaction levels (Burke et al., 2016; D'Souza & Sadananda, 2017; Langen & Dost, 2011). With our system, we were able to automatically quantify several behavioral differences that confirm these findings. More importantly, it was possible to automatically predict the strain of individual animals (with low false positive and false negative rates). Whereas automated behavioral phenotyping can be achieved in some conditions using home-cage 2D video data (Ethovision XT, Noldus, The Netherlands; Jhuang et al., 2010), this process can be greatly facilitated and improved when 3D information is available. Available solutions for automated behavioral phenotyping are often very expensive and limited to constrained/controlled environments (HomeCageScan - CleverSys, Inc, USA; LABORAS - Metris, The Netherlands; Phenocube - PsychoGenics, USA) or require the use of radiofrequency identification (RFID) implants which may affect animal behavior itself (IntelliCage, TSE, Germany; Weissbrod et al., 2013). For all the above reasons we are convinced that *CaT-z* has an important role to play in the computational ethology landscape.

The *CaT-z* software is freely available for download at GitHub (https://github.com/CaT-zTools/CaT-z_Software). The open-access dataset (41 GB) is also available at *Zenodo* (https://zenodo.org/record/3636136#.YbysrGjP2Uk).

6. Acknowledgments

AG and PA would like to thank Jaime Cardoso for a helpful discussion on potential strategies for performance improvement on SVMs. This work was partially funded by National Funds through FCT – Fundação para a Ciência e a Tecnologia, under the project EXPL-

AMAGALHÃES-IF/00753/2014/CP1241/CT0005, that also funds AM. Funding also by FEDER -Fundo Europeu de Desenvolvimento Regional funds through the COMPETE 2020 - Operational Programme for Competitiveness and Internationalisation (POCI), Portugal 2020, and by Portuguese funds through FCT - Fundação para a Ciência e a Tecnologia/Ministério da Ciência, Tecnologia e Ensino Superior in the framework of the project “Institute for Research and Innovation in Health Sciences” (POCI-01-3870145-FEDER-007274). PA was supported by Programa Ciência – Programa Operacional Potencial Humano (POPH) – Promotion of Scientific Employment, ESF. AG is funded by FCT – Fundação para a Ciência e a Tecnologia, grant contract SFRH/BD/137385/2018.

7. Author’s contributions

Ana Gerós implemented the algorithms, performed the experiments and was responsible for acquiring all the data. Ana Magalhães and Ana Gerós annotated the datasets. Ana Gerós and Paulo Aguiar developed the algorithms, analyzed and interpreted the data, and wrote the main manuscript. Paulo Aguiar devised the project and main conceptual ideas. All authors discussed the results and contributed to the final manuscript.

8. Supplementary Information

8.1 Extended Methodology

8.1.1 Animals.

Wistar rats (N = 12) and WKY rats (N = 10) from the colony of Instituto de Investigação e Inovação em Saúde, Portugal, aging 5–6 weeks, were used in the study. Rats were housed in pairs in a controlled environment (20 ± 2℃, 45–55% humidity) with a 12h light/dark cycle (lights off at 12h00). Food and water were supplied *ad libitum*. All behavioral experiments were performed during the animal’s active (dark) phase. All procedures were carried out under personal and project licenses approved by the national authority for animal protection, *‘Direção Geral de Alimentação e Veterinária’* (Portugal), and were performed in accordance with the European Directive 2010/63/EU on the protection of animals used for scientific purposes.

8.1.2 Behavioral Protocol

Behavioral experiments for dataset construction and system validation were conducted during 3 consecutive weeks for each animal, preceded by a habituation period (5 consecutive days) to reduce stress in the presence of a human experimenter. On experimental days, the rats were transported to the experimental room, and placed individually in a clean experimental environment (an opaque acrylic open-field cage, 1 m × 1 m × 0.5 m, made in-house), without bedding and accessories. The experiments were performed alternately with each animal, under 3 types of light set: dim red light, dim white light, and in total darkness (Figure 4.1A). All animals were recorded using *CaT-z* software without any external artificial markers. During the experiment, the animal was placed in the center of the apparatus and allowed to move freely for 15 mins. Between subjects, the apparatus was cleaned thoroughly with 70% ethanol.

For behavioral phenotyping studies, WKY rats were subjected to the EPM test. The apparatus consisted of a plus-shaped maze made of grey PVC with two opposing closed arms (10 cm wide × 50 cm long) with side walls (27 cm high), two opposing open arms (10 cm wide×50 cm long) and a central arena of 10 cm ×10 cm. The maze was elevated 50 cm above ground. The apparatus was placed in a room adjacent to the animal maintenance room so that animals were not disturbed by environmental stimuli. The light intensity was 30 ± 50 lux, with the light being more intense in the open arms. The test was performed by placing a rat in the central arena of the apparatus facing one of the open arms. Animals were allowed to freely explore the maze for 5 mins. Behavior was recorded using *CaT-z* software. The apparatus was cleaned between every animal test.

8.1.3 Animal detection and tracking

Landscape-change detection (LCD) method: applies background subtraction technique with constants updates of the background model. First, the absolute difference between the background model and the current depth frame is calculated to detect non-static objects (that will be potentially incorporated in the background model). After detecting the animal using connected component analysis (object with the maximum area), a new subtraction operation is performed to obtain a mask that contains new objects (including new dips or rises in the bedding material), shadows or reflections. The background model is then updated with the pixel values belonging to the new objects.

Probabilistic Gaussian Mixture model (GMM): this method was adapted from the original approach, by Stauffer and Grimson, 1999. Noteworthy, animal movement is typically heterogeneous, with variable velocities, and multiple stop periods of distinct durations. By keeping the model parameters (learning constant/rate and the proportion of background data) constant, the animal pixels are usually incorporated in the background model, leading to accumulated errors in the foreground estimation. The initial algorithm was improved so that the background update would only be performed if movement was detected between consecutive frames. Upon movement detection, the components of the adapted GMM model (weights, the mean and standard deviation for each Gaussian distribution) are determined using the Expectation-Maximization algorithm; background is then updated pixel-wise. To detect movement differences between frames, the algorithm calculates the pixel percentage that differs between two consecutive frames.

8.1.4 Pre-processing

After each background modeling approach, depth thresholding was applied to remove background noise (depth values fluctuations). Reflections and depth value errors in object borders can occur due to the intrinsic time-of-flight (ToF) operation of the *Microsoft Kinect* v2 sensor (Ganapathi, Plagemann, Koller, & Thrun, 2010a). In these cases, structures from the periphery that were miss-segmented as objects of the foreground were removed using morphological operations. Reflections, occurring because of the multi-path phenomenon in time-of-flight sensors, were removed using their distinct depth profile. After animal body detection, the center of the animal body (3D centroid coordinates) can be calculated for tracking purposes.

8.1.5 Body parts' detection and tracking

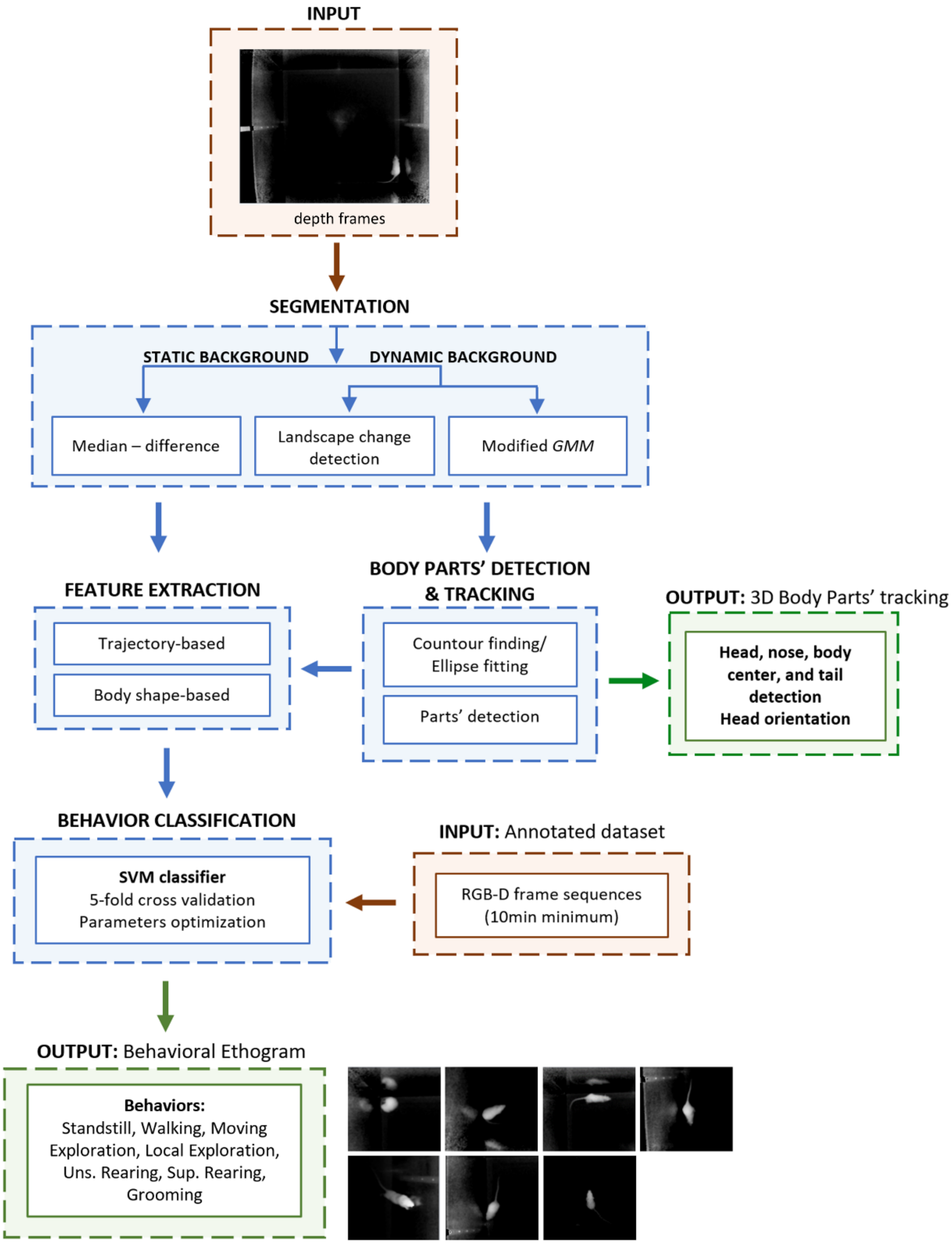
After finding the potential rodent body contours and x, y and z centroid points, the tail-end is defined as the furthest contour point from the centroid. Given the eccentricity of the mask containing animal's pixels, the nose and head points are detected using two different approaches (to cope with different positions of animal body). If the eccentricity is greater than a threshold (elliptical shape), the nose point is defined as the rostral contour point (the point from the opposite side of the tail) furthest from the centroid. To detect the head point, the thinning processing operation was applied to the animal mask in order to compute the skeleton mask and the head is then defined as the endpoint of the skeleton mask closest to the nose. For lower eccentricity values, the head point is defined as the endpoint of the skeleton mask furthest from the tail-end, and the nose as the contour point closest to the head. In both, nose coordinates are finally adjusted to correspond to the pixel whose depth value is highest one around its 7x7 neighbors, centered in the pre-determined point.

The tail-base position is calculated as the point in the skeleton mask where the distance to the periphery/background, measured using the distance transform mask, is close to a radius threshold (defined as 0.3). Finally, head orientation was calculated by taking into consideration the position of the nose and the head.

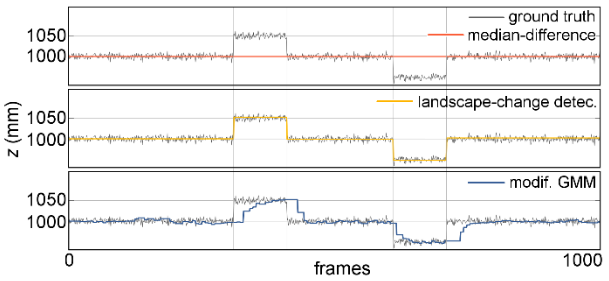
8.1.6 Statistical analysis

Statistical analysis was performed using *GraphPad Prism* version 7.0 (*GraphPad* Software Inc., CA, USA). The method of *D'Agostino* & *Pearson* was used as a normality test, and parametric or non-parametric tests were chosen as appropriate. Statistical significance was considered for $p < 0.05$. Parametric data are expressed as mean ± standard deviation (SD), and non-parametric data are expressed as median and 95% confidence intervals. The total sample size was calculated through a power calculation (using *G-Power* software): for a one-sided *Mann-Whitney* test with $\alpha = 0.05$, a power of 80%, and a standardized effect size *Cohen*'s d = 1.6, a total sample size of 12 rats (N = 6) was deemed sufficient. A total sample size of 20 would allow identifying smaller differences, with the same statistical power, for the same significance level.

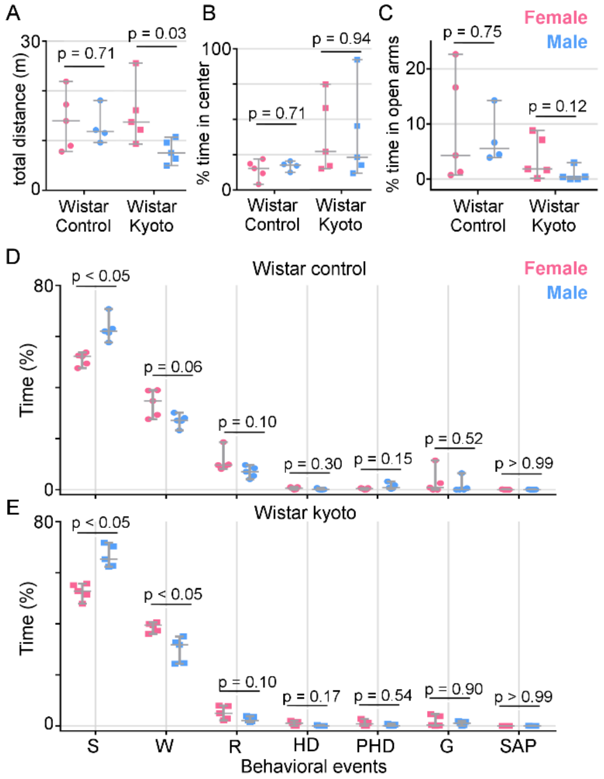
8.2 Supplementary Figures


INPUT
depth frames
SEGMENTATION
STATIC BACKGROUND
DYNAMIC BACKGROUND
Median – difference
Landscape change detection
Modified *GMM*
FEATURE EXTRACTION
Trajectory-based
Body shape-based
BODY PARTS' DETECTION & TRACKING
Countour finding/ Ellipse fitting
Parts' detection
OUTPUT: 3D Body Parts' tracking
Head, nose, body center, and tail detection
Head orientation
BEHAVIOR CLASSIFICATION
SVM classifier
5-fold cross validation
Parameters optimization
INPUT: Annotated dataset
RGB-D frame sequences (10min minimum)
OUTPUT: Behavioral Ethogram
Behaviors:
Standstill, Walking, Moving Exploration, Local Exploration, Uns. Rearing, Sup. Rearing, Grooming

Supplementary Figure S 4.1 Workflow of RGB-D tracking, segmentation and classification algorithm for the automatic recognition of rodent's behaviors. Uns. Rearing – *unsupported rearing*; Sup. Rearing – *supported rearing*.



Supplementary Figure S 4.2 Depth information improves segmentation in dynamic conditions. Comparison between the three implemented methods for background removal: static Median-difference (in orange), landscape-change detector (in yellow), and modified Gaussian Mixture Models (GMM) (in blue). The depth information for an illustrative coordinate is presented. The synthetic dataset included modifications of the ground-level (insertion and subsequent removal of objects). Ground truth profile in gray.



Supplementary Figure S 4.3 Comparison between female and male individuals show no statistical differences, for both Wistar and Kyoto rats. A-C. Motor activity measurements (%time in the open arms, %time in the center, and total distance in meters (m), respectively), for Wistar control (circles) and Wistar Kyoto (squares) rats. **D-E.** Automatically predicted behaviors for Wistar control (circles, top) and Wistar Kyoto (squares, bottom) rats. Female and male individuals are represented by pink and blue markers. Data represented as median ± 95% confidence interval. ns - no statistical differences. S – *standstill*; W – *walking*; R – *rearing*; HD – *head dipping*; PHD – *protective head dipping*; G – *grooming*; SAP – *stretch-attend posture*.

8.3 Supplementary Tables

Supplementary Table 4.1 Description of each type of behavior, and corresponding overall frequencies, on the annotated RGB-D dataset.

Behavior	**Description**	**Frequency (%)**
standstill	rest at one place, no movement of limbs or head	4.0
walking	body clearly moves from one place to another, movement of limbs (hind and forelimbs) and snout raised	2.5
moving exploration	body clearly moves from one place to another, movement of limbs (hind and forelimbs) and exploratory sniffing	21.5
local exploration	no movement of the hindlimbs, occasional micromotions (e.g. sniffing)	37.9
supported rearing	rise up on hindlimbs, and forelimbs off the ground but supported on objects/wall	8.6
unsupported rearing	rise up on hindlimbs, and forelimbs off the ground with no support	7.8
grooming	licking the fur or scratching with forepaws in curled body position	17.7

Supplementary Table 4.2 Features' description.

Type	Feature	Description
Shape-based features	Maximum depth value	Maximum depth value between all pixels of the animal's mask
	Body Area	Sum of all pixels of animal's body
	Body Radius	The longest distance between centroid and animal's body contour
	Circularity	The square proportion between body area and body radius
	Ellipticity	The ratio between long and short axes of the ellipse (after fitting)
	Flag for body parts' detection	Flag to indicate if body parts' detection was possible (if not, the feature values that depend on that detection, take feature values of previous frame)
	Depth value of the nose	The depth value of the nose point (when detected)
Trajectory-based features	Angular velocity of the nose	Head direction (angle between centroid-head vector and head-nose vector), divided by the time interval (moving average filter on a non-homogeneous time grid; time window = 0.5 seconds)
	Minimum distance to the walls	Minimum distance from the centroid and the open-field walls (given by the ROI)
	Speed of centroid	Pixel distance between animal's positions in two consecutive frames, divided by the time interval (moving average filter on a non-homogeneous time grid; time window = 0.5 seconds)
	Speed of centroid	Pixel distance between animal's positions in two consecutive frames, divided by the time interval (moving average filter on a non-homogeneous time grid; time window = 1.0 seconds)

Supplementary Table 4.3 Per event recognition performance for each lighting condition and multiple videos of approximately 10/15 minutes each, for different training sets. Results are expressed as median [95% confidence interval] (N = 4 videos), using the leave-one-video-out technique. For training with videos for a particular lighting condition, no significant differences were observed between different lighting conditions (F1-scores and accuracy) except in local exploration behavior (*Kruskal-Wallis* test; * $\boldsymbol{p} < 0.05$). For training with videos of dim red-light conditions, no significant differences were observed between different lighting conditions (F1-scores and accuracy; *Mann Whitney* test). For testing with the same lighting condition and different training sets, no significant differences were observed (F1-scores and accuracy; *Mann Whitney* test).

F1-SCORE					
	Training set (individually for a particular lighting condition)			Training set (dim red light videos)	
Testing set / **Behavior**	**Red light**	**White light**	**Total darkness**	**White light**	**Total darkness**
standstill	7.3 [-10.1,24.7]	N.A.	1.5 [-23.0,26.1]	N.A.	12.2 [-20.1,44.5]
walking	51.7 [27.8,75.7]	35.8 [13.5,58.2]	38.4 [21.0,55.7]	59.5 [32.3,86.6]	49.0 [23.5,74.4]
moving exploration	72.9 [68.8,77.1]	69.6 [61.4,77.7]	66.3 [53.2,79.4]	69.0 [54.4,83.7]	58.5 [46.8,70.1]
local exploration	64.5 [52.9,76.1]*****	81.1 [72.4,89.7]*****	72.2 [61.8,82.6]	75.6 [73.8,77.4]	67.1 [57.7,76.4]
supported rearing	88.0 [81.2,94.7]	80.8 [73.8,87.8]	87.2 [85.2,89.2]	74.6 [78.3,90.9]	85.7 [78.5,93.0]
unsupported rearing	75.5 [52.3,98.8]	76.0 [63.0,88.9]	73.1 [39.4,106.8]	73.9 [67.9,79.8]	65.7 [34.9,96.4]
grooming	79.6 [69.9,89.4]	65.3 [51.4,79.1]	75.1 [56.4,93.7]	69.2 [43.6,94.7]	72.3 [69.4,75.3]
ACCURACY (%)	**70.8** [61.9,79.7]	**76.0** [71.0,81.0]	**70.0** [67.6,72.2]	**73.2** [65.4,81.0]	**67.9** [61.3,74.5]

8.4 Supplementary Movies

Supplementary Movie 1 Automated animal segmentation video of freely-walking rat inside the open-field cage, displayed at 10 fps. Segmented video using the static Median-difference method for background removal, with the classification output of the machine learning algorithm overlaid (4 classes): *standstill*, *walking*, *rearing* and *grooming*. Black pixels correspond to background pixels. Depth colormap as in Figure 4.2A.

Available at: https://link.springer.com/article/10.3758%2Fs13428-020-01381-9.

CHAPTER 5

Deep Learning-based System for Real-Time Behavior Recognition and Automated Closed-Loop Control of Behavioral Mazes

This chapter was based on the following original research paper:

Gerós, A., Cruz, R., de Chaumont, F., Cardoso, J. S., & Aguiar, P. Deep learning-based system for real-time behavior recognition and automated closed-loop control of behavioral mazes using depth sensing (under review).

1. Abstract

Robust quantification of animal behavior is fundamental in experimental neuroscience research. Systems providing automated behavioral assessment are an important alternative to manual measurements avoiding problems such as high cost, human bias, and low reproducibility. Integrating these tools with closed-loop control systems creates conditions to correlate environment and behavioral expressions effectively, and ultimately explain the neural foundations of behavior.

We present an integrated solution for automated behavioral analysis of rodents using deep learning networks on video stream acquired from a depth-sensing camera. The use of depth sensors has notable advantages: tracking/classification performance is improved and independent of animals' coat color, and videos can be recorded in dark conditions without affecting animals' natural behavior. Convolutional and recurrent layers were combined in deep network architectures, and both spatial and temporal representations were successfully learned for a 4-classes behavior classification task (*standstill*, *walking*, *rearing* and *grooming*) using depth input sequences. Integration of an Arduino microcontroller creates an easy-to-use control platform providing real-time feedback signals based on the deep learning automatic classification of animal behavior. The complete system, combining depth-sensor camera, computer, and Arduino microcontroller, allows simple mapping of input-output control signals using the animal's current behavior and position. For example, a feeder can be controlled not by pressing a lever but by the animal behavior itself. An integrated graphical user interface completes a user-friendly and cost-effective solution for animal tracking and behavior classification. This open-software/open-hardware platform can boost the development of customized protocols for automated behavioral research, and support ever more sophisticated, reliable and reproducible behavioral neuroscience experiments.

Keywords: Deep Learning • Depth Sensing • Spatiotemporal Features • Feedback Control • Closed-Loop System • Automated Behavior Classification • Free Open Source Software • Animal Tracking • Automated Control of Behavioral Mazes

2. Introduction

Behavior is shaped by interactions between the organisms and the environment, being the most important output response of the nervous system to external (and internal) stimuli. Understanding this relationship between behavior and neural activity is the central goal of systems neuroscience, which relies on analyzing animal behavior for theorizing cognitive mechanisms and ultimately explaining the underlying neural circuits (Anderson & Perona, 2014; Berman, 2018; Krakauer et al., 2017). Besides basic neuroscience research, the study of animal behavior plays a key role in the translational analysis of disease models, preclinical assessment of therapies' efficacy, and also in food production industries (Anderson & Perona, 2014).

The research on animal behavior has benefited from the recent technological advances in machine vision and machine learning fields, allowing for the collection and automatic quantification of vast amounts of data. Besides reducing human bias and subjectivity, and consequently allowing for the standardization of measurements across laboratories, behavioral patterns that were once unnoticed to a human observer may now be explored at different scales and resolutions (Macpherson et al., 2021; M. W. Mathis & Mathis, 2020; Robie et al., 2017). The first approaches to successfully combine computer vision and machine learning techniques typically relied on hand-crafted features extracted from images or video sequences that can be then used for automated behavior classification using supervised (de Chaumont et al., 2019; Gerós, Magalhães, & Aguiar, 2020; Jhuang et al., 2010; Kabra et al., 2013) or unsupervised (Malte Lorbach et al., 2019; Marques et al., 2018; Wiltschko et al., 2015) learning methods. However, such approaches are highly dependent on domain expertise for feature engineering, often losing their generalization capability in the presence of a new environment/scenario. Recent developments in the computational neuroscience field have explored deep learning techniques to meet this challenge. Most state-of-the-art systems present powerful deep learning-based solutions for pure body-part detection and tracking for pose estimation (Dunn et al., 2021; Forys et al., 2020; Geuther et al., 2019; Graving et al., 2019; A. Mathis et al., 2018; T. D. Pereira et al., 2019; Romero-Ferrero et al., 2019), but modest progress has been made for direct recognition of behavioral events (Bohnslav et al., 2021; Jiang et al., 2019; M. Marks et al., 2020). When compared to action detection in humans, which already achieved outstanding performance in challenging benchmarks, animals' behavior is more complex to characterize. First, some animal behaviors are very similar to each other (more easily confused than those of humans), in which temporal information is necessary for a flawless detection (sometimes a single frame is not enough to label the behavior correctly). Recent approaches take advantage of deep architectures that integrate temporal information along with spatial information to this end (Bohnslav et al., 2021; Jiang et al., 2019; M. Marks et al., 2020). Also, different behaviors have different durations and temporal scales: some of them take place in long time scales, such as *grooming*, and others in short time scales, such as *rearing* or *walking*. To the best of authors' knowledge, temporal multi-scale integration has not been explored in the context of animal behavior analysis. Another concern when planning behavioral experiments is to ensure that the environment where the animal moves is adequate to allow capturing natural behavior and yet probing for multiple parameters for its study. In particular, an important limiting factor for recording natural rodent behavior is the environment lighting conditions (which may affect animals' biological cycle). Usually, the most natural conditions are left behind at the expense of recording conditions (higher image resolution or contrast). One possible strategy is to use cameras with infrared technology (such as deep sensing cameras). A few studies have recently begun combining deep learning methods with data from such technologies for animal behavior analysis (Nourizonoz et al., 2020). Finally, to effectively correlate behavioral functions with specific neural circuits, automatic behavioral analysis tools should ideally be integrated into real-time closed-loop control systems, that provide instantaneous feedback based on the current behavioral expression. There are already published tools that provide feedback control in real-time based on animal posture patterns (de Chaumont et al., 2019; Forys et al., 2020; Kane et al., 2020; Nourizonoz et al., 2020; Schweihoff et al., 2021; Sehara et al., 2021). However, they do not satisfy all these requirements simultaneously for a complete and versatile behavioral analysis system.

Here, we introduce a novel computational solution for automated, markerless, real-time three-dimensional (3D) tracking and behavior classification of 4 classes (*standstill*, *walking*, *rearing* and *grooming*) in experiments with a single freely-behaving rodent. Combining the power of low-cost depth sensors and deep learning techniques, the proposed framework is integrated into a control platform that streams real-time mapping of input-output signals using the animal's current behavior and position. First, we analyze the performance of advanced action recognition deep learning networks on the rodent behavior dataset. Acknowledging the importance of integrating temporal information in behavioral feature learning, we hypothesized whether abstract spatiotemporal features obtained from simple deep networks are suitable for recognizing multiple behaviors. In particular, the behavior of networks for increasing temporal extents and with multiple timescales' branches (partially inspired in Feichtenhofer, Fan, Malik, and He (2019)) was compared regarding their performance in detecting behavioral events. We found that temporal information from the past, using a short-time scale, is most relevant for the learning process. Second, we analyze how robust the proposed networks were at different input representations (input frame encodings, sampling rates, and resolutions), where raw depth frames at higher sampling rates and resolutions helped improve classification performance. Also, ~21 minutes (mins) of annotated video showed to be already sufficient to attain a good generalization using proposed deep networks for behavior classification. Lastly, we adapt the deep learning framework to recognize animal tracking and behavior in real-time, and we integrate it into a platform capable of closed-loop control of behavioral experiments, either for behavioral mazes or real-time drug delivery systems. Besides being non-invasive and with low latency, it provides a versatile interface to trigger different hardware actuators from either hardware sensors or behavior/tracking-dependent signals.

3. Materials and Methods

The proposed system for online rodent behavioral recognition consists of two components: deep learning networks and the real-time control module. The deep learning networks are responsible for spatiotemporal feature extraction and behavior/position detection for each frame in the video sequence. The real-time classifications are used to control sensors/actuators in any maze. All these tasks can be controlled through an easy-to-use graphical user interface (GUI) for beginning-to-end management of all experiments.

3.1 Dataset

An open-access RGB-D behavioral dataset, available at https://doi.org/10.5281/zenodo.3636135 (Gerós et al., 2020), was used for all experiments. Details on the experimental procedures, video acquisition and manual annotation of rodent's behavior can be found in (Gerós et al., 2020). In brief, the dataset is composed of 10 to 15 mins RGB-D video sequences of individual Wistar rat behavior, recorded with a *Microsoft Kinect v2* camera (512x424 depth pixel resolution). The maximum frame rate is 30 frames per second (fps), but this value typically drops to 10 to 15 fps in low light conditions. A subset list of classes was considered here with the four most commonly used state behavior states:

standstill, *walking*, *rearing* and *grooming*. A randomly selected subset of these fully annotated recordings was considered for the experiments and denoted as dataset-100k (~2.20 h in 26 subvideos, approximately 100,000 frames total, with a time difference between two consecutive frames of approximately 67 milliseconds (ms)). Only the depth frames were kept for analysis.

3.2 Proposed deep learning model

In order to create a framework that incorporates spatiotemporal features for video understanding tasks, a neural network architecture is proposed and validated. The network consists of an encoder and a classifier, which is trained end-to-end. The encoder consists of two-dimensional (2D) convolutional layers, to extract local spatial features in each frame of the video sequence. The classifier is composed of a recurrent layer to learn temporal features between adjacent frames in the video sequence, and fully-connected layers to output the behavioral classes' probabilities (Figure 5.1).

3.2.1 Architecture

Two variants of the encoder were considered – the single-branch and the dual-branch. The single-branch receives an input sequence with a time-window of size T ms, with frames equally spaced over time by a temporal stride of τ ms. The dual-branch variant receives two sequences with different temporal strides in each stream (Figure 5.1). The idea is for the two pathways to exploit temporal information of a different scale: the short-time scale provides information hidden in temporally neighboring frames, giving clues about animal's movement at fast temporal changes, while the long-time scale may help distinguish between different behaviors at slower temporal changes (namely, transitions between behavioral states). In both architectures, frames are individually encoded by four 2D convolutional layers (64 filters, 3x3 kernel size, 2x2 stride, rectified linear unit (ReLU) activation). After the encoding part, a recurrent layer (RNN, 128 hidden state features) takes as input the sequence of spatial features output by the feature extractor and integrates it over time for both temporal and spatial dynamics learning. Two fully-connected layers (64 and 32 channels) and a softmax output layer are used for the final recognition of behavioral classes.

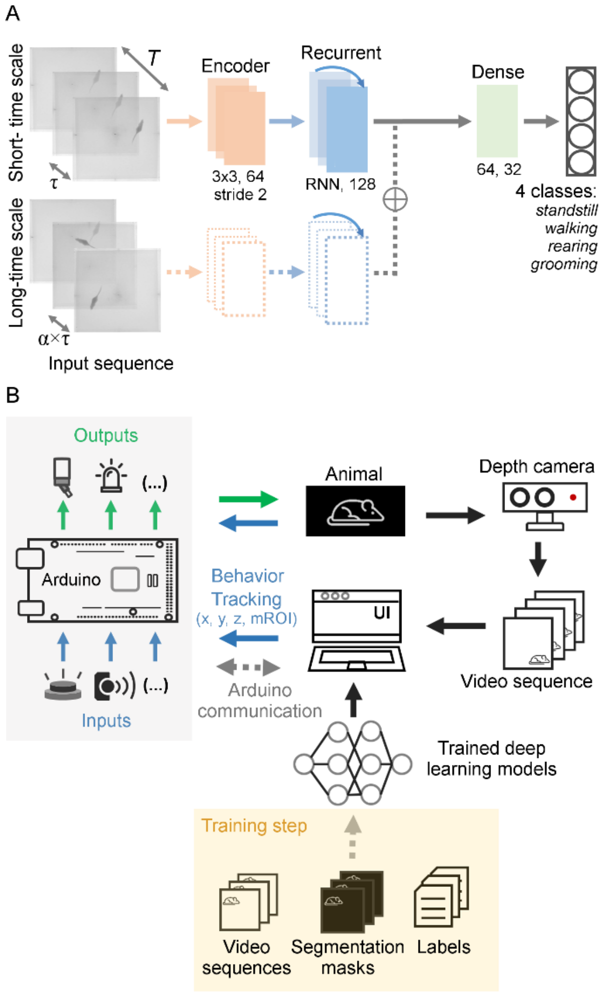


Figure 5.1 Integrated framework for the control of behavioral mazes using depth information and deep learning-based techniques. A. Deep learning architecture, with the two variants of the encoder, single-branch (solid line) and dual-branch (solid and dashed lines), for the automatic classification of 4 behavioral classes. Both variants receive one input sequence with a time-window of size $\boldsymbol{T}$ ms, with frames equally spaced over time by a temporal stride of $\boldsymbol{\tau}$. The dual-branch variant receives additionally one sequence with a different temporal stride, long-time scale pathway, that operates on a bigger time-window ($\boldsymbol{\alpha}$ × $\boldsymbol{T}$') with a temporal stride of $\boldsymbol{\alpha} \times \boldsymbol{\tau}$ ($\boldsymbol{\alpha}$ >1, where $\boldsymbol{\alpha}$ is the frame rate ratio between short- and long-time scale pathways). **B.** Workflow of the closed-loop feedback system, for controlling behavioral experiments. Depth video sequences are acquired by a depth camera, and used as inputs to deep learning networks for real-time automatic classification of behavior and detection of animal's position (x, y, and z coordinates of centroid, and any defined regions-of-interest inside the maze (mROI)). Such signals, together with input

signals coming from any sensor hardware (blue), are sent to the Arduino microcontroller for feedback control of the actuators present in the maze (green). For real-time behavior classification and detection of animal’s position, the deep learning models must first be trained using a training set with annotated depth video sequences (segmentation masks and behavioral labels).

In the case of the dual-branch, both pathways work on different time-windows: the short-time scale pathway receives as input a pre-defined time-window T' with the same temporal stride τ as the single-branch network; the long-time scale pathway operates on a bigger time-window ($\alpha \times T'$) with a temporal stride of $\alpha \times \tau$, where $\alpha > 1$ is the frame rate ratio between short- and long-time scale pathways. Two recurrent layers are used for each branch, which are then concatenated before the fully-connected layers.

Since recognizing rodent’s behavior is a challenging task, either due to the size of the animals or the nature of the behaviors (faster movement, higher similarity and greatly dependent on temporal information to be clearly distinguished), the feature extraction process needs to be carefully designed to avoid confusion between behavioral events. For this reason, 2D convolutions were chosen, instead of the currently used 3D convolutions for spatiotemporal learning, in order to process spatial and temporal content separately and thus avoid mixing information of different scales. The reduced number of convolutional layers and the number of filters at each layer allow the entire network to be computationally lightweight and capable of being used for real-time inference afterwards.

3.2.2 Training

The models were trained from scratch using the ADAM optimizer, with a batch size of 16 video sequences with a time-window of T ms , and a learning rate of 1×10^{-4}, for 100 epochs. A dropout layer was used before the recurrent layer, with a dropout ratio of 0.5.

Initially, the dataset was split into training (70%), validation (10%) and testing (20%) sets that are maintained throughout the experiments. The validation set was used to compare the performance of different models when performing ablation studies. To address the problem of having a highly imbalanced dataset (*standstill* 40.3%, *walking* 28.7%, *rearing* 11.7%, and *grooming* 19.3%), the video sequences of each class were oversampled until their frequencies were uniform.

3.2.3 Experiments

For a systematic study of networks' performance, the effect of increased temporal information was evaluated, by changing different parameters in each experiment. First, the impact of changing the time-window T of the input sequence was tested, with $T \in \{0\tau,\ 1\tau,\ 4\tau,\ 10\tau,\ 19\tau\}\, ms$, corresponding to a network input with 1 (single-frame), 2, 5, 11 and 20 frames in total, respectively, sampled with a fixed temporal stride τ of 133 ms. Also, the temporal stride τ between adjacent frames ($\tau \in \{67, 133\}\, ms$) was evaluated, which corresponds to approximately 15 or 8 frames sampled per second, with a fixed time-window. Finally, the frame rate ratio α between short- and long-time scale pathways for the multi-branch architecture ($\alpha \in \{5, 10\}$) was varied. These temporal parameters were chosen in order to make the network responsive to the different behavior timescales present in the original dataset. In this sense, and taking into consideration the camera's frame rate, the capability of the network of capturing both fast behavioral events (in the order of a few hundred milliseconds) and slower events (in the order of a few seconds) was explored. Also, different spatial resolutions of $\{64, 128, 256\}$ pixels and input encoding modalities were tested. Besides raw 8-bit depth frames, depth jet-encoding (Eitel, Springenberg, Spinello, Riedmiller, & Burgard, 2015) was applied to depth frames, in which the depth information is distributed according to the jet colormap, transforming the one-channel depth map to a three-channel color image. Also, surface normals were used to encode the depth frames into a three-channel image representing form and surface structure (implementation details in Madai-Tahy, Otte, Hanten, and Zell (2016)). Unless otherwise noted, the full dataset-100k was considered for analysis, and the default parameters for the systematic study were: $T = 10\tau$, $\tau = 133$ ms, spatial resolution of 128 pixels in raw depth frames. The influence of training set size on network generalization was also benchmarked. Different training sizes were selected and each subsampled training set was used to train the network, and compared with the same validation set (using the default parameters' set as well).

3.2.4 Data augmentation

To improve the robustness and generalization of the models, data augmentation was performed with random perturbations of the training set during training, that included: full-rotation around the center (90/180/270°); horizontal flipping; resized cropping and brightness variation (by sampling an additive value from a uniform distribution, [-0.15, 0.15]).

As the input of all models is a frame sequence of approximately T/τ frames, the same augmentation operations were performed on each frame in this set.

3.2.5 Model evaluation and metrics

The validation set was used for models' comparison and evaluation, and all analyses reported share the same validation set, for a total of 5 runs for each experiment. The hold-out testing set was further applied to evaluate the performance of the best-chosen model to an unseen set. To evaluate the overall performance of the different proposed methods, balanced accuracy (average of recall obtained on each class) and weighted F1-score (weighted average F1-score over all classes according to classes' relative frequency) were calculated. Performance per class was assessed using confusion matrices and corresponding F1-score.

The F1-score is the harmonic average of the precision and recall, calculated as follows:

$$F1\ score = 2 \times \frac{precision\ \times recall}{precision + recall} \tag{5.1}$$

where $precision = \frac{true\ positive}{(true\ positive + false\ positive)}$ and $recall = \frac{true\ positive}{(true\ positive + false\ negative)}$.

These metrics are better suited to deal with imbalanced datasets.

3.3 Real-time control system

The entire control system consists of software and hardware modules configured to create an automated closed-loop tool. It is made of five main components: the control computer, the interface board, the control software, the video camera and the maze hardware modules (Figure 5.1). Frames acquired by a depth camera are fed into the trained deep learning models, which will automatically detect both behavioral events and the animal's position in the maze. The network outputs are sent to the interface board that, together with existing sensor outputs (e.g., buttons, maze sensors), controls circuit actuators (e.g., maze feeders, light-emitting diodes (LED)s). The computer is used to operate the entire circuit by a graphical user interface (GUI), either sending messages to the interface board or acting directly on the maze hardware modules.

3.3.1 Interface board

An Arduino microcontroller (Mega 2560) was used as the interface board between the computer and the hardware modules, and the communication is established using a communication (COM) port. The microcontroller board has 16 MHz clock speed, and 54 digital input/output ins that can be connected to different maze hardware components, such as animal feeders, LEDs, maze sensors, and buttons. After being connected to the computer, the Arduino board communicates via Arduino integrated development environment (IDE). The user writes the Arduino code for the automated control in the IDE, uploads it to the microcontroller which executes the code to interact with the input and output hardware modules. Notice that, once uploaded, the code can run regardless of the connection between the Arduino and the computer.

3.3.2 Control software

The automated control software consists of the following components: the automation control code, the trained deep learning models for detection, and the data acquisition and communication protocol.

Automation control code

Arduino code is written within the Arduino IDE (in a language very similar to C++) and it was carefully organized to segregate the code for specific logic state implementations (automated control) from all other maintenance code (such as reading and writing data to the communication port (COM). To do so, a specific user-defined function was created, which has access to all critical variables for the control, such as sensors' and actuators' states, and animal's position and behavior. Inside this function, the user can easily define the conditions of stimuli-response that characterize each behavioral test experiment.

Deep learning models

In order to automatically classify the behavior and calculate the position of the animal using deep learning methods, previously trained models are imported and directly used for predictions. For the automatic classification of behavior, the single-branch model was trained according to the protocol previously described (input sequence of raw depth frames, with a time-window of approximately 1330 ms, acquired at a frame rate of 15 fps). For the estimation of animal's position, two different methods were made available to the user: deep learning-based model for semantic image segmentation, and conventional background subtraction model, both followed by centroid calculation. The deep learning-based model combines two ingredients from deep networks' knowledge in order to perform semantic segmentation taking into consideration temporal information: U-Net model as backbone architecture, and (optional) convolutional Long Short-Term Memory (ConvLSTM) layers, learn spatiotemporal features. The traditional U-Net architecture was reduced to only one convolutional layer per block, fewer filters per layer (32) and it was extended by placing two ConvLSTM layers, one between the encoder and the decoder, and the other one before the last dense layer (different positions in the network, as well as different architecture parameters, were tested to ensure maximum performance yet reduced inference time and memory (Supplementary Figure S 5.1)). The network was trained from scratch using 1220 train and 320 validation video sequences (previously annotated to obtain the segmentation masks), with ADAM optimizer and dice binary cross-entropy (BCE) loss function.

A conventional background subtraction method was integrated in parallel to provide a computationally lighter alternative yet with lower performance (mainly in frames with dynamic backgrounds). Using this method, the segmentation mask containing animal's pixels is produced by subtracting the present frame with the background model (frame of the behavioral experimental setup without the animal). From the segmentation mask, the position of the animal is calculated as the centroid of the detected object/animal. For details on algorithm's design and performance, please check Gerós et al. (2020).

For a more complete information about animal's movements inside the maze, the system allows the user to define spatial mROIs, by uploading an image file with the same resolution as the acquired frames, with the different mROIs painted uniformly with different colors. Those regions are automatically detected after getting animal's tracking, and they will be used as input for the Arduino board to control the hardware mazes, if needed.

Data acquisition and communication

To establish the communication between the COM port and the Arduino board, a communication protocol was defined. The computer communicates with the interface board by sending the behavioral classification, tracking and mROI outputs (as well as a flag for any keypress), in the form of a characters' list separated by commas. Each character encodes information for the behavioral state (S for *standstill*; W for *walking*; R for *rearing*, and G for *grooming*), tracking (x, y and z coordinates of the centroid), mROI and a key-pressed flag (both encoded as integers). On the other hand, the Arduino board sends information regarding the status of each of the sensors and actuators (binary coded, on/off) back to the computer.

3.3.3 Video camera

The acquisition protocol was developed using a new generation of low-cost depth cameras, the *Intel*® *RealSense* Depth Cameras (in particular, D435 model), acquired with 512x424 depth pixel resolution and at a maximum of 30 fps.

3.3.4 Computational performance: inference and latency times

To test time-performance of the system, a video of a freely-walking rat was used to simulate a camera feed from an animal in real-time, and single frames from the video were loaded at the maximum rate of 30Hz. The bidirectional communication with the Arduino board was achieved from either four input sensors and signals from the computer, and four output actuators (in this case, LEDs). Three latency periods were measured: (a) the delay from image acquisition to detecting the behavioral state/tracking position (image-event delay); (b) the delay from detecting one behavioral event/tracking position to the next event/tracking position (event-event delay, including Arduino response, mROI detection, GUI updates and saving images to external folder); (c) the delay between sending a behavioral state to the Arduino and turn on the corresponding LED (event-LED delay, with and without output feedback of Arduino). The first two latency times were determined using software timestamps and the last one was measured using the oscilloscope.

3.4 Computing hardware

All experiments, including inference speed and feedback control tests, were conducted on an *Intel*® Core i9-7940X (128 GB RAM), and a *NVIDIA* GeForce RTX 2080 graphics processing unit (GPU) (8 GB RAM), running *Windows* 10, with *Python* 3.9 using *PyTorch* (1.8.1) and *TensorFlow*-GPU (2.5.0) frameworks. All algorithms were integrated into a user-friendly GUI, designed in the *Qt Creator* (*The Qt Company*, Finland) environment and implemented in *Python* language.

3.5 Statistical methods

Statistical analysis was performed using *GraphPad Prism* version 7.00 (*GraphPad* Software Inc., CA, USA). The method of *D'Agostino & Pearson* was used as a normality test, and parametric or non-parametric tests were chosen as appropriate. Statistical significance was considered for $p < 0.05$. Parametric data are expressed as mean ± standard deviation (SD), and non-parametric data are expressed as median and 95% confidence intervals.

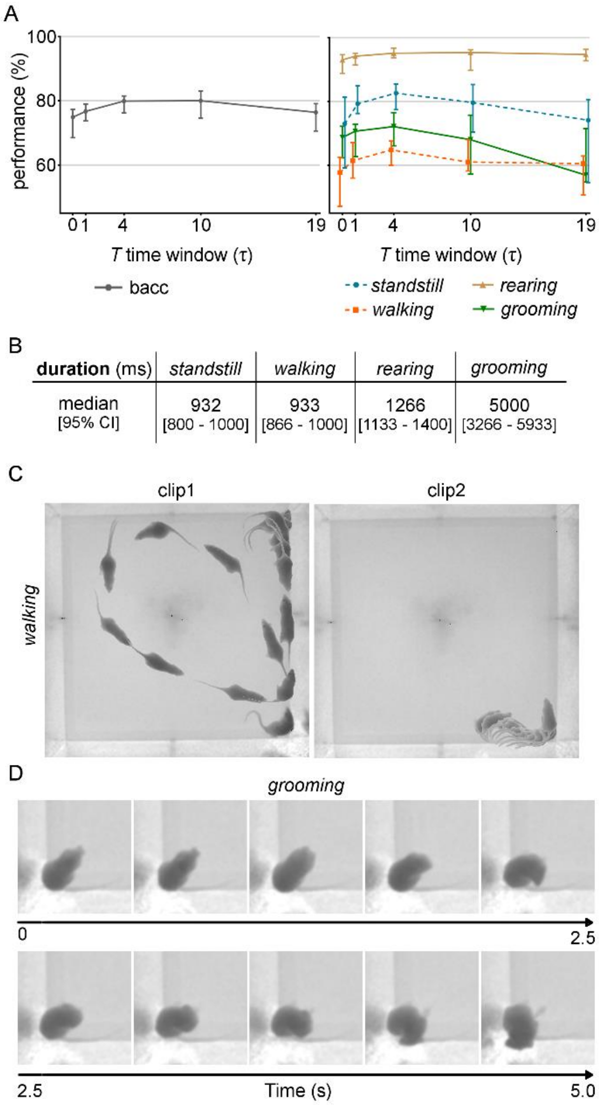
4. Results

4.1 Learning spatial and temporal features: modeling approach

To take advantage of the temporal content present in RGB-D videos, and to understand if and how temporal information can help the learning process, networks with different architectures and input representations were studied.

4.1.1 Past information improves behavioral classification performance

In particular, the time-window T of the sliding input sequences was systematically increased, with a fixed temporal stride $\tau = 133$ ms, to investigate the behavior of networks for increasing temporal extents (Figure 5.2A and Supplementary Figure S 5.2). Improvements over T were observed, where models with a time-window of 10τ (approximately 1500 ms, 11 frames in the sequence) achieved the top overall results on the validation set, with a balanced accuracy of 80.0% [74.6, 83.0]%. No statistical differences were found when using as input a time-window of 4τ. The results seem to indicate that the gain of increased time-window is clearer for networks with a smaller time-windows, with a converging trend towards time-windows above 1000 ms. This is aligned with the timescale for the analyzed animal behavior classes (where the timescale for variation is in the order of 1 second) (Figure 5.2B). For time-windows smaller than 300 ms, the performance significantly dropped.



duration (ms)	standstill	walking	rearing	grooming
median [95% CI]	932 [800 - 1000]	933 [866 - 1000]	1266 [1133 - 1400]	5000 [3266 - 5933]

Figure 5.2 How much temporal information does the network need for rodents' behavioral learning? A. Results using single-branch architecture of varying temporal extents. Left: Overall balanced accuracy (bacc) for increasing temporal extents. Right: F1-score per class. Time window $\boldsymbol{T}$ in units of $\boldsymbol{\tau}$ ($\boldsymbol{\tau}$ = 133 ms). Data represented as median ± 95% confidence interval (N = 5 trials). **B.** Behavioral events' duration, in milliseconds (ms). Data represented as median ± 95% confidence interval. **C.** Stroboscopic montage in which each animal position represents raw depth frames extracted at every 266 ms for 2 different *walking* clips. **D.** Sample clips with frames extracted at every ~500 ms, for a single *grooming* clip.

When no temporal information was taken into consideration, using a model with only one input frame, the lowest overall accuracy was achieved, as well as category F1-score, showing that not only spatial information within a particular frame may be important but also its motion content across different frames. In fact, when performing manual annotations, ethologists often need to double-check previous frames to annotate the current one, which also seems to happen in these networks.

Out of all 4 classes, no behavioral event has a monotonic decrease with the increasing temporal extent, and overall their recognition seems to benefit from time-windows smaller than 1000 ms (category F1-score systematically increasing over $\boldsymbol{T}$, until approximately 1000 ms). This effect is particularly clear during *standstill*, *walking* and *grooming* events, where F1-score performance seems to slightly decrease for time-windows greater than 1000 ms. In fact, *standstill* and *walking* are events that usually last for a shorter period of time, compared to other behavioral events, containing approximately 932 [800 – 1000] ms and 933 [866 - 1000] ms as median duration (Figure 5.2B). For this reason, they do not seem to benefit from long time-windows for accurate recognition. Furthermore, *walking* is the class with the lowest overall performance and one possible explanation could be the fact that *walking* is the class containing greater intra-class movement variability (either in terms of complexity of geometric shapes, sequences' durations and movement speeds) (Figure 5.2C.). The behavioral event that appears to be the most sensitive one to increasing the temporal extents is *grooming*. Using manual annotations given by the ethologists, this action is typically composed of several stationary periods interspersed with shorter periods of movement, in which the animal changes its position momentarily without leaving the *grooming* event. Long-term networks, with larger time-windows, can, thus, easily confuse *grooming* with *standstill* events (not shown), due to this heterogeneity within one single *grooming* sequence (one example is shown in Figure 5.2C., where a sequence of *grooming* frames was sampled at every 500 ms). On the other hand, *rearing* is the class with the highest performance for the different time-windows studied, not seeming to benefit from the increase in temporal extents. In fact, this is the less ambiguous behavior in the current classification task, because of its easy-to-distinguish geometric shape and lower depth values, and usually it is enough to analyze closer frames to confirm it.

4.1.2 Short-time scales are the most relevant for the learning process

Additionally, two variants of network encoder, single- and dual-branch, were systematically compared to study the impact of having temporal information of different scales. While in the standard single-branch networks the input is a time-sliding sequence with a fixed temporal stride between frames, this dual-branch network is fed with input sequences with different temporal strides in each pathway, as a way to understand if having multiple time scales helps in the learning process. To allow direct comparison, a single-branch architecture, with a time-window of 2τ and a temporal stride of 133 ms, and a dual-branch architecture, with different frame rate ratios α between the short- and long-time scale pathways, were trained and validated. The single-branch and dual-branch $\alpha = 5$ appear to have similar overall performances (Figure 5.3A), even for per-class recognition; however α equal to 10 (which means doubling the time-window for that pathway) seems to decrease performance. These results are in line with the conclusions of the previous section, where behavior learning does not seem to benefit from very distant temporal information (irrelevant frames are being taken into consideration, degrading network's performance).

4.1.3 Different input sequence's representations improve networks' learning

To further understand whether the temporal extent of video input sequences or their sampling frame rate with which the network is fed has more impact on learning rodents' behavior, networks with different temporal strides τ, but a fixed time window $T = 10\tau$, were also compared (Figure 5.3B). Significant improvements were observed when using higher frame rates (smaller temporal strides), with an increase of approximately 5% in the overall performance (with a frame rate equal to 14 fps, the median balanced accuracy reached 84.1% [83.0 - 86.2]%). In particular, *walking* and *grooming* events greatly benefit from increasing the input frame rate. This could indicate that a higher temporal resolution is needed to detect movement oscillations inherent to these types of heterogeneous behavioral events.

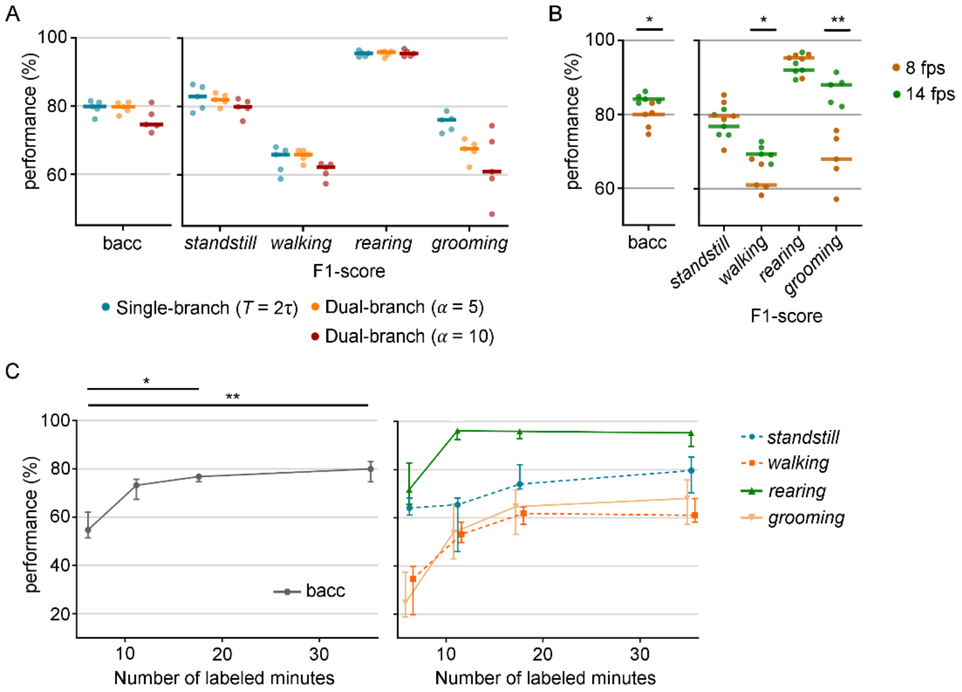


Figure 5.3 A. Which time scales are most relevant for the learning process? Comparison between architecture with different temporal scales: single-branch and dual-branch ($\boldsymbol{\alpha} = \mathbf{5}$ and $\boldsymbol{\alpha} = \mathbf{10}$), regarding overall balanced accuracy (bacc), and F1-score per class. **B. How should time be distributed to increase performance?** Comparison between different temporal strides $\boldsymbol{\tau}$ between adjacent frames ($\boldsymbol{\tau} \in \{\mathbf{67}, \mathbf{133}\}$ ms), corresponding to approximately 14 or 8 frames sampled per second, respectively). **C. How much information does the network need to learn?** Overall and per-class classification performance as function of number of labeled minutes. Data represented as median ± 95% confidence interval (N = 5 trials). * $\boldsymbol{p} < \mathbf{0.05}$; ** $\boldsymbol{p} < \mathbf{0.01}$. Statistical analysis only for overall balanced accuracy for the sake of readability. Additional statistical analysis on Supplementary Figure S 5.3.

As part of the networks' systematic study, the effects of input resolution and input depth encoding were also examined. The highest resolution (256x256) achieved the best results, with an overall performance of 85.9% [82.8 – 86.6]%. All behavioral events seem to benefit from increased resolution, in particular *grooming*, with an increase of approximately 44% over the lowest resolution (Supplementary Figure S 5.4A). When changing input depth encoding, networks trained with raw depth frames outperformed any other depth encoding techniques, with surface normal inputs reporting the worst performance, yielding an overall accuracy of 71.8% [60.9 - 75.8]% (Supplementary Figure S 5.4B and C).

4.1.4 High performances achieved with a reduced training dataset

In order to determine the approximate amount of annotated training data required for good network performance, the size of the training set was systematically varied (Figure 5.3C). As expected, overall performance increases for increasing number of training images. Even 10k labeled frames (approximately 21 mins of labeled data) were enough to achieve a good generalization, above 70%, with performance degradation in *walking* and *grooming* events. In fact, the effect of changing training size is most significant in these classes, where increasing 20 mins of annotated data leads to a gain of almost 45% in per-class performance. Peak performance was reached with 30k training examples (corresponding to approximately 1hour of labeled data).

4.1.5 Behavior is accurately detected in unseen depth videos

The behavior of the network against a completely unseen testing set is the ultimate study to quantify recognition performance and generalization capability of the model (Figure 5.4A). After being trained with the best set of parameters, the model achieved an overall accuracy of 82.2 % [78.5 – 83.9]%. Together with the ethograms automatically generated (Figure 5.4B), these results indicate that the proposed automated classification method captured the overall patterns of behavior in the new videos.

Regarding per-class performance, *rearing* is the behavioral event with the highest performance, attaining 87.2% [86.0 – 91.1]% F1-score, in accordance with previous results. Also, *walking* periods belong to the most misclassified behaviors, which are occasionally classified as *standstill* events (example in Figure 5.4A), given frames’ heterogeneity on shape and speed.

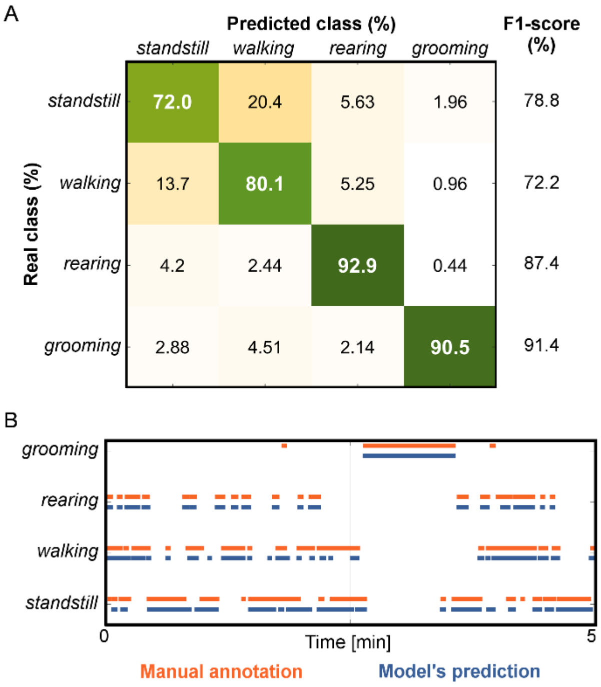


Figure 5.4 How does the best network behave for an unseen test set? A. Example of normalized confusion matrix for a detailed analysis of automated behavior recognition errors, and corresponding F1-scores for each class. **B.** Example of ethogram for a comparison between automated model's detection (orange) and manual annotation (blue), over 5 mins of testing video.

4.2 Automating closed-loop control of behavioral mazes: feedback approach

4.2.1 Closed-loop system achieves low-latency feedback based on animal behavioral/tracking patterns

In order to create a system capable of controlling a behavioral task based on animal behavior/position, it is necessary to close the loop between automatic detection of behavioral events and experimental operant conditioning hardware. A control platform, combining depth-sensor camera, computer and Arduino microcontroller was constructed to allow mapping of input-output control signals using the current deep learning detection of animal behavior and position. To demonstrate the applicability of the closed-loop framework in triggering signals based on animal behavior, an experiment was designed in which four actuators (in this case, LEDs) were turned on when the rat performed one of the four behavioral events: *standstill*, *walking*, *rearing* and *grooming*. The behaviors and tracking positions were automatically detected by previously trained deep networks, that, together with input signals coming from different sensors, are sent to the Arduino board to control the output devices (Figure 5.5C). This setup achieved delays from image acquisition to detecting the behavior+tracking position (image-event delay) as fast as 28.9 ms [26.95 – 31.86] ms, for an input resolution of 128x128 (Figure 5.5A). For larger images (256x256), the delay increased about 8.9% (full results from additional configurations can be found in Figure 5.5A). The proposed system, with the advanced hardware configuration (GPU settings) and for the smaller resolution, reached a performance time of 32.9 ms [32.8 – 34.9] ms from predicting one behavioral event+tracking position to the next one (event-event delay), including Arduino output generation, frame acquisition and processing, and behavior/tracking position detection. Finally, sending the signal to the Arduino board and sending back the signal to the computer took an additional 0.457 ms [0.457 – 0.460] ms, when compared to just turning on the LED – event-LED delay (0.914 ms [0.913 – 0.914] ms). Thus, the Arduino response is not constraining the runtime from event detection in one frame to the next frame, and it can be almost entirely attributed to intrinsic camera frame rate, behavior/tracking detection and additional processing.

4.2.2 User-interface allows end-to-end control of behavioral experiments

Acknowledging the importance of embedding all algorithms in a user-friendly application suited for research environments, we developed a full-featured, easy-to-use and freely available software interface (Figure 5.5B), requiring no programming by the end-user.

Behavior classification and/or tracking are performed using different available methods, chosen by the user, and detected using uploaded trained models. The GUI provides online information regarding hardware modules states, animal's behavior and position, allowing full control of the entire system. In particular, the state of 4 sensors and 4 actuators are updated in real-time, in which a LED-type icon is turned on upon the first image in which a behavioral pattern was detected, and subsequently turned off upon the first image in which the pattern is no longer detected (Figure 5.5C). This allows for a fully closed-loop stimulus' framework. The GUI also includes an option for users to upload an image containing ROIs for a more versatile and complete behavioral analysis. All useful information recorded during the experiment (depth frames, tracking and behavioral classes' information with sensors/actuators states for each timestamp) can be exported to a user-defined directory for further analysis.

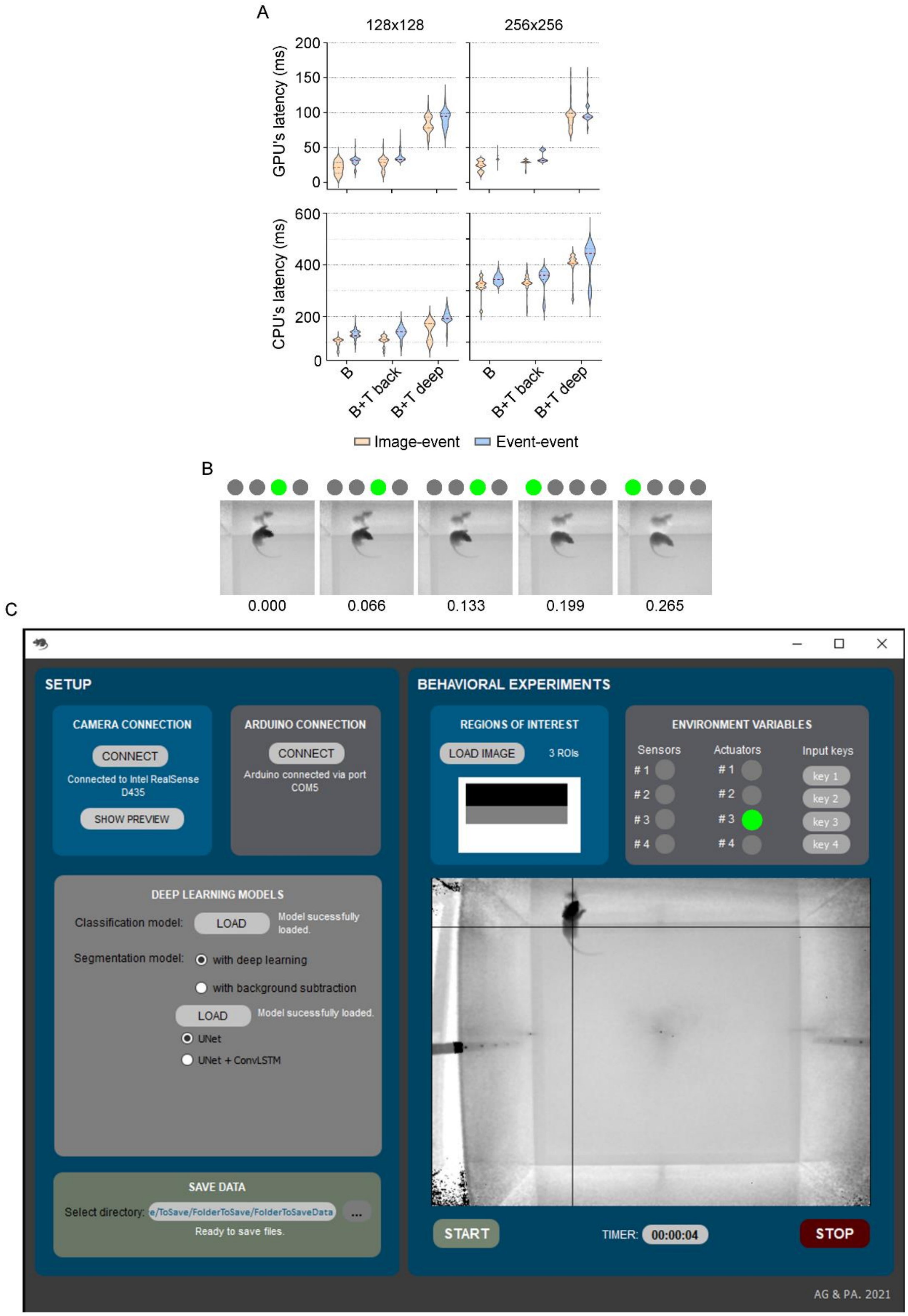
Overall, a cost-effective and easy-to-setup framework was created. The entire system consists of a computer running the GUI, connected to a depth camera (e.g., *Intel® RealSense* Depth Cameras, of ~300 €) and an Arduino (e.g. Mega 250, of ~35 €). Sensors and actuators can be directly connected to the Arduino board, and the quantity and type depend on each experiment's goal. The source code of the software, together with the user-guide manual, list of hardware materials and video examples, are publicly available for download at GitHub (https://github.com/CaT-zTools/Deep-CaT-z-Software).

5. Discussion

We have presented a fully integrated framework that can provide real-time feedback based on automated rodents' behavior classification and tracking position, using specialized deep Neural Networks (NN) to extract information from frames acquired with depth-sensing technologies.

With the developed algorithms, we demonstrate that cutting-edge deep learning models can be used to learn features from depth video sequences, without the need for feature-engineering approaches. In fact, this is one of the main reasons why deep learning-based methods can be more powerful than conventional behavior classification ones, avoiding user bias in the learning process and allowing for more easily tunable and generalizable systems. This is particularly important in basic research where environmental setups or animals' appearance/strains may be changed depending on the objectives of each experiment and yet it is possible to successfully apply the same methods (Anderson & Perona, 2014; M. W. Mathis & Mathis, 2020).

Furthermore, the capabilities of these deep learning networks were extended to learn feature representations exclusively from depth information. Although several deep learning-based studies have been published using depth frames for detecting human behavior, depth information is usually incorporated using multi-branch architectures, combining color and depth inputs from multiple streams for motion capture (Elboushaki, Hannane, Afdel, & Koutti, 2020; Singh, Khurana, Kushwaha, & Srivastava, 2020; L. Zhang et al., 2017).



Connected to Intel RealSense D435

Arduino connected via port COM5

Classification model: LOAD Model sucessfully loaded.

Segmentation model: ● with deep learning

○ with background subtraction

LOAD Model sucessfully loaded.

● UNet

○ UNet + ConvLSTM

Select directory: e/ToSave/FolderToSave/FolderToSaveData

Ready to save files.

TIMER: 00:00:04

Figure 5.5 How to close the loop for behavioral experiments? A. Latencies, in milliseconds (ms), from image acquisition to obtaining an event (image-event) and from the

last event detected to the current event detected (event-event), using CPU or GPU processing. Latencies were estimated for automated predictions of behavior only (B), behavior and tracking using the background subtraction method (B + T back), and behavior and tracking using a deep model-based method (B + T deep). The width of the violin plots represents the probability density of the data, with the median and 95% confident interval represented as red and black dashed lines. **B.** Example of a *rearing* followed by a *walking* sequence, with corresponding LED status (as it appears in the graphical user interface), from the test video sequence. Image timestamps in seconds are presented at the bottom of each image. **C.** Graphical user interface for automating real-time closed-loop behavioral experiments.

Here, we focused on depth images and how information can be successfully retrieved for animal behavior extraction. Analyzing behavior with only depth information has four important advantages. Since these frames are acquired by infrared sensors, videos can be recorded in dark conditions (where color information is useless) without disrupting animals' natural behavior (mainly in nocturnal animals, such as rodents). Also, with this technology, color contrast between the animal and the background is no longer a problem for detection/tracking purposes. Conventional methods usually use markers or methods dependent on animals' color coating (Hong et al., 2015; Machado et al., 2015; Ohayon et al., 2013; Pérez-Escudero et al., 2014; Unger et al., 2017), which can be avoided using depth-sensing information. In addition, 3D information can be retrieved from a single camera, and so setting complicated stereo-vision setups is no longer needed. Finally, to further facilitate the integration of computational methods in the laboratory and industry fields, low-cost acquisition devices are required, combined with good performance and, at the same time, quick data acquisition and low computational cost. Therefore, the use of depth technology, such as *Kinect*-based cameras, showed to be an alternative strategy to be applied in behavioral experiments. Since there are no state-of-the-art studies exploring the use of depth information in the context of feature extraction for animal behavior classification, we also perform a systematic study to understand the best ways to represent network inputs and how we can improve models' performance. By using deep learning networks that incorporate spatiotemporal features, it was possible to conclude that temporal information is very relevant for learning animal behavioral patterns, especially in some classes (*standstill* and *walking*, which contain a strong dynamic component). These results are in agreement with the fact that temporal information of video data can provide additional clues hidden in temporally neighboring frames for the recognition of actions/behaviors or segmentation of frames (Elboushaki et al., 2020; Simonyan & Zisserman, 2014). By using a fixed temporal stride between input frames of approximately 133 ms, the performance of networks is significantly improved for input video sequences with a time-window of approximately 1.5 seconds. As expected, some animal behaviors are of very short duration, with rapid transitions, sometimes imperceptible by humans, and for this reason, deep NN for animal behavior classification must be carefully designed to support finer temporal analyses. In addition, results showed that neither long-time scales nor multi-scales seemed to be advantageous for detecting animal behavior. One possible explanation is that long-time scales include frames too far apart in time, containing irrelevant information to learn useful feature representations for the current frame. Although with our system we didn't see advantages in the multi-scale analysis, we hope that it can be further explored in the context of animal behavior. For example, in a system with higher frame rates, it may be useful to also explore shorter time scales.

Along with the fact that higher resolutions and higher sampling rates in raw frames (without preprocessing or encoding) significantly improve the performance of proposed deep networks, the results give an insight on how to build, train and fine-tune networks to better learn rodent behavior using depth-sensing information. Finding that ~21 mins of annotated videos are already sufficient to achieve high generalization rates strengthens the contributions of the proposed system since a core goal of automating the analysis of behavior is reducing the manual annotation effort. In this sense, once the deep learning model is trained, the system is ready to assist in any behavioral experiment without additional user-time, allowing for more reproducible results and reducing variability imposed by inter-human annotations. Recent works have made some progress toward the goal of supervised classification of rodents' behavior using deep learning techniques to improve conventional feature-engineering-dependent methods. M. Marks et al. (2020) developed *SIPEC:BehaveNet* for behavior recognition, which was tested in a dataset acquired with a conventional camera and containing freely behaving mice whose behavior was labeled with only 3 classes (Sturman et al., 2020). Although claiming superior performance to Sturman et al. (2020) proposal, *SIPEC:BehavNet* achieved lower overall performances for *supported rearing* and *grooming* events (mean ± standard error of the mean: 0.84 ± 0.04 and 0.49 ± 0.21, respectively), when compared to what we were able to report here. *DeepEthogram* is another recent tool for frame-based classification of animal behavior in RGB videos

(Bohnslav et al., 2021). High overall performances (overall accuracy) were obtained for datasets containing mice behavior with more than 4 classes. However, performance per-class (F1-score) is substantially impaired for some behaviors, in particular, the rarest and most challenging behaviors in the dataset (average F1-score below 70%). This shows evidence that attention must be paid to metrics performance when dealing with highly unbalanced datasets. Overall, both methods fall behind some strengths that our method shows, needing more than 70 mins of labeled data to achieve a comparable performance (overall accuracy above 70%) and not being suitable for natural environmental conditions in the analysis of rodents' behavior.

In order to improve the potential of the proposed system and create an integrated tool that would boost future development in understanding behavioral patterns and neuronal activity relationship, deep learning-based detection of behavior was used to provide event-triggered feedback in real-time. The loop between animals' maze, depth frames acquisition, and automatic streaming of behavioral patterns was closed using input and output devices connected to an Arduino microcontroller. From detecting one behavioral event to the next event in a consecutive frame, the system was able to achieve real-time feedback control, with latencies of less than 33 ms with GPU-based configuration. These results are below the frame rate of the camera used (which typically is reduced to ~15 fps in low light conditions), and so, in theory, more powerful cameras could be tested. Research on developing real-time applications for neuroscience research has been advancing in recent years. However, efforts have essentially been directed towards tools to detect animal's posture, rather than classifying directly the behavior. Both Forys et al. (2020) and Schweihoff et al. (2021) developed software and hardware to enable real-time estimation of mice posture, and achieved latencies of 30ms using comparable computational configurations, from frame acquisition to detecting a posture of interest (slower image-event delay than what we were able to achieve) (Forys et al., 2020; Schweihoff et al., 2021). Kane et al. (2020) reported higher computational performances for the same task, with a 16ms delay from image-LED event (for equivalent image resolution and hardware configurations) . However, it is worth emphasizing that our 30-fps figure is achieved when both behavior classification and tracking position are available, which gives the tool versatility for different research applications. To the best of authors' knowledge, Nourizonoz et al. (2020) were the first to try to detect animal postures as well as simple behaviors in naturalistic environments, using multiple cameras with infrared-based technology. Real-time detections were achieved to enable reinforcing a simple behavior (*rearing*) by operant conditioning. Although with high performance in naturalistic environments and taking the first steps in moving forward to correlate posture with neural circuits by optogenetics stimulation, the detection of a single behavior from posture was achieved using a set of geometrical rules. This approach may not be sufficient to classify more sophisticated behaviors, or computationally heavier when classifying multiple behaviors.

A key aspect of the design of the whole system is its versatility and how different modules can be adapted to different research goals. In particular, several tracking algorithms were made available, depending on model's performance and computational power. This flexibility may be important when real-time detection is not required but offline high-performance detection is needed. Also, many sensors and actuators can be easily adapted to the Arduino microcontroller to finer control of animal's maze, and the automation control code is prepared to be further extended. Even so, recent advances in multiple animal behavior analysis and tracking (de Chaumont et al., 2019; Pérez-Escudero et al., 2014; Romero-Ferrero et al., 2019) could be included to further enhance this versatility. System adaptation is, in theory, straightforward, however, the triggers for feedback control need to be carefully designed when dealing with complex social behavior. Furthermore, the list of behavioral events/classes can be further extended. Here, the potential of deep NN can be explored, since they are able to extract relevant features without the need for feature engineering, unlike conventional machine learning methods.

Taking all the contributions together, we believe that the flexibility and yet easy-to-use characteristics of this real-time feedback framework may open the door to further studies and broader applications, allowing more high-throughput and rigorous behavioral experiments while less invasive for laboratory animals.

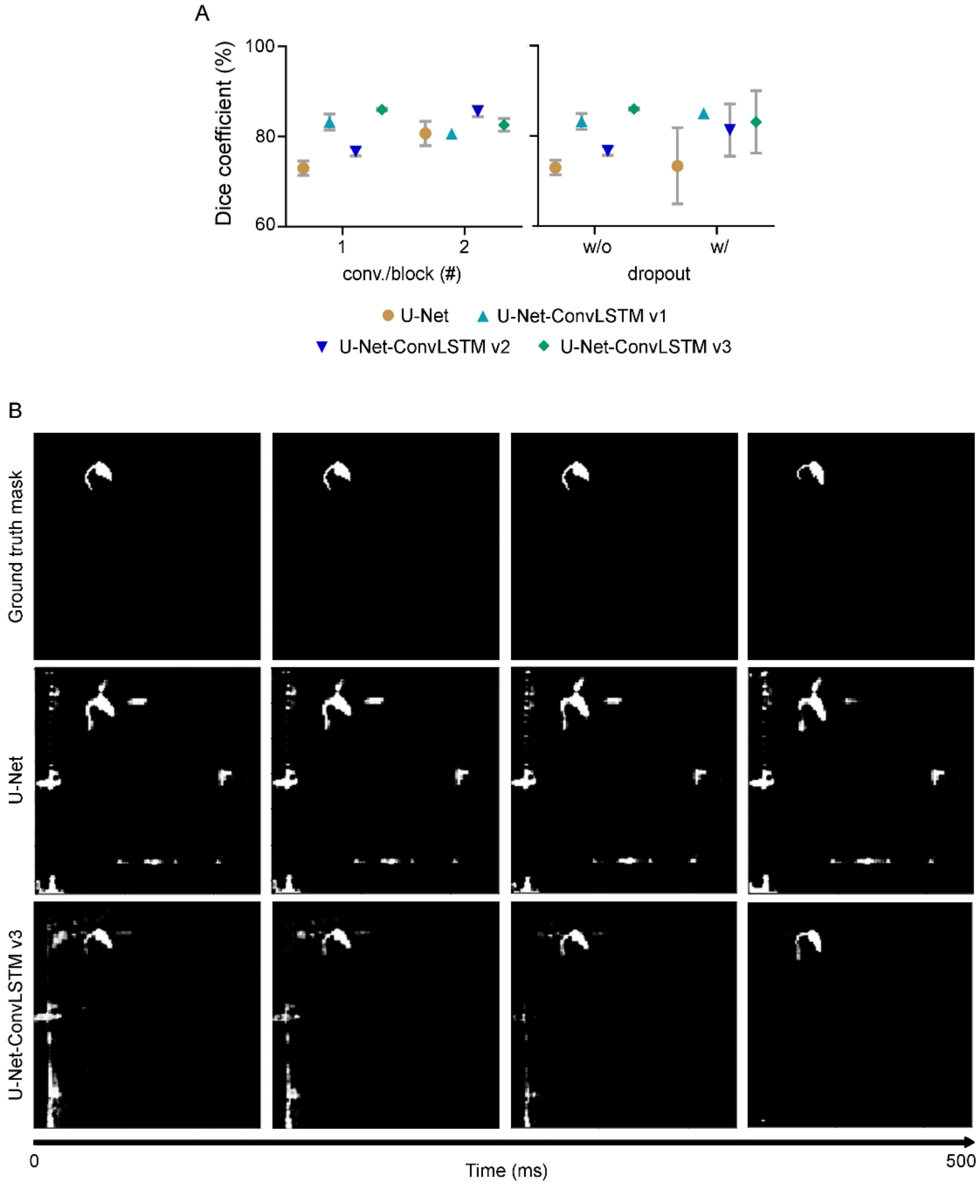
6. Conclusions

We present a versatile and real-time software solution using deep learning techniques on depth video sequences for the automatic detection of behavioral events. Using feature extraction on depth information introduces a new paradigm in behavioral neuroscience by allowing recordings and analysis of natural animals' movements with markerless and contrast-free deep learning methods. When combined in a feedback control system, it automates communication using closed-loop signals directly dependent on current detected behavioral patterns. By creating conditions for low-cost, high-throughput analysis and reproducible quantitative measurements of animal behavior experiments, we anticipate that this tool will contribute to faster and more reliable investigation in ethology and neuroscience fields.

7. Supplementary Information

7.1 Extended Methodology

7.1.1 Semantic segmentation performance using U-Net and U-Net-ConvLSTM networks

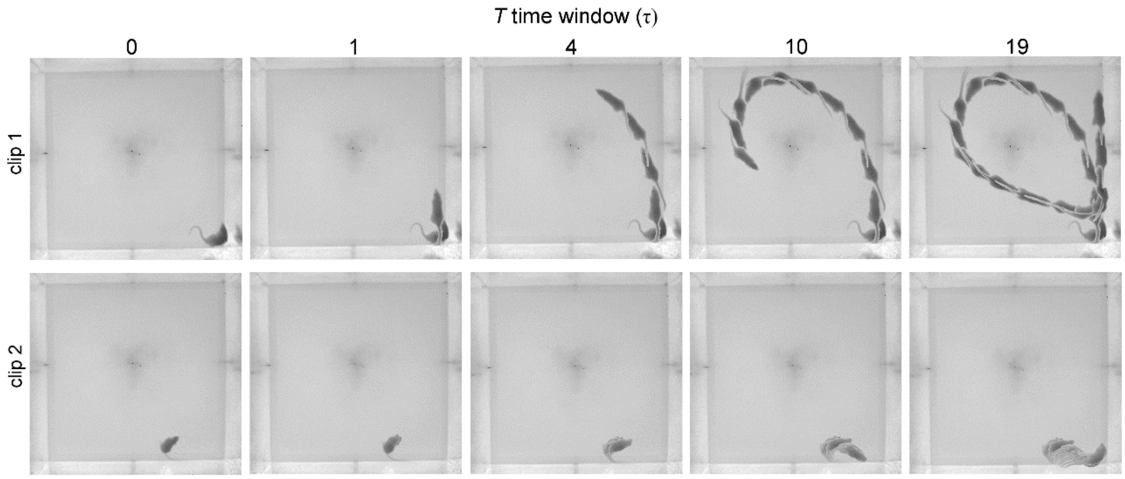


Supplementary Figure S 5.1 Semantic segmentation results of U-Net-based networks. A. Networks' performance in terms of Dice coefficient for different architectural parameters. Left: number of convolutional layers per block; Right: networks without (w/o) and with (w/)

dropout layer at the end of the encoder. The traditional U-Net architecture was extended by placing a ConvLSTM layer at different positions in the network (U-Net-ConvLSTM), in order to find which position is most suitable for the depth images segmentation task (following Pfeuffer, Schulz, and Dietmayer (2019) methodology). U-Net-ConvLSTM version 1 (v1) – ConvLSTM layer placed between the encoder and the decoder. U-Net-ConvLSTM version 2 (v2) – ConvLSTM layer placed in the end of the network. U-Net-ConvLSTM version 3 (v3) – a combination of the last two versions. Data represented as median ± 95% confidence interval (N = 2 trials). **B.** Sample clips representing original (top) and predicted segmentation masks by the U-Net (middle) and U-Net-ConvLSTM v3 (bottom) networks, for a time window of 500 ms. Black pixels represent the background predictions and white pixels represent foreground (animal) predictions. During the inference, the presence of ConvLSTM layers improves the segmentation masks over time.

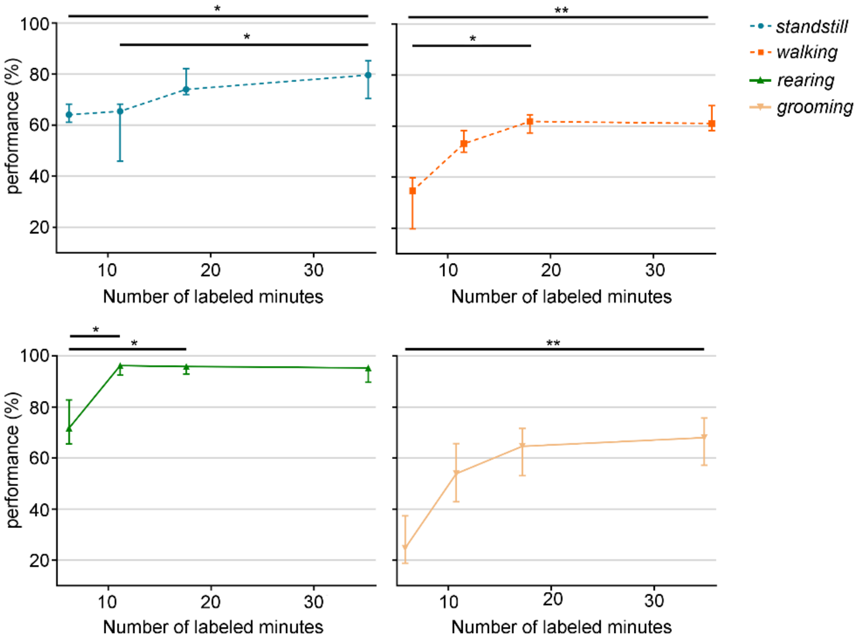
7.2 Extended Results

7.2.1 Past information improves behavioral classification performance



Supplementary Figure S 5.2 How much temporal information does the network need for rodents' behavioral learning? Stroboscopic montages in which each animal position represents raw depth frames extracted at every 133 ms, for 2 different walking clips and different time windows T, in units of τ ($\tau = 133$ ms). Each stroboscopic image illustrates the depth video sequence input fed to the deep learning network for different values of T.

7.2.2 High performances achieved with a reduced training dataset



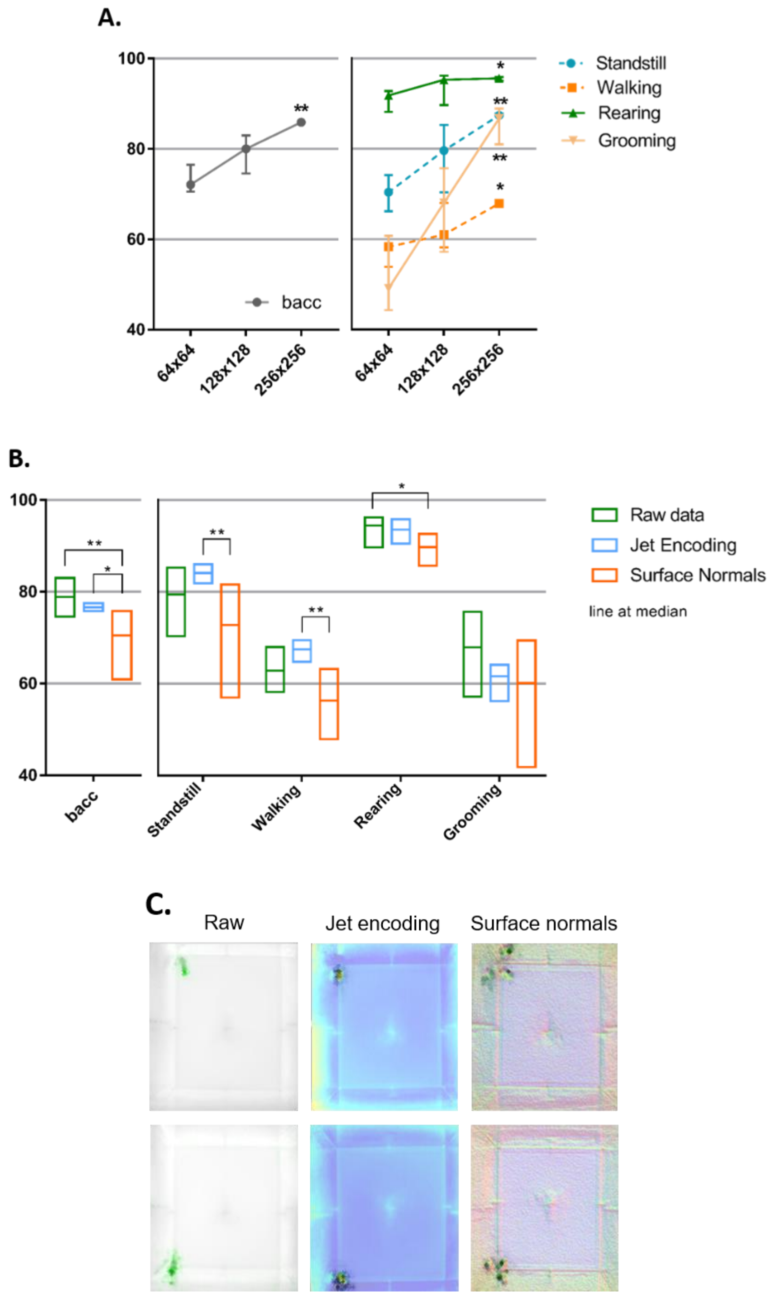
Supplementary Figure S 5.3 How much information does the network need to learn? Extended statistical analysis for per-class classification performance as function of number of labeled minutes. Data represented as median ± 95% confidence interval (N = 5 trials). * $p < 0.05$; ** $p < 0.01$.

7.2.3 Input resolution improves behavioral classification performance

As part of the networks' study, the effect of input resolution was also examined, keeping the single-branch architecture with default parameters (Supplementary Figure S 5.4A). As expected, the highest resolution (256x256) achieved the best results, with an overall performance of 85.9% [82.8 – 86.6]%. All behavioral events seem to benefit from increased resolution, in particular *grooming*, with an increase of approximately 44% over the lowest resolution. The fact that *grooming* events seem to need both higher temporal and spatial resolutions makes it the most sensitive and complex behavior to recognize.

7.2.4 Raw depth video inputs are the most informative for the learning process

Depth data encodes distance from the sensor to the captured scene and the information of each pixel is of a different nature than the RGB counterparts (originally directly used as input for the CNNs). Thereby, the questions that arise are will CNNs learn as effectively when using raw depth images without any encoding? If not, how should a depth image be encoded to be used as inputs in CNNs so that it can learn more meaningful features for rodents' classification challenge? Networks were then trained with varying input depth encoding (Supplementary Figure S 5.4B). Regarding per-class recognition, the negative effect on network's learning when using surface normal encoding is even more pronounced. One possible explanation is that when using a colorization method based on the calculation of surface normal, the reflexes on the walls of the open-field (OF) during, for example, *grooming* events (which are always near OF's periphery) are more visible and may be interfering with networks' learning. Sensitivity analysis can be used to identify the most relevant input features during the learning process, by calculating heatmaps from pixel-wise normalized gradients (derivative of class model's predictions with respect to pixel values). This impact on model's prediction is exemplified on Supplementary Figure S 5.4C, where, by using surface normals, periphery pixels seem to have a stronger influence on model's prediction (gradient colored as black pixels), when compared to pixels from networks trained with raw depth frames (gradient colored as green pixels). Overall, behavioral learning does not seem to benefit from any of these typical depth input representations.



Supplementary Figure S 5.4 Which input sequence representation is most informative for network's learning? A. Recognition performance of the single-branch architecture with different input resolutions. * and ** denote statistical significance when compared to the lowest resolution (64x64). **B.** Different depth encodings and corresponding performance, when compared to raw depth input frames. Data represented as median ± 95% confidence interval (N = 5 trials). * $p < 0.05$; ** $p < 0.01$. **C.** Sensitivity analysis for different depth encoding methods (two different frames are shown), with gradients in green or black.

CHAPTER 6

Main Conclusions and Future Perspectives

1. Main Conclusions

Animal behavioral experiments are a fundamental mainstay in neuroscience research, where behavioral patterns' characterization is used to assess the effects of pharmacological manipulations, rehabilitation protocols, neurological diseases, etc. (Berman, 2018; Krakauer et al., 2017). Unfortunately, animal behavior experiments are still extremely complex to analyze and often end up relying on human judgment for manual quantification and classification, which brings strong subjectivity, high costs, and reduced reproducibility. Computational ethology has appeared to automate behavior analysis and ultimately accelerate the development of novel therapies for prevalent and devastating human diseases, such as neurodegenerative diseases, cancer, and mental illnesses (Anderson & Perona, 2014; Egnor & Branson, 2016). Although solutions already exist to improve the conventional manual methods for behavior analysis through machine vision and machine learning techniques, these approaches are still insufficient for a complete and effective classification of animal behavior (Egnor & Branson, 2016; M. W. Mathis & Mathis, 2020; von Ziegler et al., 2021; Zilkha et al., 2016). Aiming to address some important unsolved challenges in computational ethology, this thesis proposes different machine vision and machine learning-based techniques to probe and analyze animal behavior in an automated and reproducible way.

Behavioral researchers, along with computational neuroscientists, have developed over the last years an immense array of experimental methods to study complex behaviors for different applications (analysis of locomotion, anxiety, depression, cognitive functions, animal welfare, etc.). However, a limitation shared among most of these experimental paradigms is the study of behavioral patterns in very restricted environments (e.g., with static backgrounds), or using invasive or stressful techniques. Besides diverging from ethologically-relevant behaviors, even the mildest disturbances in environmental conditions can impact on behavioral outcomes (Reardon, 2016). Alternative non-invasive methods that allow capturing animal behavior in enriched and dynamic environments were presented in CHAPTER 3. Here, the potential of thermal infrared cameras was initially studied for the overall assessment of mean body surface temperature (MBST) in freely-moving mice, and dedicated software was developed for thermal image analysis. We showed that thermography allows for non-invasive thermal assessment of laboratory animals, avoiding the effect of handling stress on animals' physiology or behavior.

Depth cameras with infrared technology are another interesting option over conventional optical sensors, and their applicability in capturing animal behavior under naturalistic lighting conditions was first presented in CHAPTER 4. The proposed system allows segmentation and tracking in dark/low contrast conditions: even when the background color matches the animal's fur (impairing color segmentation), segmentation in depth is still possible. Besides that, machine vision-based methods for depth segmentation in enriched and dynamic environments were proposed and tested for a freely-walking rat in an open-field setup. Another important step was taken towards improving the reproducibility of behavioral experiments by creating a benchmark dataset that could be used by the scientific community (CHAPTER 4). In fact, we introduced the first published RGB-D rat behavioral dataset, which can be further used to train and test automatic behavior recognition systems in rodents, and improve existing machine learning-based methods in the computational neuroscience field.

The precise estimation of animals' pose in two-dimensional (2D) video-analysis systems is very limited, impairing detailed pose characterization, and, later, accurate and precise behavior classification. Although some studies have therefore started to address the problem in three-dimensions (3D), limitations such as the use of markers to distinguish the animals (de Chaumont et al., 2019; Hong et al., 2015), human interventions (de Chaumont et al., 2019; Matsumoto et al., 2013), or equipment and setup of high cost (Dunn et al., 2021; Günel et al., 2019; Matsumoto et al., 2013), makes these solutions far from being completely integrated into a laboratory environment. With the fundamental study presented in CHAPTER 4, 3D segmentation and tracking of multiple body parts were made possible thanks to the combination of depth-sensing technology and machine vision techniques. Five anatomical points were detected (nose, head, body center (centroid), tail-base and tail-end), using scale-free geometrical constraints/properties. Semantic segmentation of animals' whole-body using depth information was further improved with the methods presented in CHAPTER 5, using deep learning techniques to improve model's performance and avoid human interventions during the feature-engineering process. Here, the traditional U-Net architecture (Ronneberger, Fischer, & Brox, 2015) was evaluated in its ability to semantically segment purely depth videos. Also, this architecture was extended by placing a Convolutional Long Short-Term Memory (ConvLSTM) layer at different positions in the network (Pfeuffer et al., 2019), to both take advantage of the temporal information present in contiguous depth frames and find which layer’s position is most suitable for the depth segmentation task.

To create integrated systems in which it is possible to perform multiple tasks for different behavioral experiments, two computational solutions were created (described in CHAPTER 4 and CHAPTER 5). These solutions, in addition to the segmentation and tracking of the animal, allow performing automatic classification of behavioral patterns. This outcome is important when tracking alone is not enough for the complete characterization of animals’ behavior. Furthermore, the ablation studies performed for the automatic classification task unraveled important clues on how machine learning classifiers and deep learning networks should be constructed to improve learning performance of rodents’ behavior. Here, besides spatial information encoded in depth frames, temporal information showed to be crucial for the learning process, either in terms of temporal windows used as networks’ inputs or in the frame rate of the input video sequences. In this sense, with this new information, studies that intend to explore the dynamics of laboratory animals’ behavior must bear in mind its temporal characteristics and adapt their models accordingly to gain further useful insights on behavioral patterns of interest. The features of the computational tool initially presented in CHAPTER 5 have been extended to increase the range of applications in behavioral experiments. Whether the objective is to control behavioral maze modules, such as feeders or levers, or real-time drug delivery protocols, the proposed tool now allows for real-time recognition of animal behavioral patterns and feedback control of sensors/actuators in any maze for high-throughput behavioral experiments. An Arduino microcontroller was used as the interface board between the computer and any hardware modules, and a communication protocol was defined to allow sending positional, behavioral, and modules status information between the computer and the microcontroller.

In order to facilitate integration in laboratory environments, the three computational solutions, described in CHAPTER 3, CHAPTER 4, and CHAPTER 5, were designed to be easy to install (only the computer and the infrared camera are necessary; in the case of the feedback control system, a connection of the control modules to a microcontroller is required), and easy to use (graphical user interfaces (GUI) were carefully designed to guide the acquisition and processing, without the need for extra programming knowledge). Altogether, they provide tools for analyzing and quantifying animal behavior in an automatic, user-friendly, standardized, and reproducible way, all essential characteristics in the context of experimental neuroscience.

With the promising advances that lie ahead in the coming years, mainly in the fields of machine vision and machine learning, behavioral neuroscience is expected to become more and more quantitative. In fact, this is the only way to progress as to ensure reproducibility and standardization in animal behavioral experiments. This thesis has contributed to the computational ethology field with novel tools for the automatic characterization of animal behavior. High-throughput behavioral experiments are now possible thanks to the use of non-invasive techniques, capable of working in dark environments (such as thermography and depth-sensing technologies), as well as in dynamic and enriched backgrounds. Behavioral events can be detected using state-of-the-art machine vision and machine learning methods and control operant mazes in real-time analysis. Having this in mind, the author believes that these computational tools boost subsequent investigations, and accelerate the understanding of behavioral mechanisms, either in neuroscience research or industrial environments.

2. Future Perspectives

Over the last years, the computational ethology field has undergone remarkable advances, in particular in the past two years, where a rapid innovation in computational methods and techniques has been noted thanks to the integration of deep learning tools in laboratory experiments. At the same time, technology has evolved to provide more accurate and sophisticated hardware (acquisition sensors, maze modules, graphics processing unit (GPU) boards, etc.) that can now be used to acquire and process information in a faster and more robust way. When combined, these advances bear the potential not only to provide larger scale and standardized automated analysis but also to increase the quality of extracted data and to provide unparalleled power to reveal novel patterns and biological mechanisms.

Although the work developed in this thesis addressed some important challenges concerning the automatic quantification of animal behavior, some questions were raised that still need further attention, as well as new directions for future studies. With the plethora of behavioral assays and experimental setups available to analyze animal behavior comes the difficulty in creating robust methods both to changes in the environmental conditions (background, behavioral apparatus, lighting conditions, cameras, etc.) and to animals' appearance. The generalization capability of the machine learning classifier explored in

CHAPTER 4 was tested by using different lighting conditions (dim red light, dim white light, and total darkness) and a different apparatus (elevated plus maze (EPM)) for phenotyping Wistar and Wistar Kyoto (WKY) rats. Nevertheless, further studies are necessary to extend this capability to different animal strains and behavioral apparatus. Besides data augmentation, which is a popular strategy to increase the robustness of deep learning networks and that was explored in CHAPTER 5, transfer learning is an alternative to improve out-of-domain performance. Another advantage of using such techniques is the reduction of dataset sizes. This is an important challenge for applying machine learning methods to neuroscience since deep learning networks typically require large-scale datasets, which are not readily available for laboratory experiments. Semi-, weakly- or self-supervised learning are also emerging research directions in machine learning that applied to increase generalization even with little data. Recently, such methods have surpassed the performance of some state-of-the-art supervised methods in different areas, such as image recognition (T. Chen, Kornblith, Norouzi, & Hinton, 2020) and speech processing (Baevski, Zhou, Mohamed, & Auli, 2020; Ravanelli et al., 2020), and since they are able to reduce the amount of labeled data significantly, they are promising methods to be applied in computational ethology (Y. LeCun et al., 2015; von Ziegler et al., 2021).

Considering the social component of animal behavior, many experiments in neuroscience require the detection and measurement of multiple actions and interactions between animals. In this thesis, the behavior of a single animal was explored; however, if the objective is to quantify social behavior, the proposed methods would have to be further adapted to more than one animal. Recently, deep learning approaches have been used to solve the multi-object' tracking task. For that, a methodology followed by C. Romero-Ferrero et al. (2019) could be adapted to work with depth-sensing images. For centroid tracking over time, the authors used two Convolutional Neural Networks (CNNs) to first detect if each pre-segmented mask corresponds to a single animal or a crossing, and then to identify each individual between two crossings. With this approach, several unmarked animals can be tracked at the same time, and such information could then be used to further analysis or automatic classification of behavior. An alternative followed by several studies is to apply a popular and fast object localization network, *You Only Look Once* (YOLO) network, for multiple animals' detection (Arac et al., 2019; Z. Chen et al., 2020), and then combine it with different pose estimation packages for tracking during social behaviors.

Finally, the deep learning approach for automatic classification of behavior brought advantages in reducing human interventions and increasing overall performance for the classification of 4 classes. Nevertheless, there is still room to extend the proposed set of behavioral events. As opposed to the traditional machine learning techniques, deep Neural Networks (NN) allow for more versatility both in terms of architecture and optimization, and, for that reason, the increasing knowledge on this field will continue to pave the way for an improved study of animal behavioral patterns.

REFERENCES

Adriaan Bouwknecht, J., Olivier, B., & Paylor, R. E. (2007). The stress-induced hyperthermia paradigm as a physiological animal model for anxiety: A review of pharmacological and genetic studies in the mouse. *Neuroscience & Biobehavioral Reviews, 31*(1), 41-59. doi:http://dx.doi.org/10.1016/j.neubiorev.2006.02.002

Aguiar, P., Mendonca, L., & Galhardo, V. (2007). OpenControl: a free opensource software for video tracking and automated control of behavioral mazes. *J Neurosci Methods, 166*(1), 66-72. doi:10.1016/j.jneumeth.2007.06.020

Aguiar, P., Mendonça, L., & Galhardo, V. (2007). OpenControl: a free opensource software for video tracking and automated control of behavioral mazes. *Journal of neuroscience methods, 166*(1), 66-72.

Ahrendt, P., Gregersen, T., & Karstoft, H. (2011). Development of a real-time computer vision system for tracking loose-housed pigs. *Computers and Electronics in Agriculture, 76*(2), 169-174. doi:10.1016/j.compag.2011.01.011

Almeida, M. C., Steiner, A. A., Branco, L. G., & Romanovsky, A. A. (2006). Cold-seeking behavior as a thermoregulatory strategy in systemic inflammation. *Eur J Neurosci, 23*(12), 3359-3367. doi:10.1111/j.1460-9568.2006.04854.x

Anderson, D. J., & Perona, P. (2014). Toward a science of computational ethology. *Neuron, 84*(1), 18-31. doi:10.1016/j.neuron.2014.09.005

Arac, A., Zhao, P., Dobkin, B. H., Carmichael, S. T., & Golshani, P. (2019). DeepBehavior: A Deep Learning Toolbox for Automated Analysis of Animal and Human Behavior Imaging Data. *Front Syst Neurosci, 13*, 20. doi:10.3389/fnsys.2019.00020

Ardekani, R., Biyani, A., Dalton, J. E., Saltz, J. B., Arbeitman, M. N., Tower, J., . . . Tavare, S. (2013). Three-dimensional tracking and behaviour monitoring of multiple fruit flies. *J R Soc Interface, 10*(78), 20120547. doi:10.1098/rsif.2012.0547

Attanasi, A., Cavagna, A., Del Castello, L., Giardina, I., Jelic, A., Melillo, S., . . . Viale, M. (2013). Tracking in three dimensions via multi-path branching. *CoRR abs/1305.1495*.

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011). *Sequential deep learning for human action recognition.* Paper presented at the International workshop on human behavior understanding.

Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Bailoo, J. D., Reichlin, T. S., & Würbel, H. (2014). Refinement of Experimental Design and Conduct in Laboratory Animal Research. *ILAR journal, 55*(3), 383-391. doi:10.1093/ilar/ilu037

REFERENCES

Bains, R. S., Cater, H. L., Sillito, R. R., Chartsias, A., Sneddon, D., Concas, D., . . . Armstrong, J. D. (2016). Analysis of Individual Mouse Activity in Group Housed Animals of Different Inbred Strains using a Novel Automated Home Cage Analysis System. *Front Behav Neurosci, 10*, 106. doi:10.3389/fnbeh.2016.00106

Baker, M. (2011). Inside the minds of mice and men. *Nature, 475*(7354), 123-128.

Bauer, P. R. (2002). Microvascular responses to sepsis: clinical significance. *Pathophysiology, 8*(3), 141-148. doi:https://doi.org/10.1016/S0928-4680(02)00007-X

Bautista, A., Zepeda, J. A., Reyes-Meza, V., Féron, C., Rödel, H. G., & Hudson, R. (2017). Body mass modulates huddling dynamics and body temperature profiles in rabbit pups. *Physiology & behavior, 179*, 184-190. doi:https://doi.org/10.1016/j.physbeh.2017.06.005

Belzung, C. (1999). .11 Measuring rodent exploratory behavior. In *Techniques in the behavioral and neural sciences* (Vol. 13, pp. 738-749): Elsevier.

Ben-Shaul, Y. (2017). OptiMouse: a comprehensive open source program for reliable detection and analysis of mouse body and nose positions. *BMC Biol, 15*(1), 41. doi:10.1186/s12915-017-0377-3

Bengio, Y. (2009). *Learning deep architectures for AI*: Now Publishers Inc.

Berman, G. J. (2018). Measuring behavior across scales. *BMC Biol, 16*(1), 23. doi:10.1186/s12915-018-0494-7

Berman, G. J., Choi, D. M., Bialek, W., & Shaevitz, J. W. (2014). Mapping the stereotyped behaviour of freely moving fruit flies. *J R Soc Interface, 11*(99). doi:10.1098/rsif.2014.0672

Bishop, C. M. (2006). Pattern recognition. *Machine learning, 128*(9).

Blanqué, R., Meakin, C., Millet, S., & Gardner, C. R. (1996). Hypothermia as an indicator of the acute effects of lipopolysaccharides: comparison with serum levels of IL1β, IL6 and TNFα. *General Pharmacology: The Vascular System, 27*(6), 973-977. doi:https://doi.org/10.1016/0306-3623(95)02141-8

Boenisch, F., Rosemann, B., Wild, B., Dormagen, D., Wario, F., & Landgraf, T. (2018). Tracking all members of a honey bee colony over their lifetime using learned models of correspondence. *Frontiers in Robotics and AI, 5*, 35.

Bohnslav, J. P., Wimalasena, N. K., Clausing, K. J., Dai, Y. Y., Yarmolinsky, D. A., Cruz, T., . . . Woolf, C. J. (2021). DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels. *Elife, 10*, e63377.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). *A training algorithm for optimal margin classifiers.* Paper presented at the Proceedings of the fifth annual workshop on Computational learning theory.

Branson, K., Robie, A. A., Bender, J., Perona, P., & Dickinson, M. H. (2009). High-throughput ethomics in large groups of Drosophila. *Nature Methods, 6*(6), 451-457.

Braun, E., Geurten, B., & Egelhaaf, M. (2010). Identifying prototypical components in behaviour using clustering algorithms. *PLoS One, 5*(2), e9361. doi:10.1371/journal.pone.0009361

Brooks, S. P., & Dunnett, S. B. (2009). Tests to assess motor phenotype in mice: a user's guide. *Nature Reviews Neuroscience, 10*(7), 519-529.

Burgos-Artizzu, X. P., Dollár, P., Lin, D., Anderson, D. J., & Perona, P. (2012). *Social behavior recognition in continuous video.* Paper presented at the Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.

Burke, N. N., Coppinger, J., Deaver, D. R., Roche, M., Finn, D. P., & Kelly, J. (2016). Sex differences and similarities in depressive- and anxiety-like behaviour in the Wistar-Kyoto rat. *Physiol Behav, 167*, 28-34. doi:10.1016/j.physbeh.2016.08.031

Byeon, W., Breuel, T. M., Raue, F., & Liwicki, M. (2015). *Scene labeling with lstm recurrent neural networks.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Cachat, J., Stewart, A., Utterback, E., Hart, P., Gaikwad, S., Wong, K., . . . Kalueff, A. V. (2011). Three-dimensional neurophenotyping of adult zebrafish behavior. *PLoS One, 6*(3), e17597. doi:10.1371/journal.pone.0017597

Całkosiński, I., Dobrzyński, M., Rosińczuk, J., Dudek, K., Chrószcz, A., Fita, K., & Dymarek, R. (2015). The Use of Infrared Thermography as a Rapid, Quantitative, and Noninvasive Method for Evaluation of Inflammation Response in Different Anatomical Regions of Rats. *BioMed Research International, 2015*, 9. doi:10.1155/2015/972535

Cantzler, H. (1997). An overview of range cameras. Retrieved from http://homepages.inf.ed.ac.uk/rbf/CVonline/CVentry.htm

Che, W., & Peng, S. (2018). *Convolutional LSTM Networks and RGB-D Video for Human Motion Recognition.* Paper presented at the 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC).

Chen, C., Liu, K., & Kehtarnavaz, N. (2016). Real-time human action recognition based on depth motion maps. *Journal of real-time image processing, 12*(1), 155-163.

Chen, C., Liu, M., Liu, H., Zhang, B., Han, J., & Kehtarnavaz, N. (2017). Multi-temporal depth motion maps-based local binary patterns for 3-D human action recognition. *IEEE Access, 5*, 22590-22604.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence, 40*(4), 834-848. Retrieved from https://ieeexplore.ieee.org/document/7913730/

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). *A simple framework for contrastive learning of visual representations.* Paper presented at the International conference on machine learning.

Chen, Z., Zhang, R., Zhang, Y. E., Zhou, H., Fang, H.-S., Rock, R. R., . . . Tye, K. M. (2020). AlphaTracker: a multi-animal tracking and behavioral analysis tool. *BioRxiv*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Church, J. S., Cook, N. J., & Schaefer, A. L. (2009). *Recent Applications of Infrared Thermography for Animal Welfare and Veterinary Research: Everything from Chicks to Elephants* Paper presented at the InfraMation 2009.

Clausing, T. (2016). Thermography–Past, Present and Future. *Badania Nieniszczące i Diagnostyka*(1-2), 49--50.

CleverSys_Inc. GroupHousedScan software. http://cleversysinc.com/?csi_products=grouphousedscan.

Couprie, C., Farabet, C., Najman, L., & LeCun, Y. (2013). Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*.

Crall, J. D., Gravish, N., Mountcastle, A. M., & Combes, S. A. (2015). BEEtag: a low-cost, image-based tracking system for the study of animal behavior and locomotion. *PLoS One, 10*(9), e0136487.

Cryan, J. F., & Holmes, A. (2005). Model organisms: the ascent of mouse: advances in modelling human depression and anxiety. *Nature reviews Drug discovery, 4*(9), 775.

D'Souza, D., & Sadananda, M. (2017). Anxiety- and depressive-like profiles during early- and mid-adolescence in the female Wistar Kyoto rat. *Int J Dev Neurosci, 56*, 18-26. doi:10.1016/j.ijdevneu.2016.11.003

Dalal, N., & Triggs, B. (2005). *Histograms of oriented gradients for human detection.* Paper presented at the Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.

David, J. M., Chatziioannou, A. F., Taschereau, R., Wang, H., & Stout, D. B. (2013). The Hidden Cost of Housing Practices: Using Noninvasive Imaging to Quantify the Metabolic Demands of Chronic Cold Stress of Laboratory Mice. *Comparative Medicine, 63*(5), 386-391. Retrieved from http://www.ingentaconnect.com/content/aalas/cm/2013/00000063/00000005/art00001

Dawar, N., Ostadabbas, S., & Kehtarnavaz, N. (2018). Data augmentation in deep learning-based fusion of depth and inertial sensing for action recognition. *IEEE Sensors Letters, 3*(1), 1-4.

de Chaumont, F., Coura, R. D.-S., Serreau, P., Cressant, A., Chabout, J., Granon, S., & Olivo-Marin, J.-C. (2012). Computerized video analysis of social interactions in mice. *Nature Methods, 9*(4), 410-417.

de Chaumont, F., Ey, E., Torquet, N., Lagache, T., Dallongeville, S., Imbert, A., . . . Bourgeron, T. (2019). Real-time analysis of the behaviour of groups of mice via a depth-sensing camera and machine learning. *Nature biomedical engineering, 3*(11), 930-942.

Dell, A. I., Bender, J. A., Branson, K., Couzin, I. D., de Polavieja, G. G., Noldus, L. P., . . . Wikelski, M. (2014). Automated image-based tracking and its application in ecology. *Trends in ecology & evolution, 29*(7), 417-428.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). *Imagenet: A large-scale hierarchical image database.* Paper presented at the 2009 IEEE conference on computer vision and pattern recognition.

Dickinson, M. H., Farley, C. T., Full, R. J., Koehl, M., Kram, R., & Lehman, S. (2000). How animals move: an integrative view. *science, 288*(5463), 100-106.

Dollár, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). *Behavior recognition via sparse spatio-temporal features.* Paper presented at the Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). *Long-term recurrent convolutional networks for visual recognition and description.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., & Pal, C. (2016). The importance of skip connections in biomedical image segmentation. In *Deep Learning and Data Labeling for Medical Applications* (pp. 179-187): Springer.

Dunn, T. W., Marshall, J. D., Severson, K. S., Aldarondo, D. E., Hildebrand, D. G., Chettih, S. N., . . . Aronov, D. (2021). Geometric deep learning enables 3D kinematic profiling across species and environments. *Nature Methods, 18*(5), 564-573.

Ebbesen, C. L., & Froemke, R. C. (2020). Automatic tracking of mouse social posture dynamics by 3D videography, deep learning and GPU-accelerated robust optimization. *BioRxiv*.

Egnor, S. R., & Branson, K. (2016). Computational Analysis of Behavior. *Annual review of neuroscience, 39*, 217-236.

Eitel, A., Springenberg, J. T., Spinello, L., Riedmiller, M., & Burgard, W. (2015). *Multimodal deep learning for robust RGB-D object recognition.* Paper presented at the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Elboushaki, A., Hannane, R., Afdel, K., & Koutti, L. (2020). MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences. *Expert Systems with Applications, 139*, 112829.

Elgammal, A., Harwood, D., & Davis, L. (2000). Non-parametric model for background subtraction. *Computer Vision—ECCV 2000*, 751-767.

Fabio Luzi, Malcolm Mitchell, Leonardo Nanni Costa, & Redaelli, V. (Eds.). (2013). *Thermography - Current status and advances in livestock animals and in veterinary medicine*. Brescia: Fondazione Iniziative Zooprofilattiche e Zootechniche.

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). *Slowfast networks for video recognition.* Paper presented at the Proceedings of the IEEE/CVF international conference on computer vision.

Feichtenhofer, C., Pinz, A., & Wildes, R. P. (2017). *Spatiotemporal multiplier networks for video action recognition.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Fiaschi, L., Diego, F., Gregor, K., Schiegg, M., Koethe, U., Zlatic, M., & Hamprecht, F. A. (2014). *Tracking indistinguishable translucent objects over time using weakly supervised structured learning.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Forys, B. J., Xiao, D., Gupta, P., & Murphy, T. H. (2020). Real-time selective markerless tracking of forepaws of head fixed mice using deep neural networks. *Eneuro, 7*(3).

Francisco, F. A., Nührenberg, P., & Jordan, A. (2020). High-resolution, non-invasive animal tracking and reconstruction of local environment in aquatic ecosystems. *Movement ecology, 8*(1), 1-12.

Gabbi, A. M., Kolling, G. J., Fischer, V., Pereira, L. G. R., Tomich, T. R., Machado, F. S., . . . Santos, M. K. (2021). Use of infrared thermography to estimate enteric methane production in dairy heifers. *Quantitative InfraRed Thermography Journal*, 1-9.

Ganapathi, V., Plagemann, C., Koller, D., & Thrun, S. (2010a). *Real time motion capture using a single time-of-flight camera.* Paper presented at the Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.

Ganapathi, V., Plagemann, C., Koller, D., & Thrun, S. (2010b). *Real time motion capture using a single time-of-flight camera.* Paper presented at the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Geros, A., Horta, R., & Aguiar, P. (2016). Facegram - Objective quantitative analysis in facial reconstructive surgery. *J Biomed Inform, 61*, 1-9. doi:10.1016/j.jbi.2016.03.011

Gerós, A., Magalhães, A., & Aguiar, P. (2020). Improved 3D tracking and automated classification of rodents' behavioral activity using depth-sensing cameras. *Behavior research methods, 52*(5), 2156-2167.

Gershow, M., Berck, M., Mathew, D., Luo, L., Kane, E. A., Carlson, J. R., & Samuel, A. D. (2012). Controlling airborne cues to study small animal navigation. *Nature methods, 9*(3), 290-296.

Geuther, B. Q., Deats, S. P., Fox, K. J., Murray, S. A., Braun, R. E., White, J. K., . . . Kumar, V. (2019). Robust mouse tracking in complex environments using neural networks. *Communications biology, 2*(1), 124. doi:10.1038/s42003-019-0362-1

Giancardo, L., Sona, D., Huang, H., Sannino, S., Managò, F., Scheggia, D., . . . Murino, V. (2013). Automatic visual tracking and social behaviour analysis with multiple mice. *PLoS One, 8*(9), e74557.

Gilbert, C., McCafferty, D. J., Giroud, S., Ancel, A., & Blanc, S. (2012). Private Heat for Public Warmth: How Huddling Shapes Individual Thermogenic Responses of Rabbit Pups. *PLoS One, 7*(3), e33553. doi:10.1371/journal.pone.0033553

Girshick, R. (2015). *Fast r-cnn.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Gjendal, K., Franco, N. H., Ottesen, J. L., Sørensen, D. B., & Olsson, I. A. S. (2018). Eye, body or tail? Thermography as a measure of stress in mice. *Physiology & behavior, 196*, 135-143. doi:https://doi.org/10.1016/j.physbeh.2018.08.022

Glorot, X., Bordes, A., & Bengio, Y. (2011). *Deep sparse rectifier neural networks.* Paper presented at the Proceedings of the fourteenth international conference on artificial intelligence and statistics.

Gomez-Marin, A., Partoune, N., Stephens, G. J., & Louis, M. (2012). Automated tracking of animal posture and movement during exploration and sensory orientation behaviors. *PloS one, 7*(8), e41642.

Gomez-Marin, A., Paton, J. J., Kampff, A. R., Costa, R. M., & Mainen, Z. F. (2014). Big behavioral data: psychology, ethology and the foundations of neuroscience. *Nat Neurosci, 17*(11), 1455-1462. doi:10.1038/nn.3812

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*: MIT press.

Gordon, C. (2012). Thermal physiology of laboratory mice: Defining thermoneutrality. *Journal of Thermal Biology, 37*(8), 654-685.

Gordon, C. J., Spencer, P. J., Hotchkiss, J., Miller, D. B., Hinderliter, P. M., & Pauluhn, J. (2008). Thermoregulation and its influence on toxicity assessment. *Toxicology, 244*(2), 87-97.

Gouveia, K., & Hurst, J. L. (2013). Reducing mouse anxiety during handling: Effect of experience with handling tunnels. *PLoS One, 8*(6), e66401.

Graving, J. M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B. R., & Couzin, I. D. (2019). DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife, 8*, e47994.

Gribovskiy, A., Halloy, J., Deneubourg, J.-L., Bleuler, H., & Mondada, F. (2010). *Towards mixed societies of chickens and robots.* Paper presented at the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Günel, S., Rhodin, H., Morales, D., Campagnolo, J., Ramdya, P., & Fua, P. (2019). DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila. *Elife, 8*, e48571.

Gupta, S., Girshick, R., Arbeláez, P., & Malik, J. (2014). *Learning rich features from RGB-D images for object detection and segmentation.* Paper presented at the European conference on computer vision.

Gutschoven, B., & Verlinde, P. (2000). *Multi-modal identity verification using support vector machines (SVM).* Paper presented at the Proceedings of the Third International Conference on Information Fusion.

Halloy, J., Sempo, G., Caprari, G., Rivault, C., Asadpour, M., Tâche, F., . . . Amé, J. M. (2007). Social integration of robots into groups of cockroaches to control self-organized choices. *science, 318*(5853), 1155-1158.

Hansard, M., Lee, S., Choi, O., & Horaud, R. P. (2012). *Time-of-flight cameras: principles, methods and applications*: Springer Science & Business Media.

Hara, K., Kataoka, H., & Satoh, Y. (2018). *Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?* Paper presented at the Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Harshaw, C., & Alberts, J. R. (2012). Group and individual regulation of physiology and behavior: A behavioral, thermographic, and acoustic study of mouse development. *Physiology & behavior, 106*(5), 670-682. doi:http://dx.doi.org/10.1016/j.physbeh.2012.05.002

Hartinger, J., Külbs, D., Volkers, P., & Cussler, K. (2002). Suitability of temperature-sensitive transponders to measure body temperature during animal experiments required for regulatory tests. *Altex, 20*(2), 65-70.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *Mask r-cnn.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Helwig, B. G., Ward, J. A., Blaha, M. D., & Leon, L. R. (2012). Effect of intraperitoneal radiotelemetry instrumentation on voluntary wheel running and surgical recovery in mice. *Journal of the American Association for Laboratory Animal Science: JAALAS, 51*(5), 600.

Herborn, K. A., Graves, J. L., Jerem, P., Evans, N. P., Nager, R., McCafferty, D. J., & McKeegan, D. E. F. (2015). Skin temperature reveals the intensity of acute stress. *Physiology & behavior, 152*, 225-230. doi:https://doi.org/10.1016/j.physbeh.2015.09.032

Herborn, K. A., Jerem, P., Nager, R. G., McKeegan, D. E. F., & McCafferty, D. J. (2018). Surface temperature elevated by chronic and intermittent stress. *Physiology & behavior, 191*, 47-55. doi:https://doi.org/10.1016/j.physbeh.2018.04.004

Herbut, E., & Walczak, J. (2013). Infrared thermography as a method for evaluating the welfare of animals subjected to invasive procedures-a review. *Annals of Animal Science, 13*(3), 423-434.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780. Retrieved from https://www.mitpressjournals.org/doi/pdfplus/10.1162/neco.1997.9.8.1735

Hong, W., Kennedy, A., Burgos-Artizzu, X. P., Zelikowsky, M., Navonne, S. G., Perona, P., & Anderson, D. J. (2015). Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. *Proceedings of the National Academy of Sciences, 112*(38), E5351-E5360.

Hsu, A. I., & Yttri, E. A. (2021). An Open Source Unsupervised Algorithm for Identification and Fast Prediction of Behaviors. *BioRxiv*, 770271.

Intel®RealSenseTM. (2020). Product Family D400 Series - Datasheet (Document Number: 337029-009).

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). *The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition workshops.

Jhuang, H., Garrote, E., Yu, X., Khilnani, V., Poggio, T., Steele, A. D., & Serre, T. (2010). Automated home-cage behavioural phenotyping of mice. *Nature communications, 1*, 68.

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence, 35*(1), 221-231.

Jiang, Z., Chazot, P. L., Celebi, M. E., Crookes, D., & Jiang, R. (2019). Social behavioral phenotyping of Drosophila with a 2D–3D hybrid CNN framework. *IEEE Access, 7*, 67972-67982.

Jin, T. L., & Duan, F. (2019). Rat Behavior Observation System Based on Transfer Learning. *IEEE Access, 7*, 62152-62162. doi:10.1109/Access.2019.2916339

Joy, A., Taheri, S., Dunshea, F., Leury, B., DiGiacomo, K., Osei-Amponsah, R., . . . Chauhan, S. (2021). Non-invasive measure of heat stress in sheep using machine learning techniques and infrared thermography. *Small Ruminant Research*, 106592.

Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S., & Branson, K. (2013). JAABA: interactive machine learning for automatic annotation of animal behavior. *Nature Methods, 10*(1), 64-67.

Kane, G. A., Lopes, G., Saunders, J. L., Mathis, A., & Mathis, M. W. (2020). Real-time, low-latency closed-loop feedback using markerless posture tracking. *Elife, 9*, e61909.

Karashchuk, P., Rupp, K. L., Dickinson, E. S., Walling-Bell, S., Sanders, E., Azim, E., . . . Tuthill, J. C. (2021). Anipose: a toolkit for robust markerless 3D pose estimation. *Cell reports, 36*(13), 109730.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). *Large-scale video classification with convolutional neural networks.* Paper presented at the Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Khoshelham, K., & Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors, 12*(2), 1437-1454.

Kim, C., Ruberto, T., Phamduy, P., & Porfiri, M. (2018). Closed-loop control of zebrafish behaviour in three dimensions using a robotic stimulus. *Scientific reports, 8*(1), 1-15.

Kim, C., Yun, S., Jung, S.-W., & Won, C. S. (2015). Color and depth image correspondence for Kinect v2. In *Advanced Multimedia and Ubiquitous Engineering* (pp. 111-116): Springer.

Kim, S., & Hidaka, Y. (2021). Breathing Pattern Analysis in Cattle Using Infrared Thermography and Computer Vision. *Animals, 11*(1), 207.

Klibaite, U., Berman, G. J., Cande, J., Stern, D. L., & Shaevitz, J. W. (2017). An unsupervised method for quantifying the behavior of paired animals. *Phys Biol, 14*(1), 015006. doi:10.1088/1478-3975/aa5c50

Koller, O., Ney, H., & Bowden, R. (2016). *Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Kort, W., Hekking-Weijma, J., Tenkate, M., Sorm, V., & VanStrik, R. (1998). A microchip implant system as a method to determine body temperature of terminally ill rats and mice. *Laboratory Animals, 32*(3), 260-269. doi:10.1258/002367798780559329

Krakauer, J. W., Ghazanfar, A. A., Gomez-Marin, A., MacIver, M. A., & Poeppel, D. (2017). Neuroscience Needs Behavior: Correcting a Reductionist Bias. *Neuron, 93*(3), 480-490. doi:10.1016/j.neuron.2016.12.041

Kramida, G., Aloimonos, Y., Parameshwara, C. M., Fermüller, C., Francis, N. A., & Kanold, P. (2016). *Automated mouse behavior recognition using VGG features and LSTM networks.* Paper presented at the Proc. Vis. Observ. Anal. Vertebrate Insect Behav. Workshop (VAIB).

Kriegeskorte, N., Simmons, W. K., Bellgowan, P. S. F., & Baker, C. I. (2009). Circular analysis in systems neuroscience: the dangers of double dipping. *Nature Neuroscience, 12*, 535. doi:10.1038/nn.2303
https://www.nature.com/articles/nn.2303#supplementary-information

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks.* Paper presented at the Advances in neural information processing systems.

Kulikov, V. A., Khotskin, N. V., Nikitin, S. V., Lankin, V. S., Kulikov, A. V., & Trapezov, O. V. (2014). Application of 3-D imaging sensor for tracking minipigs in the open field test. *J Neurosci Methods, 235*, 219-225. doi:10.1016/j.jneumeth.2014.07.012

Kumar, A., Shrivatsav, S. N., Subrahmanyam, G. R. S., & Mishra, D. (2016). *Application of transfer learning in RGB-D object recognition.* Paper presented at the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

Lachat, E., Macher, H., Landes, T., & Grussenmeyer, P. (2015). Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling. *Remote Sensing, 7*(10), 13070-13097. doi:10.3390/rs71013070

Lachat, E., Macher, H., Mittet, M., Landes, T., & Grussenmeyer, P. (2015). First experiences with Kinect v2 sensor for close range 3D modelling. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40*(5), 93.

Langen, B., & Dost, R. (2011). Comparison of SHR, WKY and Wistar rats in different behavioural animal models: effect of dopamine D1 and alpha2 agonists. *Atten Defic Hyperact Disord, 3*(1), 1-12. doi:10.1007/s12402-010-0034-y

Lauer, J., Zhou, M., Ye, S., Menegas, W., Schneider, S., Nath, T., . . . Feng, G. (2022). Multi-animal pose estimation, identification and tracking with DeepLabCut. Nature Methods, 1-9.

Lecorps, B., Rödel, H. G., & Féron, C. (2016). Assessment of anxiety in open field and elevated plus maze using infrared thermography. *Physiology & behavior, 157*, 209-216. doi:http://doi.org/10.1016/j.physbeh.2016.02.014

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444. doi:10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). *Convolutional networks and applications in vision.* Paper presented at the Proceedings of 2010 IEEE international symposium on circuits and systems.

Lenz, P., Geiger, A., & Urtasun, R. (2015). *Followme: Efficient online min-cost flow tracking with bounded memory and computation.* Paper presented at the Proceedings of the IEEE International Conference on Computer Vision.

Li, S., Gunel, S., Ostrek, M., Ramdya, P., Fua, P., & Rhodin, H. (2020). *Deformation-aware unpaired image translation for pose estimation on laboratory animals.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Liang, X., Shen, X., Feng, J., Lin, L., & Yan, S. (2016). *Semantic object parsing with graph lstm.* Paper presented at the European Conference on Computer Vision.

Litomisky, K. (2012). Consumer rgb-d cameras and their applications. *Rapport technique, University of California, 20*.

Liu, K., Liu, W., Gan, C., Tan, M., & Ma, H. (2018). *T-C3D: Temporal convolutional 3D network for real-time action recognition.* Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence.

Liu, X., Yu, S.-y., Flierman, N., Loyola, S., Kamermans, M., Hoogland, T. M., & De Zeeuw, C. I. (2020). OptiFlex: video-based animal pose estimation using deep learning enhanced by optical flow. *BioRxiv*.

Liu, Z., Zhang, C., & Tian, Y. (2016). 3D-based deep convolutional neural network for action recognition with depth sequences. *Image and Vision Computing, 55*, 93-100.

Lloyd, J. M. (2013). *Thermal imaging systems*: Springer Science & Business Media.

Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully convolutional networks for semantic segmentation.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Lopes, G., Bonacchi, N., Frazão, J., Neto, J. P., Atallah, B. V., Soares, S., . . . Correia, P. A. (2015). Bonsai: an event-based framework for processing and controlling data streams. *Frontiers in neuroinformatics, 9*, 7.

Lorbach, M., Kyriakou, E. I., Poppe, R., van Dam, E. A., Noldus, L., & Veltkamp, R. C. (2018). Learning to recognize rat social behavior: Novel dataset and cross-dataset application. *J Neurosci Methods, 300*, 166-172. doi:10.1016/j.jneumeth.2017.05.006

Lorbach, M., Poppe, R., & Veltkamp, R. C. (2019). Interactive rodent behavior annotation in video using active learning. *Multimedia Tools and Applications, 78*(14), 19787-19806.

Ludwig, N., Gargano, M., Luzi, F., Carenzi, C., & Verga, M. (2010). Technical note: Applicability of infrared thermography as a non invasive measurements of stress in rabbit. *World Rabbit Science, 15*(4), p. 199-206.

Machado, A. S., Darmohray, D. M., Fayad, J., Marques, H. G., & Carey, M. R. (2015). A quantitative framework for whole-body coordination reveals specific deficits in freely walking ataxic mice. *Elife, 4*. doi:10.7554/eLife.07892

Macpherson, T., Churchland, A., Sejnowski, T., DiCarlo, J., Kamitani, Y., Takahashi, H., & Hikida, T. (2021). Natural and Artificial Intelligence: A brief introduction to the interplay between AI and neuroscience research. *Neural Networks, 144*, 603-613.

Macrì, S., Mainetti, L., Patrono, L., Pieretti, S., Secco, A., & Sergi, I. (2015). *A tracking system for laboratory mice to support medical researchers in behavioral analysis.* Paper presented at the Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE.

Madai-Tahy, L., Otte, S., Hanten, R., & Zell, A. (2016). *Revisiting deep convolutional neural networks for RGB-D based object recognition.* Paper presented at the International Conference on Artificial Neural Networks.

Mahadevan, S., Athar, A., Ošep, A., Hennen, S., Leal-Taixé, L., & Leibe, B. (2020). Making a Case for 3D Convolutions for Object Segmentation in Videos. *arXiv preprint arXiv:2008.11516*.

Maillot, O., Leduc, N., Atallah, V., Escarmant, P., Petit, A., Belhomme, S., . . . Vinh-Hung, V. (2018). Evaluation of acute skin toxicity of breast radiotherapy using thermography: Results of a prospective single-centre trial. *Cancer/Radiothérapie*. doi:https://doi.org/10.1016/j.canrad.2017.10.007

Majd, M., & Safabakhsh, R. (2020). Correlational convolutional LSTM for human action recognition. *Neurocomputing, 396*, 224-229.

Manzano-Szalai, K., Pali-Schöll, I., Krishnamurthy, D., Stremnitzer, C., Flaschberger, I., & Jensen-Jarolim, E. (2016). Anaphylaxis imaging: non-invasive measurement of surface body temperature and physical activity in small animals. *PLoS One, 11*(3), e0150819.

Marks, A., Vianna, D. M. L., & Carrive, P. (2009). Nonshivering thermogenesis without interscapular brown adipose tissue involvement during conditioned fear in the rat. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology, 296*(4), R1239-R1247. doi:10.1152/ajpregu.90723.2008

Marks, M., Qiuhan, J., Sturman, O., von Ziegler, L., Kollmorgen, S., von der Behrens, W., . . . Yanik, M. F. (2020). SIPEC: the deep-learning Swiss knife for behavioral data analysis. *BioRxiv*.

Marques, J. C., Lackner, S., Felix, R., & Orger, M. B. (2018). Structure of the Zebrafish Locomotor Repertoire Revealed with Unsupervised Behavioral Clustering. *Curr Biol, 28*(2), 181-195 e185. doi:10.1016/j.cub.2017.12.002

Martinez, M. D., Ghamari-Langroudi, M., Gifford, A., Cone, R., & Welch, E. B. (2015). *Automated pipeline to analyze non-contact infrared images of the paraventricular nucleus specific leptin receptor knock-out mouse model.* Paper presented at the SPIE Medical Imaging.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat Neurosci, 21*(9), 1281-1289. doi:10.1038/s41593-018-0209-y

Mathis, M. W., & Mathis, A. (2020). Deep learning tools for the measurement of animal behavior in neuroscience. *Current opinion in neurobiology, 60*, 1-11.

Matsumoto, J., Urakawa, S., Takamura, Y., Malcher-Lopes, R., Hori, E., Tomaz, C., . . . Nishijo, H. (2013). A 3D-video-based computerized analysis of social and sexual interactions in rats. *PloS one, 8*(10), e78460.

Mazur-Milecka, M. (2016). *Thermal imaging in automatic rodent's social behaviour analysis*. Paper presented at the 13th Quantitative InfraRed Thermography Conference, Gdańsk, Poland. 10.21611/qirt.2016.083

Mazur-Milecka, M., & Ruminski, J. (2020). Deep learning based thermal image segmentation for laboratory animals tracking. *Quantitative InfraRed Thermography Journal*, 1-18.

Mazur-Milecka, M., & Rumiński, J. (2017). *Automatic analysis of the aggressive behavior of laboratory animals using thermal video processing.* Paper presented at the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).

McCafferty, D. J., Gallon, S., & Nord, A. (2015). Challenges of measuring body temperatures of free-ranging birds and mammals. *Animal Biotelemetry, 3*(1), 33.

Mench, J. (1998). Why It Is Important to Understand Animal Behavior. *ILAR journal, 39*(1), 20-26. doi:10.1093/ilar.39.1.20

Meyer, C. W., Ootsuka, Y., & Romanovsky, A. A. (2017). Body Temperature Measurements for Metabolic Phenotyping in Mice. *Frontiers in Physiology, 8*(520). doi:10.3389/fphys.2017.00520

Mikhailov, A. (2019). Turbo Colormap Look-up Table. GitHub repository: https://gist.github.com/mikhailov-work/6a308c20e494d9e0ccc29036b28faa7a.

Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., & Kautz, J. (2016). *Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Monteiro, J. P., Oliveira, H. P., Aguiar, P., & Cardoso, J. S. (2012). *Depth-map images for automatic mice behavior recognition.* Paper presented at the 1st PhD Students Conference in Electrical and Computer Engineering, Porto, Portugal.

Morton, D. B., Hawkins, P., Bevan, R., Heath, K., Kirkwood, J., Pearce, P., . . . Webb, A. (2003). Refinements in telemetry procedures. *Laboratory Animals, 37*(4), 261-300.

Mota-Rojas, D., Olmos-Hernández, A., Verduzco-Mendoza, A., Lecona-Butrón, H., Martínez-Burnes, J., Mora-Medina, P., . . . Orihuela, A. (2021). Infrared thermal imaging associated with pain in laboratory animals. *Experimental Animals, 70*(1), 1-12.

Mufford, J. T., Paetkau, M. J., Flood, N. J., Regev-Shoshani, G., Miller, C. C., & Church, J. S. (2016). The development of a non-invasive behavioral model of thermal heat stress in laboratory mice (Mus musculus). *Journal of neuroscience methods, 268*, 189-195. doi:https://doi.org/10.1016/j.jneumeth.2015.12.011

Muhammad, G., Alshehri, F., Karray, F., El Saddik, A., Alsulaiman, M., & Falk, T. H. (2021). A comprehensive survey on multimodal medical signals fusion for smart healthcare systems. *Information Fusion*.

Murari, K. (2019). *Recurrent 3D Convolutional Network for Rodent Behavior Recognition.* Paper presented at the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Nakamura, T., Matsumoto, J., Nishimaru, H., Bretas, R. V., Takamura, Y., Hori, E., . . . Nishijo, H. (2016). A markerless 3D computerized motion capture system incorporating a skeleton model for monkeys. *PLoS One, 11*(11), e0166154.

Nestler, E. J., & Hyman, S. E. (2010). Animal models of neuropsychiatric disorders. *Nature neuroscience, 13*(10), 1161-1169.

Nguyen, N. G., Phan, D., Lumbanraja, F. R., Faisal, M. R., Abapihi, B., Purnama, B., . . . Satou, K. (2019). Applying deep learning models to mouse behavior recognition. *Journal of Biomedical Science and Engineering, 12*(2), 183-196.

Nilsson, S. R., Goodwin, N. L., Choong, J. J., Hwang, S., Wright, H. R., Norville, Z., . . . Eshel, N. (2020). Simple Behavioral Analysis (SimBA): an open source toolkit for computer classification of complex social behaviors in experimental animals. *BioRxiv*.

Noldus_Information_Technology_BV. Ethovision XT software. http://www.noldus.com/ethovision.

Nourizonoz, A., Zimmermann, R., Ho, C. L. A., Pellat, S., Ormen, Y., Prévost-Solié, C., . . . Herrel, A. (2020). EthoLoop: automated closed-loop neuroethology in naturalistic environments. *Nature Methods, 17*(10), 1052-1059.

Ohayon, S., Avni, O., Taylor, A. L., Perona, P., & Egnor, S. R. (2013). Automated multi-day tracking of marked mice for the analysis of social behaviour. *Journal of neuroscience methods, 219*(1), 10-19.

OpenCV. Introduction to support vector machines. Retrieved from http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

Ostrower, D. (2006). Optical Thermal Imaging–replacing microbolometer technology and achieving universal deployment. *III-Vs Review, 19*(6), 24-27.

Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1), 62-66.

Ou-Yang, T.-H., Tsai, M.-L., Yen, C.-T., & Lin, T.-T. (2011). An infrared range camera-based approach for three-dimensional locomotion tracking and pose reconstruction in a rodent. *Journal of neuroscience methods, 201*(1), 116-123.

Overstreet, D. H. (2012). Modeling depression in animal models. In *Psychiatric Disorders* (pp. 125-144): Springer.

Overton, J. M. (2010). Phenotyping small animals as models for the human metabolic syndrome: thermoneutrality matters. *Int J Obes (Lond), 34 Suppl 2*, S53-58. doi:10.1038/ijo.2010.240

Parker, R. M. A., & Browne, W. J. (2014). The Place of Experimental Design and Statistics in the 3Rs. *ILAR journal, 55*(3), 477-485. doi:10.1093/ilar/ilu044

Paulino Fernandez, O., van Dam, E. A., Noldus, L., & Veltkamp, R. (2014). *Robust Point Cloud Segmentation of Rodents using Close Range Depth Cameras in Controlled Environments.* Paper presented at the proceedings ICPR workshop on Visual observation and analysis of Vertebrate And Insect Behavior.

Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S., Murthy, M., & Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks. *Nat Methods, 16*(1), 117-125. doi:10.1038/s41592-018-0234-5

Pereira, T. D., Shaevitz, J. W., & Murthy, M. (2020). Quantifying behavior to understand the brain. *Nature neuroscience, 23*(12), 1537-1549.

Pereira, T. D., Tabris, N., Li, J., Ravindranath, S., Papadoyannis, E. S., Wang, Z. Y., . . . Falkner, A. L. (2020). SLEAP: Multi-animal pose tracking. *BioRxiv*.

Pérez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S., & De Polavieja, G. G. (2014). idTracker: tracking individuals in a group by automatic identification of unmarked animals. *Nature Methods, 11*(7), 743-748.

Pfeuffer, A., Schulz, K., & Dietmayer, K. (2019). *Semantic segmentation of video sequences with convolutional lstms.* Paper presented at the 2019 IEEE Intelligent Vehicles Symposium (IV).

Piccione, G., Gianesella, M., Morgante, M., & Refinetti, R. (2013). Daily rhythmicity of core and surface temperatures of sheep kept under thermoneutrality or in the cold. *Research in Veterinary Science, 95*(1), 261-265. doi:https://doi.org/10.1016/j.rvsc.2013.03.005

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In *Machine learning* (pp. 101-121): Elsevier.

Powers, D. R., Langland, K. M., Wethington, S. M., Powers, S. D., Graham, C. H., & Tobalske, B. W. (2017). Hovering in the heat: effects of environmental temperature on heat regulation in foraging hummingbirds. *Royal Society Open Science, 4*(12). Retrieved from http://rsos.royalsocietypublishing.org/content/4/12/171056.abstract

Preisig, D. F., Kulic, L., Kruger, M., Wirth, F., McAfoose, J., Spani, C., . . . Welt, T. (2016). High-speed video gait analysis reveals early and characteristic locomotor phenotypes in mouse models of neurodegenerative movement disorders. *Behav Brain Res, 311*, 340-353. doi:10.1016/j.bbr.2016.04.044

Priego Quesada, J. I., Kunzler, M. R., & Carpes, F. P. (2017). Methodological Aspects of Infrared Thermography in Human Assessment. In J. I. Priego Quesada (Ed.), *Application of Infrared Thermography in Sports Science* (pp. 49-79). Cham: Springer International Publishing.

Qiu, Z., Yao, T., & Mei, T. (2017). Learning deep spatio-temporal dependence for semantic video segmentation. *IEEE Transactions on Multimedia, 20*(4), 939-949.

Rahman, M. M., Tan, Y., Xue, J., Shao, L., & Lu, K. (2019). 3D object detection: Learning 3D bounding boxes from scaled down 2D bounding boxes in RGB-D images. *Information Sciences, 476*, 147-158.

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., & Bengio, Y. (2020). *Multi-task self-supervised learning for robust speech recognition.* Paper presented at the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Ravbar, P., Branson, K., & Simpson, J. H. (2019). An automatic behavior recognition system classifies animal behaviors using movements and their temporal context. *Journal of neuroscience methods, 326*, 108352.

Reardon, S. (2016). A mouse's house may ruin experiments. *Nature News, 530*(7590), 264.

Rebol, M., & Knöbelreiter, P. (2020). Frame-To-Frame consistent semantic segmentation. *arXiv preprint arXiv:2008.00948*.

Redfern, W. S., Tse, K., Grant, C., Keerie, A., Simpson, D. J., Pedersen, J. C., . . . Armstrong, J. D. (2017). Automated recording of home cage activity and temperature of individual rats housed in social groups: The Rodent Big Brother project. *PLoS One, 12*(9), e0181068. doi:10.1371/journal.pone.0181068

Ren, L., Lu, J., Feng, J., & Zhou, J. (2017). Multi-modal uniform deep learning for RGB-D person re-identification. *Pattern Recognition, 72*, 446-457.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster r-cnn: Towards real-time object detection with region proposal networks.* Paper presented at the Advances in neural information processing systems.

Richardson, C. A. (2015). The power of automated behavioural homecage technologies in characterizing disease progression in laboratory mice: A review. *Applied Animal Behaviour Science, 163*, 19-27. doi:10.1016/j.applanim.2014.11.018

Robie, A. A., Seagraves, K. M., Egnor, S. R., & Branson, K. (2017). Machine vision methods for analyzing social interactions. *Journal of Experimental Biology, 220*(1), 25-34.

Rodriguez, A., Zhang, H. Q., Klaminder, J., Brodin, T., Andersson, P. L., & Andersson, M. (2018). ToxTrac: A fast and robust software for tracking organisms. *Methods in Ecology and Evolution, 9*(3), 460-464. doi:10.1111/2041-210x.12874

Romero-Ferrero, F., Bergomi, M. G., Hinz, R. C., Heras, F. J. H., & de Polavieja, G. G. (2019). idtracker.ai: tracking all individuals in small or large collectives of unmarked animals. *Nat Methods, 16*(2), 179-182. doi:10.1038/s41592-018-0295-5

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-net: Convolutional networks for biomedical image segmentation.* Paper presented at the International Conference on Medical image computing and computer-assisted intervention.

Rousseau, J. B., Van Lochem, P. B., Gispen, W., & Spruijt, B. (2000). Classification of rat behavior with an image-processing method and a neural network. *Behavior research methods, 32*(1), 63-71.

Saberioon, M. M., & Cisar, P. (2016). Automated multiple fish tracking in three-Dimension using a Structured Light Sensor. *Computers and Electronics in Agriculture, 121*, 215-221. doi:10.1016/j.compag.2015.12.014

Saito, H., Sherwood, E. R., Varma, T. K., & Evers, B. M. (2003). Effects of aging on mortality, hypothermia, and cytokine induction in mice with endotoxemia or sepsis. *Mechanisms of Ageing and Development, 124*(10), 1047-1058. doi:https://doi.org/10.1016/j.mad.2003.08.002

Salvador, A., Bellver, M., Campos, V., Baradad, M., Marques, F., Torres, J., & Giro-i-Nieto, X. (2017). Recurrent neural networks for semantic instance segmentation. *arXiv preprint arXiv:1712.00617*.

Sanford, L. D., Yang, L., & Wellman, L. L. (2011). Telemetry in Mice: Applications in Studies of Stress and Anxiety Disorders. In *Mood and Anxiety Related Phenotypes in Mice* (pp. 43-60): Springer.

Schwarz, R. F., Branicky, R., Grundy, L. J., Schafer, W. R., & Brown, A. E. (2015). Changes in Postural Syntax Characterize Sensory Modulation and Natural Variation of C. elegans Locomotion. *PLoS Comput Biol, 11*(8), e1004322. doi:10.1371/journal.pcbi.1004322

Schweihoff, J. F., Loshakov, M., Pavlova, I., Kück, L., Ewell, L. A., & Schwarz, M. K. (2021). DeepLabStream enables closed-loop behavioral experiments using deep learning-based markerless, real-time posture detection. *Communications biology, 4*(1), 1-11.

Segalin, C., Williams, J., Karigo, T., Hui, M., Zelikowsky, M., Sun, J. J., . . . Kennedy, A. (2020). The Mouse Action Recognition System (MARS): a software pipeline for automated analysis of social behaviors in mice. *BioRxiv*.

Sehara, K., Zimmer-Harwood, P., Larkum, M. E., & Sachdev, R. N. (2021). Real-time closed-loop feedback in behavioral time scales using DeepLabCut. *Eneuro, 8*(2).

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Shah, S. A. A., Bennamoun, M., Boussaid, F., & While, L. (2017). Evolutionary feature learning for 3-D object recognition. *IEEE Access, 6*, 2434-2444.

Shi, Q., Ishii, H., Kinoshita, S., Takanishi, A., Okabayashi, S., Iida, N., . . . Shibata, S. (2013). Modulation of rat behaviour by using a rat-like robot. *Bioinspiration & biomimetics, 8*(4), 046002.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(1), 60.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*.

Singh, R., Khurana, R., Kushwaha, A. K. S., & Srivastava, R. (2020). Combining CNN streams of dynamic image and depth data for action recognition. *Multimedia Systems*, 1-10.

Snowdon, C. P. o. t. A. B. S. (2017). Significance of Animal Behavior Research.

Sobral, A., & Vacavant, A. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding, 122*, 4-21.

Socher, R., Huval, B., Bath, B., Manning, C. D., & Ng, A. Y. (2012). *Convolutional-recursive deep learning for 3d object classification.* Paper presented at the Advances in neural information processing systems.

Song, X., Herranz, L., & Jiang, S. (2017). *Depth cnns for rgb-d scene recognition: Learning from scratch better than transferring from rgb-cnns.* Paper presented at the Thirty-First AAAI Conference on Artificial Intelligence.

Spink, A., Tegelenbosch, R., Buma, M., & Noldus, L. (2001). The EthoVision video tracking system—a tool for behavioral phenotyping of transgenic mice. *Physiology & behavior, 73*(5), 731-744.

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

Spruijt, B. M., & DeVisser, L. (2006). Advanced behavioural screening: automated home cage ethology. *Drug Discov Today Technol, 3*(2), 231-237. doi:10.1016/j.ddtec.2006.06.010

Sridhar, V. H., Roche, D. G., & Gingins, S. (2019). Tracktor: image-based automated tracking of animal movement and behaviour. *Methods in Ecology and Evolution, 10*(6), 815-820.

Stauffer, C., & Grimson, W. E. L. (1999). *Adaptive background mixture models for real-time tracking.* Paper presented at the Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149).

Stavrakakis, S., Li, W., Guy, J. H., Morgan, G., Ushaw, G., Johnson, G. R., & Edwards, S. A. (2015). Validity of the Microsoft Kinect sensor for assessment of normal walking patterns in pigs. *Computers and Electronics in Agriculture, 117*, 1-7. doi:10.1016/j.compag.2015.07.003

Stern, U., He, R., & Yang, C.-H. (2015). Analyzing animal behavior via classifying each video frame using convolutional neural networks. *Scientific reports, 5*.

Stonehouse, B. (1978). *Animal marking: recognition marking of animals in research*: Macmillan London.

Straw, A. D., Branson, K., Neumann, T. R., & Dickinson, M. H. (2010). Multi-camera real-time three-dimensional tracking of multiple flying animals. *Journal of The Royal Society Interface*, rsif20100230.

Sturman, O., von Ziegler, L., Schläppi, C., Akyol, F., Privitera, M., Slominski, D., . . . Grewe, B. (2020). Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. *Neuropsychopharmacology, 45*(11), 1942-1952. Retrieved from https://www.nature.com/articles/s41386-020-0776-y.pdf

Sutherland, M. A., Worth, G. M., Dowling, S. K., Lowe, G. L., Cave, V. M., & Stewart, M. (2020). Evaluation of infrared thermography as a non-invasive method of measuring the autonomic nervous response in sheep. *PLoS One, 15*(5), e0233558.

Suwajanakorn, S., Snavely, N., Tompson, J., & Norouzi, M. (2018). Discovery of latent 3d keypoints via end-to-end geometric reasoning. *arXiv preprint arXiv:1807.03146*.

Tang, X., & D. Sanford, L. (2002). Telemetric recording of sleep and home cage activity in mice. *Sleep, 25*(6), 677-685.

Tang, X., & Sanford, L. D. (2005). Home cage activity and activity-based measures of anxiety in 129P3/J, 129X1/SvJ and C57BL/6J mice. *Physiology & behavior, 84*(1), 105-115.

Tanha, J., Van Someren, M., de Bakker, M., Bouteny, W., Shamoun-Baranesy, J., & Afsarmanesh, H. (2012). *Multiclass semi-supervised learning for animal behavior recognition from accelerometer data.* Paper presented at the 2012 IEEE 24th International Conference on Tools with Artificial Intelligence.

Tatem, K. S., Quinn, J. L., Phadke, A., Yu, Q., Gordish-Dressman, H., & Nagaraju, K. (2014). Behavioral and locomotor measurements using an open field activity monitoring system for skeletal muscle diseases. *JoVE (Journal of Visualized Experiments)*(91), e51785-e51785.

Tattersall, G. J. (2016). Infrared thermography: A non-invasive window into thermal physiology. *Comparative Biochemistry and Physiology Part A: Molecular & Integrative Physiology, 202*, 78-98. doi:https://doi.org/10.1016/j.cbpa.2016.02.022

Thanos, P. K., Restif, C., O'Rourke, J. R., Lam, C. Y., & Metaxas, D. (2017). Mouse Social Interaction Test (MoST): a quantitative computer automated analysis of behavior. *J Neural Transm (Vienna), 124*(1), 3-11. doi:10.1007/s00702-015-1487-0

Theodoridis, S., & Koutroumbas, K. (2003). *Pattern Recognition*: Elsevier, Academic Press.

Thermos, S., Papadopoulos, G. T., Daras, P., & Potamianos, G. (2020). Deep sensorimotor learning for RGB-D object recognition. *Computer Vision and Image Understanding, 190*, 102844.

Thevenot, J., López, M. B., & Hadid, A. (2017). A survey on computer vision for assistive medical diagnosis from faces. *IEEE journal of biomedical and health informatics, 22*(5), 1497-1511.

Tong, M., Yu, X., Shao, J., Shao, Z., Li, W., & Lin, W. (2020). Automated measuring method based on Machine learning for optomotor response in mice. *Neurocomputing, 418*, 241-250.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). *Learning spatiotemporal features with 3d convolutional networks.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Tresch, C. (2016). *CNN in RGB-D Image Segmentation: Preprocessing, Training, Filtering and Visualization.* (Bachelor). University of Zurich,

TSEsystems. PhenoTracker software. http://www.tse0systems.com/products/behavior/video0tracking0software/phenotracker/index.htm.

Twining, C., Taylor, C., & Courtney, P. (2001). Robust tracking and posture description for laboratory rodents using active shape models. *Behavior Research Methods, Instruments, & Computers, 33*(3), 381-391.

Uhlmann, V., Ramdya, P., Delgado-Gonzalo, R., Benton, R., & Unser, M. (2017). FlyLimbTracker: An active contour based approach for leg segment tracking in unmarked, freely behaving Drosophila. *PLoS One, 12*(4), e0173433.

Unger, J., Mansour, M., Kopaczka, M., Gronloh, N., Spehr, M., & Merhof, D. (2017). An unsupervised learning approach for tracking mice in an enclosed area. *BMC Bioinformatics, 18*(1), 272. doi:10.1186/s12859-017-1681-1

Vadlejcha, J., Kníˇzkovác, I., Makovcováa, K. r., Kuncc, P., Jankovskáa, I., Jandab, K., . . . Langrováa, I. (2010). Thermal Profile of Rabbits Infected with Eimeria intestinalis. *Veterinary Parasitology, 171* 343–345.

Valgma, L. (2016). 3D reconstruction using Kinect v2 camera. *University of Tartu*.

Valletta, J. J., Torney, C., Kings, M., Thornton, A., & Madden, J. (2017). Applications of machine learning in animal behaviour studies. *Animal Behaviour, 124*, 203-220. doi:10.1016/j.anbehav.2016.12.005

van Dam, E. A., Noldus, L. P., & van Gerven, M. A. (2020). Deep learning improves automated rodent behavior recognition within a specific experimental setup. *Journal of neuroscience methods, 332*, 108536.

van Dam, E. A., van der Harst, J. E., ter Braak, C. J., Tegelenbosch, R. A., Spruijt, B. M., & Noldus, L. P. (2013). An automated system for the recognition of various specific rat behaviours. *Journal of neuroscience methods, 218*(2), 214-224.

van den Boom, B. J. G., Pavlidi, P., Wolf, C. J. H., Mooij, A. H., & Willuhn, I. (2017). Automated classification of self-grooming in mice using open-source software. *J Neurosci Methods, 289*, 48-56. doi:10.1016/j.jneumeth.2017.05.026

Vannetti, F., Matteoli, S., Finocchio, L., Lacarbonara, F., Sodi, A., Menchini, U., & Corvi, A. (2014). Relationship between ocular surface temperature and peripheral vasoconstriction in healthy subjects: A thermographic study. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, 228*(3), 297-302. doi:10.1177/0954411914523755

Vapnik, V. (1999). *The nature of statistical learning theory*: Springer science & business media.

Varol, G., Laptev, I., & Schmid, C. (2017). Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence, 40*(6), 1510-1517.

Veeraraghavan, A., Srinivasan, M., Chellappa, R., Baird, E., & Lamont, R. (2006). *Motion based correspondence for 3D tracking of multiple dim objects.* Paper presented at the Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on.

Vianna, D. L., & Carrive, P. (2012). Visualisation of Thermal Changes in Freely Moving Animals. In E. Badoer (Ed.), *Visualization Techniques* (Vol. 70, pp. 269-281): Humana Press.

Vianna, D. M., & Carrive, P. (2005). Changes in cutaneous and body temperature during and after conditioned fear to context in the rat. *Eur J Neurosci, 21*(9), 2505-2512. doi:10.1111/j.1460-9568.2005.04073.x

Vogel, B., Wagner, H., Gmoser, J., Wörner, A., Löschberger, A., Peters, L., . . . Frantz, S. (2016). Touch-free measurement of body temperature using close-up thermography of the ocular surface. *MethodsX, 3*, 407-416.

von Ziegler, L., Sturman, O., & Bohacek, J. (2021). Big behavior: challenges and opportunities in a new era of deep behavior profiling. *Neuropsychopharmacology, 46*(1), 33-44.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience, 2018*.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., & Schmid, C. (2009). *Evaluation of local spatio-temporal features for action recognition.* Paper presented at the BMVC 2009-British Machine Vision Conference.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L. (2016). *Temporal segment networks: Towards good practices for deep action recognition.* Paper presented at the European conference on computer vision.

Wang, W., & Neumann, U. (2018). *Depth-aware cnn for rgb-d segmentation.* Paper presented at the Proceedings of the European Conference on Computer Vision (ECCV).

Wang, X., Miao, Z., Zhang, R., & Hao, S. (2019). *I3d-lstm: A new model for human action recognition.* Paper presented at the IOP Conference Series: Materials Science and Engineering.

Wang, Z., Mirbozorgi, S. A., & Ghovanloo, M. (2015). *Towards a kinect-based behavior recognition and analysis system for small animals.* Paper presented at the Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE.

Wang, Z., Mirbozorgi, S. A., & Ghovanloo, M. (2018). An automated behavior analysis system for freely moving rodents using depth image. *Med Biol Eng Comput, 56*(10), 1807-1821. doi:10.1007/s11517-018-1816-1

Weissbrod, A., Shapiro, A., Vasserman, G., Edry, L., Dayan, M., Yitzhaky, A., . . . Kimchi, T. (2013). Automated long-term tracking and social behavioural phenotyping of animal colonies within a semi-natural environment. *Nature communications, 4*(1), 1-10.

Wiltschko, A. B., Johnson, M. J., Iurilli, G., Peterson, R. E., Katon, J. M., Pashkovski, S. L., . . . Datta, S. R. (2015). Mapping sub-second structure in mouse behavior. *Neuron, 88*(6), 1121-1135.

Wu, A., Buchanan, E. K., Whiteway, M., Schartner, M., Meijer, G., Noel, J.-P., . . . Schaffer, E. (2020). Deep Graph Pose: a semi-supervised deep graphical model for improved animal pose tracking. *BioRxiv*.

Wutke, M., Schmitt, A. O., Traulsen, I., & Gültas, M. (2020). Investigation of Pig Activity Based on Video Data and Semi-Supervised Neural Networks. *AgriEngineering, 2*(4), 581-595.

Xu, Y., Dong, J., Zhang, B., & Xu, D. (2016). Background modeling methods in video analysis: A review and comparative evaluation. *CAAI Transactions on Intelligence Technology, 1*(1), 43-60. doi:10.1016/j.trit.2016.03.005

Xudong, Z., Xi, K., Ningning, F., & Gang, L. (2020). Automatic recognition of dairy cow mastitis from thermal images by a deep learning detector. *Computers and Electronics in Agriculture, 178*, 105754.

Yang, X., Zhang, C., & Tian, Y. (2012). *Recognizing actions using depth motion maps-based histograms of oriented gradients.* Paper presented at the Proceedings of the 20th ACM international conference on Multimedia.

Ye, W., Cheng, J., Yang, F., & Xu, Y. (2019). Two-stream convolutional network for improving activity recognition using convolutional long short-term memory networks. *IEEE Access, 7*, 67772-67780.

Yuan, H., Liu, C., Wang, H., Wang, L., & Sun, F. (2022). Optimization and comparison of models for core temperature prediction of mother rabbits using infrared thermography. *Infrared Physics & Technology, 120*, 103987.

Yuan, Y., Chen, X., & Wang, J. (2019). Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*.

Zethof, T. J., Van Der Heyden, J. A., Tolboom, J. T., & Olivier, B. (1994). Stress-induced hyperthermia in mice: a methodological study. *Physiology & behavior, 55*(1), 109-115.

Zhang, G.-W., Shen, L., Li, Z., Tao, H. W., & Zhang, L. I. (2019). Track-Control, an automatic video-based real-time closed-loop behavioral control toolbox. *BioRxiv*.

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., . . . Manmatha, R. (2020). Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*.

Zhang, L., Zhu, G., Shen, P., Song, J., Afaq Shah, S., & Bennamoun, M. (2017). *Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition.* Paper presented at the Proceedings of the IEEE International Conference on Computer Vision Workshops.

Zhang, Z. Y. (2012). Microsoft Kinect Sensor and Its Effect. *IEEE multimedia, 19*(2), 4-10. doi:Doi 10.1109/Mmul.2012.24

Zhou, S., & Xu, L. (2018). *Mouse Behavior Recognition Based on Convolution Neural Network.* Paper presented at the 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER).

Zia, S., Yuksel, B., Yuret, D., & Yemez, Y. (2017). *RGB-D object recognition using deep convolutional neural networks.* Paper presented at the Proceedings of the IEEE International Conference on Computer Vision Workshops.

Zilkha, N., Sofer, Y., Beny, Y., & Kimchi, T. (2016). From classic ethology to modern neuroethology: overcoming the three biases in social behavior research. *Current opinion in neurobiology, 38*, 96-108.

Zimmermann, C., Schneider, A., Alyahyay, M., Brox, T., & Diester, I. (2020). Freipose: A deep learning framework for precise animal motion capture in 3d spaces. *BioRxiv*.

Zörner, B., Filli, L., Starkey, M. L., Gonzenbach, R., Kasper, H., Röthlisberger, M., . . . Schwab, M. E. (2010). Profiling locomotor recovery: comprehensive quantification of impairments after CNS damage in rodents. *Nature Methods, 7*(9), 701-708.

Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., & Wang, Q. (2019). Robust lane detection from continuous driving scenes using deep neural networks. *IEEE transactions on vehicular technology, 69*(1), 41-54.

APPENDIX

1. Fundamentals of Time-of-Flight operation system

Time-of-flight (ToF) cameras provide technology for the acquisition of range information, in real-time, into a single and compact device at a low cost. *Microsoft Kinect v2* uses this optical ToF technology to measure distances and produce, simultaneously, depth maps and RGB images. By using a laser or light pulse, it determines the distance measuring the signal time of flight between the camera (emitter) and the subject (target), for each point of the image. The distance d to be measured is proportional to the time travelled by the illumination source (E. Lachat et al., 2015), i.e., the phase difference between the radiated and reflected IR waves, $\Delta\varphi$, (Hansard, Lee, Choi, & Horaud, 2012) (Figure A 1.1), as follows:

$$d = \frac{c}{2}\frac{\Delta\varphi}{2\pi f} \tag{6.1}$$

where *f* denotes waves' frequency.

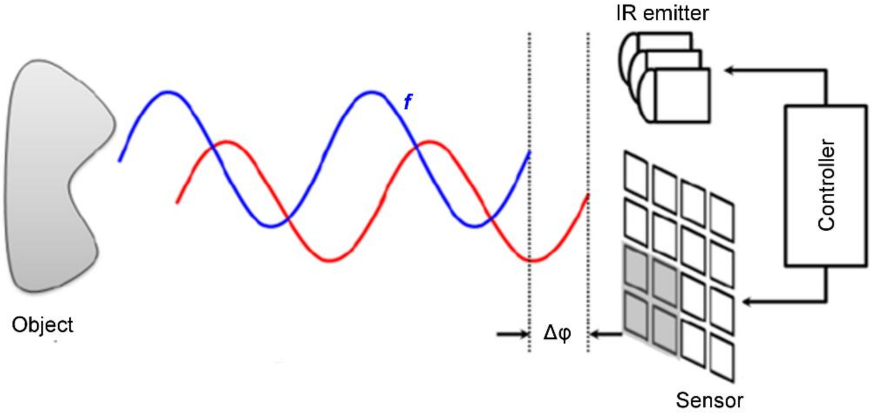


Figure A 1.1 Time-of-flight principle for measuring distances: the infrared (IR) wave indicated in red is directed to the target object, and the sensor detects the reflected IR component. The phase delay between emitted and reflected IR signals is measured to calculate the distance. Image adapted with permission from Hansard et al. (2012).

1.1 Representation of 3D coordinates

Once the distance to the corresponding point in the scene is calculated using the ToF principle, this information can be used to estimate the three-dimensional (3D) structure directly. In order to model the real space, it is necessary to calculate the 3D coordinates of each point in the scene.

The first step is to calculate the depth map from the distance map, provided by the ToF sensor. According to Figure A 1.2, given the specific point *P*, its distance *d* from the camera center *C* (outputted by the ToF sensor), the focal length f_l of the camera and the distance *x* from the principal point to point *P* projected on image plane, the distance *z* from camera center to point P_c can then be calculated. Distance *z* is the desired output value to populate the depth map, extracted from the RGB-D sensor. In this sense, each pixel depth value represents the distance to the plane that contains the object point and is perpendicular to camera principal axis: P_c - *C* line.

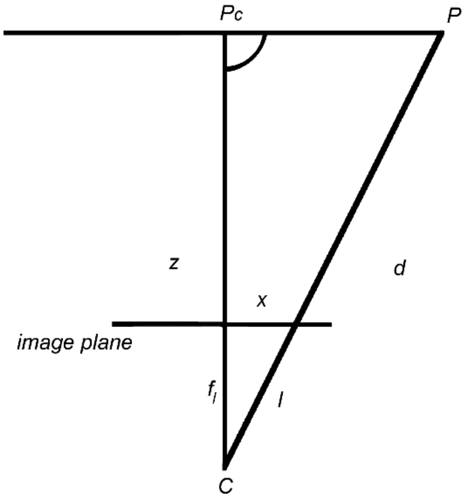


Figure A 1.2 Representation of depth calculation. *P* – point in scene; *d* – distance from the camera center *C*; $\boldsymbol{f_l}$ – focal length of the camera; *x* – distance from the principal point to the projection of *P* on the image plane, *Pc*; *z* – distance from the camera center to *Pc*. Image adapted with permission from Valgma (2016).

First, it is necessary to calculate the distance l from *C* to *P* projection on image plane:

$$l = \sqrt{{f_l}^2 + x^2} \tag{6.2}$$

And finally, distance *z* can be obtained as follows:

$$\mathrm{z} = \mathrm{d}\frac{f_l}{l} = \mathrm{d}\,\frac{f_l}{\sqrt{{f_l}^2 + \mathrm{x}^2}} \tag{6.3}$$

In order to describe the mathematical relationship between the coordinates of a 3D point and its projection onto the image plane, and, ultimately, to calculate the true 3D coordinates for the desired points, the standard pinhole camera model can be used in this type of camera. This model can only be used as an approximation of the 2D image from a 3D scene, since it does not take into consideration geometric distortions or blurring effects. Lenses aberrations' analysis will be addressed in Section A 1.3.

The pinhole camera model is based on the representation illustrated in Figure A 1.3, where d_C is the distance between the center of projection and the image plane, principal axis corresponds to the line starting from the center of projection and perpendicular to the image plane and the principal point is the intersection between principal axis and image plane.

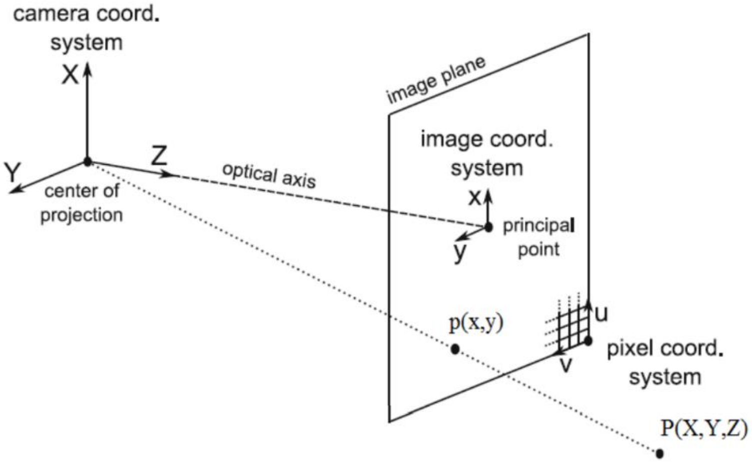


Figure A 1.3 Standard pinhole camera model. $\boldsymbol{d_C}$- distance between center of projection and the image plane; *(u, v)* – pixel coordinate system; *P(X ,Y, Z)* - 3D coordinates of point *P*; *p(x, y)* – coordinates of point *P* on the camera image plane. Image adapted with permission from Valgma (2016).

If (X,Y,Z) are the 3D coordinates of point *P*, the corresponding coordinates on the camera image plane (*p(x,y)*) are given by:

$$x = d_C \frac{X}{Z} \qquad y = d_C \frac{Y}{Z} \tag{6.4}$$

In order to transform these real world coordinates into pixel coordinates, in the pixel coordinate system given by *(u,v)* a scale factor, pixels per millimeter, should be given: (k_u, k_v). Also, the coordinates of the principal point in the pixel coordinate system are required: ($-x_0$, $-y_0$). In this sense the coordinates of point *p* can be calculated in the pixel coordinate system:

$$u = k_u(x + x_0) = k_u\, d_C\, \frac{X}{Z} + k_u x_0 \tag{6.5}$$

$$v = k_v(y + y_0) = k_v\, d_C\, \frac{Y}{Z} + k_v y_0 \tag{6.6}$$

In matrix form, it can be written as follows:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} k_u\, d_C & 0 & k_u\, x_0 \\ 0 & k_v\, d_C & k_v\, y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = KP' \tag{6.7}$$

where the coordinates of point *P* were normalized, by dividing them with its *Z* coordinate:

$$P' = \frac{P}{Z} = \begin{pmatrix} X/Z \\ Y/Z \\ 1 \end{pmatrix} = \begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} \tag{6.8}$$

The set of values in the matrix *K* are commonly called camera intrinsic parameters, and are usually denoted as:

Focal length: $\alpha_u = k_u d_C$, $\alpha_v = k_v d_C$

Coordinates of the principal point: $u_0 = k_u x_0$, $v_0 = k_v y_0$

These parameters can be found out by calibration, and, in the case of *Microsoft Kinect* sensors, the software development kit (*SDK*) already provides this information. Finally, considering that *u, v, K* and *Z* are known for point *P*, the coordinates *X* and *Y* can be calculated:

$$P = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = Z\,K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{6.9}$$

SDK for *Microsoft Kinect* sensors provides a look-up table of (γ_u, γ_v) values for each pixel *(u, v)*:

$$X = \gamma_u.Z \quad Y = \gamma_v.Z \tag{6.10}$$

And so, the *x*- and *y*- coordinates can be easily obtained, for the perfect pinhole camera model (Valgma, 2016).

1.2 Alignment of depth and color sensor

The correspondence between RGB and depth sensors can be done by a transformation matrix *T*, since depth and color information are captured by different fixed sensors, and there is a rigid transformation from depth coordinates to RGB coordinates. In this sense, and since RGB camera can also be modelled as a pinhole camera, the expression that calculates the coordinates of *P* in RGB camera coordinate system is given by (Changhee Kim, Yun, Jung, & Won, 2015; Valgma, 2016):

$$Z_c {K_c}^{-1} \begin{pmatrix} u_c \\ v_c \\ 1 \end{pmatrix} = \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \tag{6.11}$$

where X_c, X_c and X_c are point *P* coordinates in RGB camera coordinate system, and K_c are the camera intrinsic parameters matrix for RGB camera.

The correspondence can, then, be obtained as follows:

$$T \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \tag{6.12}$$

Using depth coordinates expression, the equation that describes the mapping between depth and color camera can be written as:

$$T\ Z\ K^{-1}\begin{pmatrix}u\\v\\1\end{pmatrix} = Z_c K_c{}^{-1}\begin{pmatrix}u_c\\v_c\\1\end{pmatrix} \tag{6.13}$$

where all parameters can be obtained by calibration.

Similarly, *Microsoft Kinect SDK* provides mapping functions to simplify depth and color registration (Changhee Kim et al., 2015; Valgma, 2016).

1.3 Distortion analysis

The perfect pinhole camera model is usually updated to take into consideration lenses distortion, and the importance of this adaptation increases from the center of the image to the edges as lens distortion effects increase. The most common and significant type of distortion is called radial distortion, which can be further classified as *barrel* distortion or *pincushion* distortion (Figure A 1.4).

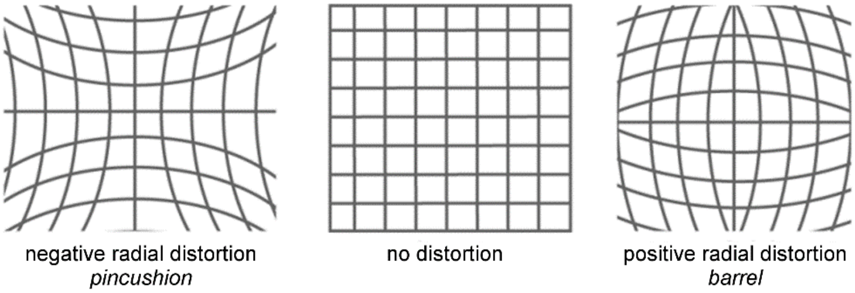


Figure A 1.4 Radial distortions: pincushion (left) and barrel (right) distortions, when compared to no distortion (middle).

Since it is primarily dominated by low order radial components, radial distortion can be corrected using relative simple models, such as *Brown*'s distortion model. This model can be used to corrected for both radial and tangential distortions, but for the particular case of RGB-D images generated by *Microsoft Kinect* sensors, it is only necessary to consider radial distortions.

In this sense, the correction of the ideal (distortion free) coordinates (*(x,y)*) using *Brown*'s model is given by:

$$x_d = x.(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{6.14}$$

$$y_d = y.(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{6.15}$$

where (x_d, y_d) are the real observed coordinates, $r = \sqrt{x^2 + y^2}$, and k_1, k_2 and k_3 are radial distortion coefficients. *Microsoft Kinect* SDK provides the first three coefficients for radial distortion but, as far as the author knows, it is not clear if they are automatically used to correct for distortion (Valgma, 2016).

2. Fundamentals of Support Vector Machines

Support Vector Machine (SVM), pioneered by Vapnik (1999) is a discriminative and binary classification method that works by optimizing a linear decision surface based on the concept of risk minimization. In this margin-type classifier, for a linearly separable dataset, the key idea is to find the optimal decision boundary (hyperplane) that maximizes the gap (concept of geometric functional margin) to the classes' closest points (support vectors). The boundary between classes is then obtained by a weighted combination of support vectors. Formally, the binary dataset of *N* samples used to train the SVM algorithm is represented by $X = \{(x_i, y_i) | \, x_i \in \mathbb{R}^m, y_i \in \{-1,\ 1\}\}_{i=1}^{N}$, where x_i corresponds to the *i*-th data point in the *m*-dimensional real space $\mathbb{R}^m$, and $y_i \in \{-1,\ 1\}$ represents its class label from one of two classes. For a linearly separable training dataset, the linear decision surface can then be formalized as:

$$w \cdot x + b = 0 \tag{6.16}$$

where *w* is the normal vector to these planes and *b* determines their location relative to the origin. The optimization problem consists in finding the hyperplane, parametrized by Equation (6.16). Geometrically, the optimal hyperplane maximizes the sum of the distances to the closest positive and negative training samples (this sum is referred to as the margin of the separating hyperplane) (Figure A 2.1). The problem of maximizing the margin $\frac{2}{\|w\|}$ is equivalent to the problem of minimizing $\frac{1}{2}\|w\|^2$ subject to constraints that ensure class separability (all training samples x_i are correctly classified). This quadratic optimization problem, denoted as hard-margin SVM formulation (Figure A 2.1A), can be expressed as follows:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \tag{6.17}$$

$$\text{subject to } y_i \,(w \cdot x_i + b) \ \geq 1, \ \ i = 1, \dots, N$$

When the training samples are not linearly separable (e.g., noisy data, outliers), slack variables ξ_i can be introduced to the constraints to allow misclassification of difficult or noisy data points and the violation of the separation constraints to a certain degree.

In this case, the objective function, known as soft-margin linear SVM formulation (Figure A 2.1B), can be described as follows:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i}^{N} \xi_i \tag{6.18}$$

$$\text{subject to } y_i \, (w \cdot x_i + b) \, \geq 1 - \xi_i,$$

$$\xi_i \, \geq 0, i = 1, \ldots, N$$

where $C > 0$ tunes the trade-off between minimizing the sum of the slack errors and maximizing the margin (Bishop, 2006; Gutschoven & Verlinde, 2000).

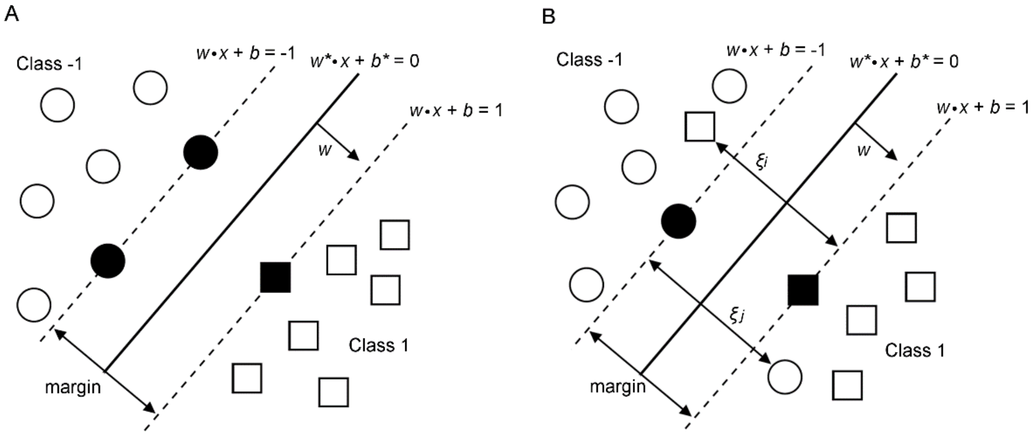


Figure A 2.1 Support Vector Machine (SVM) representation. A. Hard-margin SVM in a linearly separable dataset. **B.** Soft-margin SVM in a nonlinearly separable dataset. Square and circle symbols represent data points from positive and negative classes, respectively. Filled symbols denote the support vectors. $\boldsymbol{\xi_i}$ represent slack variables. Margin = $\frac{\mathbf{2}}{\|\boldsymbol{w}\|}$. Image inspired by OpenCV ; Theodoridis and Koutroumbas (2003).

This SVM formulation is solved via its Lagrangian dual problem, written in terms of multipliers, α_i, and both data and slack variables become implicitly represented: data is represented by a kernel matrix, *K*, of all inner products between pairs of data points ($K\left(x_i, x_j\right) = \left\langle x_i, x_j \right\rangle$), and each slack variable is associated with a Lagrangian multiplier.

When using Lagrangian multipliers method, the dual problem can be formulated as:

$$\max_{\alpha_i} \sum_{i}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j K\left(x_i, x_j\right), i, j = 1, \dots, N \tag{6.19}$$

constrained to $\sum_i \alpha_i y_i = 0$ and $0 < \alpha_i < C$. Once the optimal solutions α^* are determined, both parameters w^* and b^* that define the optimal hyperplane can be calculated:

$$w^* = \sum_{i}^{N} \alpha_i y_i x_i \tag{6.20}$$

$$b^* = y_i - w^* \cdot x_i \tag{6.21}$$

Finally, the classifier can be defined as follows:

$$g(x) = w^* \cdot x_i + b^* = \sum_{i}^{N} \alpha_i{}^* y_i K(x_i, x) + b^* \tag{6.22}$$

In some cases, data is not linearly separable in the original feature space, but may be linearly separable in high-dimensional spaces (Figure A 2.2). In order to operate in higher dimensions, kernel methods, which use kernel functions, can be employed, without actually mapping the input points into the high-dimensional space. The so-called kernel trick allows for an efficient mapping by manipulating the kernel function $K(\cdot,\cdot)$ presented in the dual formulation (Gutschoven & Verlinde, 2000). Some of the most commonly used kernel functions are as follows:

Linear SVM: $K\left(x_i, x_j\right) = x_i \cdot x_j$

Polynomial SVM of degree p: $K\left(x_i, x_j\right) = (x_i \cdot x_j + 1)^p$

Radial-Basis Function (RBF) or Gaussian SVM: $K\left(x_i, x_j\right) = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}$

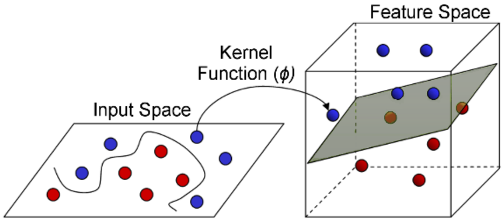


Figure A 2.2 Kernel trick for non-linearly separable datasets. For a linear Support Vector Machine: $g(x) = w \cdot x + b$. In a higher-dimensional feature space: $\tilde{g}(x) = w \cdot \phi(x) + b$. Adapted with permission from Pisner and Schnyer (2020).

3. Fundamentals of Neural Networks and Convolutional Neural Networks

Neural Networks (NN) (also known as Multilayer Perceptrons (MLPs)) represent the general foundation of deep learning methods. Deep learning is a representation-learning method where useful features (or representations) are directly learned from the raw input data, instead of relying on hand-crafted feature extractors (as in the traditional machine learning methods). Multiple levels of representation are obtained by composing simple non-linear modules that iteratively transform the raw input into representations with higher and more abstract levels (Yann LeCun et al., 2010; Voulodimos et al., 2018).

In NN, several units or neurons are arranged in layers, comprising an input layer, an output layer and several hidden layers (Figure A 3.1). Formally, the output of the *i*-th unit within layer *l* is given by a weighted sum of the neuron's activations (outputs) in the previous layer (*l* - 1), plus a constant bias, followed by a nonlinear activation function $\phi(\cdot)$:

$$a_i{}^{(l)} = \phi^{(l)}(z_i{}^{(l)}) \tag{6.23}$$

$$\text{with } z_i{}^{(l)} = \sum_{k=1}^{m^{(l-1)}} w_{i,k}{}^{(l)}\, a_k{}^{(l-1)} + b_i{}^{(l)}$$

where $a_k{}^{(0)} = x_k = [x_1, \dots, x_D]$ corresponds to the *k*-th input feature with *D* input units, $w_{i,k}{}^{(l)}$ corresponds to the weighted connection (or weights) from the *k*-th neuron in layer (*l-1*) to the *i*-th neuron in layer *l*, $b_i{}^{(l)}$ denotes the bias, and $m^{(l)}$ corresponds to the number of units in layer *l*. The most commonly used activation functions include the sigmoid function:

$$\phi(z) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{6.24}$$

or the hyperbolic tangent function:

$$\phi(z) = tanh(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}} \tag{6.25}$$

or, the recently introduced Rectified Linear Unit (ReLU) function (Glorot, Bordes, & Bengio, 2011):

$$\phi(z) = \mathrm{ReLU}(z) = \max(0, z) \tag{6.26}$$

For classification tasks, in which the neurons of the last layers output class posterior probabilities, the activation function usually is the softmax function, defined as:

$$\sigma(z)_i = \frac{\exp(z_i)}{\sum_{j=1}^{C}\exp(z_j)} \tag{6.27}$$

where $\sigma(z)_i$ denotes the output of the *i*-th neuron in the output layer, and represents the probability of a given instance belonging of the class *i*, in a total number of classes *C* (Bengio, 2009).

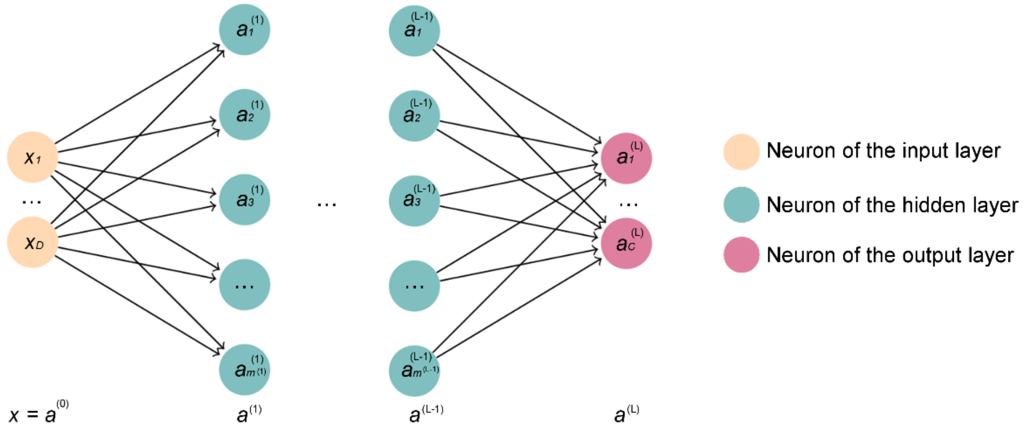


Figure A 3.1 Fully Connected Neural Network representation, with *L* layers, *D* input units and *C* output units. The *l*-th layer contains *m(l)* hidden units.

During the learning process, also called training, all trainable parameters of the network, θ, are adjusted to minimize a given loss (cost or objective) function. This objective function measures the error (or distance) between the output scores of the neural network, $a\ (\cdot, \theta)$ and the desired target output, y. For classification problems, the most commonly used objective function is the categorical cross-entropy (with *K* output classes):

$$\mathcal{L}(\theta) = -\sum_{n=1}^{N}\sum_{k=1}^{K} y_n(\mathrm{k})\ \log\ \left(a_k(x_n,\theta)\right) + (1 - y_n(k))\ \log\left(a_k(x_n,\theta)\right) \tag{6.28}$$

where *N* is the number of training examples, and the output y_n is coded as 1-of-*K*. Generically, to adjust the internal parameters (or weights), the learning algorithm computes a gradient vector that, for each weight, measures what would be the modification of the error if that weight were increased by a small amount. The weights are then adjusted in the opposite direction to the gradient vector (Y. LeCun et al., 2015).

Usually, the learning algorithm consists of multiple iterations, or epochs, with two phases: forward pass, where the signal flows from the neurons in the input layers to the neurons in the output layer, and the current weights are used to compute this signal; backward pass, for the optimization of the neural network's trainable parameters. This optimization is usually performed using a gradient descent method. Stochastic Gradient Descent (SGD) is the most commonly used optimization procedure, and consists of iteratively computing the outputs and the errors after the forward pass, computing the average gradient for those input examples (formally, the partial derivatives of the loss function with respect to all trainable parameters) (Y. LeCun et al., 2015; Yann LeCun et al., 2010), and adjust the weights accordingly, as follows:

$$\Delta\theta^t = \alpha \frac{\partial E(\theta^t, X_i)}{\partial \theta^t} \tag{6.29}$$

$$\theta^{t+1} \leftarrow \theta^t - \Delta\theta^t \tag{6.30}$$

where θ^t denotes the parameters at epoch t, X_i represents the mini-batches (or small sets of examples randomly sampled from the overall training dataset X), and α is the learning rate, an hyperparameter that defines how far the parameters should be updated in the optimization space.

Convolutional Neural Networks (CNNs) are a particular type of deep neural network designed to process data that come in the form of multiple arrays, such as time-series (one-dimension (1D)), images (2D), or videos or volumetric images (3D). CNNs follow an architecture similar to conventional NN, and take advantage of different properties to deal with array-form data: local connections, shared weights, and different types of layers (convolutional and pooling layers). A typical CNN architecture is comprised of two blocks (Figure A 3.2): a set of L alternating pairs of convolutional and pooling layers, also known as the feature extraction block, where the activations after the convolution and pooling operations are stored in feature maps; followed by fully connected layers (as in MLPs) in the classification/regression block. The main goal of the last block is to provide a prediction based on the feature maps produced by the feature extraction block (Goodfellow, Bengio, & Courville, 2016; Y. LeCun et al., 2015; Yann LeCun et al., 2010).

In the convolutional layers, the input data is convolved with a set of kernels (or learnable filters) to produce an activation map (or feature map). The activation at a spatial location (i, j) in layer *l* can be computed as follows:

$$a(i,j)^l = \phi\left(\sum_{m=0}^{K-1}\sum_{n=0}^{K-1} W_{m,n}\, a_{i+m,j+n}{}^{l-1} + b^l\right) \tag{6.31}$$

where $\phi\,(\cdot)$ is the non-linear activation function, b^l is the bias, a^{l-1} denotes the incoming activation map, and *W* corresponds to the trainable filter with a kernel size of $K \times K$ pixels (Voulodimos et al., 2018).

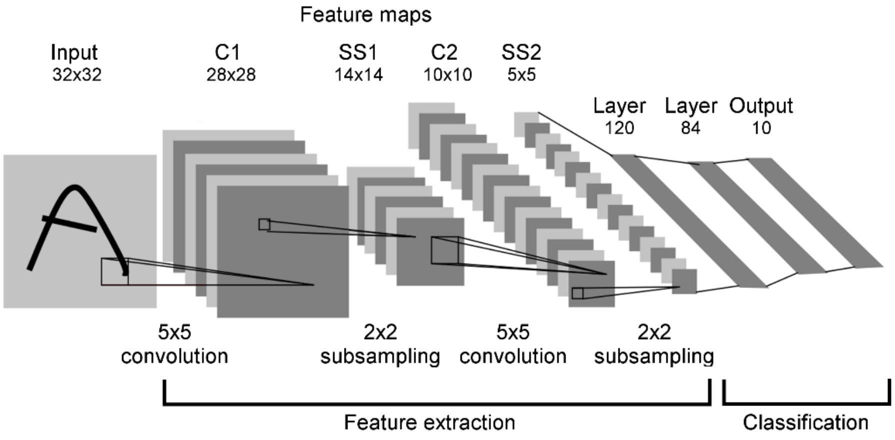


Figure A 3.2 Standard architecture of a Convolutional Neural Network for classification tasks, with *C* convolutional layers, *SS* pooling layers, with the corresponding feature maps' sizes. Image adapted with permission from Yann LeCun, Bottou, Bengio, and Haffner (1998).

Unlike conventional NN where all neurons in one layer are fully connected to every neuron in the next layer, in CNNs the neurons are locally connected to small regions of the input data (local receptive fields). These receptive fields are related to the filter size and slide across the entire input structure. This way, neurons are capable of combining local neighborhood information to detect local features, that are iteratively combined by the subsequent convolutional layers. Besides that, each activation map within the output volume shares the same weights and bias, allowing the detection of the same feature in different locations in the input data (Goodfellow et al., 2016; Y. LeCun et al., 2015; Yann LeCun et al., 2010).

In the pooling layers, semantically identical features produced by the convolutional layers are merged into one, downsampling the output information of the neuron. The advantage is that by computing summary statistics over local regions of the feature maps, feature representations and, ultimately, final predictions are robust to small variations in the input space (Goodfellow et al., 2016; Y. LeCun et al., 2015). One common type of pooling is called Max Pooling, where only the maximum activation value is kept for each local region in the feature map, and passed to the next convolutional layer.

4. Deep Learning Approaches for Learning from Depth-Sensing Information

This section depicts the experiments and corresponding results that were obtained to support the experimental work described in CHAPTER 5.

4.1 Semantic Segmentation with U-Net – segmentation of animal’s whole-body

Although the most common task when using CNNs is classification, it sometimes becomes essential to identify/label specific regions/objects in images to characterize them, extract measurements, or assist in further classification. This is particularly relevant in biomedical imaging, allowing, for example, reducing the time required to run diagnostic tests, identifying lesions, or, in the specific case of behavioral neuroscience, identifying animals for further analysis or tracking. Unlike the classification problems in which the objective is to make predictions for the whole input, in semantic segmentation, a class is assigned to each pixel of the image. In this way, semantic segmentation achieves fine-grained inference by making dense predictions, and that is why it is one of the high-level tasks that paves the way towards complete scene/image understanding. A general semantic segmentation architecture can be constructed as an encoder followed by a decoder network. In this structure, the encoder downsamples the spatial resolution of the input, creating lower-resolution feature mappings that are learned to be highly efficient at discriminating between classes. The encoder is usually a pre-trained classification network, such as *VGG*/*ResNet*/*AlexNet*, followed by a decoder network, which upsamples the feature representations into a full-resolution segmentation map to get a dense classification. The goal of the downsampling steps is to capture semantic/contextual information, whereas the upsampling goal is to recover spatial information. To fully recover the fine-grained spatial information lost during downsampling and reconstruct accurate shapes for segmentation boundaries, it is common to use skip connections that pass information from the downsampling to the upsampling steps. Consequently, features are merged from different resolution levels and it helps to combine spatial with context information (Ronneberger et al., 2015).

Different model architectures were initially proposed to perform semantic segmentation, using several approaches as part of the decoding mechanism and built on top of powerful CNN backbone architectures. The Fully Convolutional Network (FCN) was first used by

Long, Shelhamer, and Darrell (2015) as a solution for semantic segmentation. This network was trained end-to-end, learning to map from pixels to pixels, and using the *AlexNet* model as the encoder module of the network. The encoder is a stack of convolutional and max-pooling layers, in which the decoder module was appended with transpose convolutional layers to upsample the coarse feature maps into a full-resolution segmentation map.

To improve the predictions of FCN and increase the resolution at the boundaries due to loss of information from the encoding, Ronneberger et al. (2015) proposed the U-Net architecture, which is built upon the FCN and modified to obtain a contracting path (encoder) that captures context and a symmetric expanding path (decoder) that enables precise localization (Figure A 4.1). Similar to FCN, the U-Net architecture uses various blocks of convolution and max-pooling layers applied to the input image. In turn, every step in the expansive path consists of an upsampling of the feature map followed by convolutions that halves the number of feature channels and concatenation with the correspondingly cropped feature map from the contracting path. Because of its symmetry, the network has a larger number of feature maps to propagate context information to higher resolution layers. Besides being computationally less demanding than the FCN model, when combined with data augmentation, it is possible to train the U-Net with a few training examples (in the original paper, the authors reached an average intersection-over-union (IOU) of 92% using only 35 partially annotated training images from the *PhC-U373* dataset of the ISBI cell tracking challenge 2014 and 2015). This simpler architecture has grown to be commonly used and widely adapted for a variety of semantic segmentation problems, where the stacked convolutional layers were substituted by different blocks, such as residual blocks (Drozdzal, Vorontsov, Chartrand, Kadoury, & Pal, 2016), dense blocks (Jégou, Drozdzal, Vazquez, Romero, & Bengio, 2017), or even using different types of convolutional techniques (L.-C. Chen, Papandreou, Kokkinos, Murphy, & Yuille, 2017; L.-C. Chen, Papandreou, Schroff, & Adam, 2017).

A different approach to semantic segmentation, and also object detection, includes region-based methods that follow a common pipeline. The first stage, called Region Proposal Network (RPN), extracts free-form regions from an image and proposes candidate object bounding boxes. The next stage extracts features from each candidate and performs region-based classification and bounding box regression. This model, named Fast R-CNN, and proposed by Girshick (2015), was later modified to improve training and testing speed, and also increase detection accuracy (S. Ren, He, Girshick, & Sun, 2015).

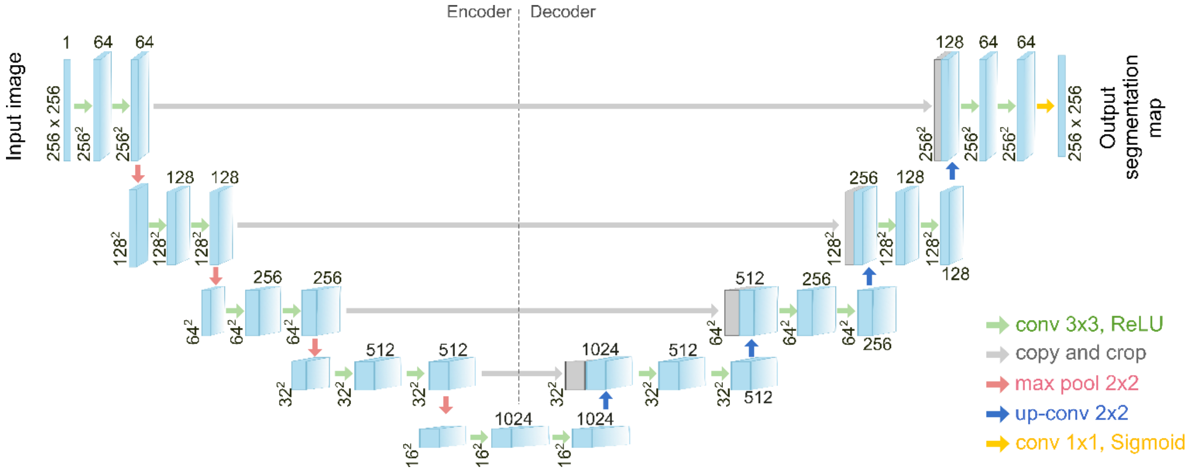


Figure A 4.1 U-Net architecture (example for 16x16 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map, in which the number of channels is denoted on top of the box and the corresponding x-y-size at the lower left edge of the box. The network is divided into two convolutional parts: encoder, on the left, and the decoder, on the right. Grey boxes in the decoder section represent copied feature maps. The arrows denote the different operations, subtitled in the image itself. Inspired by Ronneberger et al. (2015).

Mask R-CNN, also developed by the Girshick team (He, Gkioxari, Dollár, & Girshick, 2017) is based on these last improved models but adds complexity by simultaneously generating a high-quality segmentation mask. Therefore, this model architecture is able to go even further in the segmentation task and be applied for instance segmentation: detection and identification of each object of interest appearing in an image at the pixel level. This is why Mask R-CNN model architecture is one of the most representative works of region-based methods for object segmentation. Following the previous pipeline, this model uses selective search to extract a large number of object proposals, and then through the computation of CNN features for each of them, it performs classification. To perform semantic segmentation, in addition to a class label, and a bounding box, Mask R-CNN also includes a third branch that outputs the object mask, through an FCN. In this sense, Mask R-CNN combines two different networks: Faster R-CNN and FCN, and, formally, a multi-task loss function is defined for the whole model as a combination of classification, bounding box and mask generation losses.

Current state-of-the-art methods that lead the performance ranking for solving semantic segmentation problems applied object region representations' techniques. In particular, one of the methods that achieved the best performance on several segmentation benchmarks was developed by Y. Yuan, Chen, and Wang (2019), and applies the object-contextual representation (OCR) concept to improve the performance of the semantic segmentation. OCR is the weighted aggregation of all the object region representation with the weights calculated according to the relations between pixels and object regions, following the motivation that the class label assigned to one particular pixel is the category of the object that that pixel belongs to. Therefore, it is considered a late-stage processing algorithm, only using deep networks to learn the division of pixel area into a set of object regions by their corresponding class.

Another promising network architecture was recently developed by Zhang and a group of researchers from Amazon and UC Davis in 2020 (H. Zhang et al., 2020). It is also one of the state-of-art performance architectures for image classification, object detection, instance and semantic segmentation. In this work, the researchers suggest a new *ResNet*-like network architecture, called *ResNeSt*, that incorporates attention across groups of feature maps. The created Split-Attention block is a computational unit composed of the feature-map group and split-attention operations that, when stacked together with multiple of these blocks, creates the new *ResNeSt* network architecture (Figure A 4.2A). In more detail, initially, all features are divided into *K* groups (being *K* the cardinality hyperparameter). Within each cardinal group, the network calculates the attention across all splits inside it (*R* stands for the radix hyperparameter that gives the number of splits within a cardinal group). Each split attention is multiplied by a feature map and added together to generate the output cardinal representation (Figure A 4.2B). These final *K* representations are then concatenated along the channel dimension, and the results are added to the input through a shortcut connection, as in a regular *ResNet*. With these modifications, the overall *ResNet* structure is maintained in a simple modular network, without adding additional computational effort.

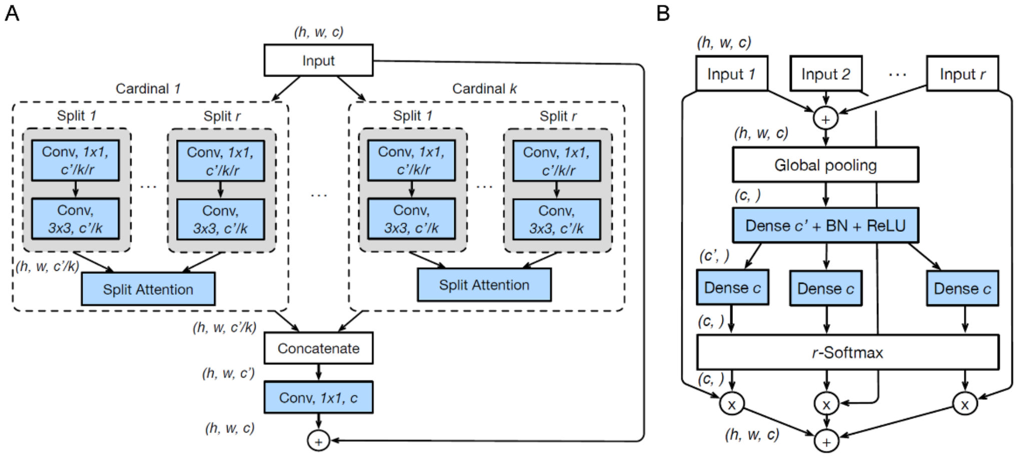


Figure A 4.2 *ResNeSt* block and detailed view of the Split-Attention unit. A. *ResNeSt* block. For simplicity, it is shown the *ResNeSt* block in cardinality-major view (the feature map groups with same cardinal group index reside next to each other). **B**. Split-Attention within a cardinal group. Adapted with permission from H. Zhang et al. (2020).

4.1.1 Experiments

To prove that the RGB-D dataset presented in CHAPTER 5 can be used for segmentation and classification tasks using deep learning methods, the performance of a single U-Net architecture was initially evaluated in its ability to semantically segment purely depth images. The objective is to separate animal's body in a frame-by-frame approach, eliminating the background.

Although there are already other segmentation methods that currently outperform the initially proposed by Ronneberger et al. (2015), this architecture was chosen for being a simple approach to the semantic segmentation problem (the goal is to classify only two classes) while being quick to train using augmentation techniques.

Dataset

A subset of 500 depth frames (16-bit depth; 512x424 pixels) were selected from the original RGB-D dataset and manually segmented to create segmentation masks (ground-truth). The labeling process was performed in two stages: primary automatic segmentation and manual correction. In the first step, a sequence of image processing techniques was applied to obtain primary masks. Linear thresholding was used to eliminate depth values bigger than 2000 (2 meters; camera-derived errors) and *Canny Edge* detection algorithm was applied to perform edge detection on the depth images. The resulting mask was processed using morphological operations to remove small objects and merged with a second mask obtained after *Otsu* histogram thresholding method. The final mask was obtained after applying morphological operations to fill existent holes. To fine-tune this primary automatic segmentation, the masks were corrected using the *ImageLabeler* application in *MATLAB*.

Finally, and before training experiments, all depth images were normalized in the interval [0, 255], all frames were converted to 8-bits, re-scaled to the input size of the U-Net (256x256), and normalized again ([0, 1]). The depth frames' subset was split into 300 images for training, 100 images for validation, and 100 images for testing. After obtaining the predicted masks, these were post-processed, using image analysis techniques to eliminate small-sized particles/objects left by thresholding the probability maps.

Implementation Details

All experiments were performed using the publicly available U-Net model (Ronneberger et al., 2015). It consists of 23 convolutional layers. The contracting path consists of repeated two 3x3 convolutions (unpadded convolutions), each block followed by a ReLU, and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step, the number of feature channels is doubled. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. At the final layer, a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes, in this case, 2: background and foreground (animal) (Figure A 4.3).

The input images, with size 256x256, and their corresponding segmentation masks were used to train the U-Net network, with SGD implementation of *TensorFlow* and ADAM optimizer, with an initial learning rate of 0.0001 (that is reduced by a factor of 0.75 after 5 iterations, and training is stopped after 10 iterations if validation loss doesn't decrease, for a maximum of 300 iterations), a momentum of 0.9, dropout in all layers equal to 0.5 and batch size of 10.

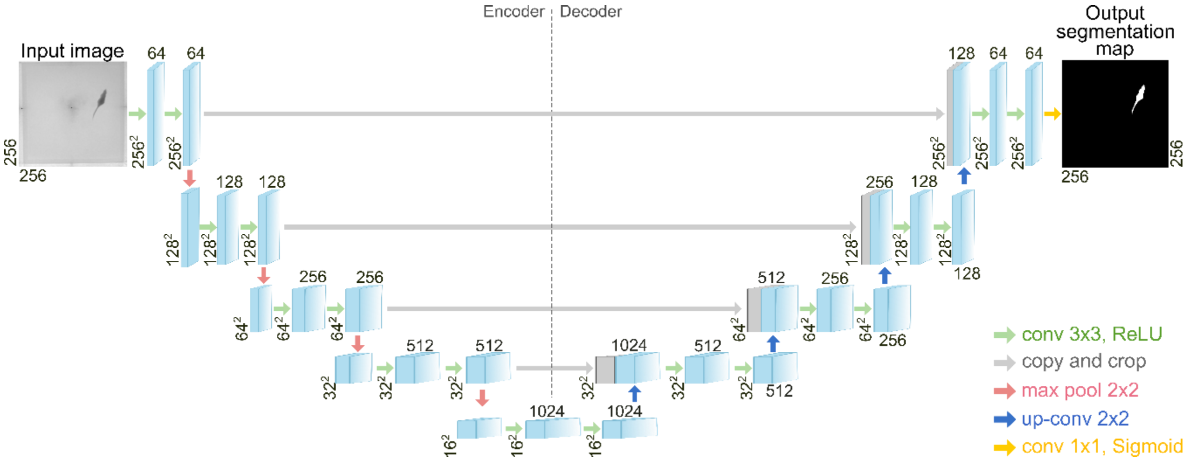


Figure A 4.3 U-Net architecture for a frame extracted from the original RGB-D dataset (with input size of 256x256x1). Each blue box corresponds to a multi-channel feature map, in which the number of channels is denoted on top of the box and the corresponding x-y-size at the lower left edge of the box. The network is divided into two convolutional parts: encoder, on the left, and the decoder, on the right. Grey boxes in the decoder section represent copied feature maps. The arrows denote the different operations, subtitled in the image itself. Inspired by Ronneberger et al. (2015).

Four different loss functions were tested for model evaluation. Binary cross-entropy (BCE) loss measures the performance of a classification model and calculates the loss by computing the following average:

$$BCE\ (y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i\ \log(\ \hat{y}_i\ ) + (1 - \ y_i)\ \log(\ 1 - \hat{y}_i\ ) \tag{6.32}$$

where $\hat{y}_i$ is the i-th scalar value in the model output (predicted value), y_i the corresponding target value (true value) and N is the total number of samples. The model is trained using BCE loss to minimize the entropy between two probabilistic distributions by penalizing misclassification.

Dice binary cross-entropy and Jaccard binary cross-entropy combine BCE and Dice or Jaccard coefficients (Dice BCE or Jacc BCE), respectively. Dice and Jaccard coefficients are performance metrics commonly used in semantic segmentation problems with class imbalance (for instance, when the ratio between background and foreground pixels is very high). Dice coefficient, also known as the Sørensen-Dice coefficient, measures the overlap area between the prediction and the ground truth:

$$Dice\ (y, \hat{y}) = \frac{2\ \sum_{i,j} y_{i,j} \hat{y}_{i,j}}{\sum_i \sum_j \hat{y}_{i,j}\ +\ \sum_i \sum_j y_{i,j}} \tag{6.33}$$

In turn, Jaccard Index (also named IoU) is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between both, given by:

$$Jacc\ (y, \hat{y}) = \frac{\sum_{i,j} y_{i,j} \hat{y}_{i,j}}{\sum_i \sum_j \hat{y}_{i,j}\ +\ \sum_i \sum_j y_{i,j}\ - \sum_{i,j} y_{i,j} \hat{y}_{i,j}} \tag{6.34}$$

In this sense, when these metrics are combined with BCE to formulate different loss functions, the objective is to minimize the entropy while maximizing the overlap between the predicted and ground truth class (actually, to achieve the same objective, it is usual to minimize ($1 -\ Dice\ (y, \hat{y})$) or ($-Jacc\ (y, \hat{y})$)):

$$Dice\ BCE\ (y, \hat{y}) = 0.5\ BCE\ (y, \hat{y}) + 0.5\ (1 - Dice\ (y, \hat{y})) \tag{6.35}$$

$$Jacc\ BCE\ (y, \hat{y}) = 0.5\ BCE\ (y, \hat{y}) - Jacc\ (y, \hat{y})) \tag{6.36}$$

These metrics were also used to evaluate and compare the performance of the models and 3 independent tests were carried out for performance comparison, maintaining the training conditions previously described. Training of a single network run took ~20 mins (a set of 50/60 epochs), using an *NVIDIA GeForce* RTX 2080 GPU (8 GB RAM).

Another common strategy to improve deep learning results and, simultaneously, avoid overfitting by increasing model's generalization ability, is called data augmentation (Shorten & Khoshgoftaar, 2019). This technique allows for a significant increase in the diversity of the data available for training, without collecting new data, rather than transforming the already available one. By introducing additional variability to the dataset, the augmented data will represent a more comprehensive set of possible data points, approaching overfitting problems from the first step - the training step. In this case, the data augmentation effect on training the model was evaluated, and the dataset was augmented by random rotation (30 degrees), zooming (0.2), and vertical and horizontal translation (10% of total height/width).

4.1.2 Results

Table A 4.1 shows preliminary results of the median validation loss, Dice and Jaccard coefficients obtained after training with different loss functions. The U-Net model was initially trained with different loss functions (with or without augmentation), for 3 experiments each.

Table A 4.1 Segmentation performance results. Summary of the performance on the test set of different methods on the semantic segmentation task, using different loss functions. Reported values are expressed as median values [95% confidence interval] (N = 3 consecutive runs; optimal threshold; after post-processing). BCE – Binary cross-entropy; Dice BCE – Binary cross-entropy combined with Dice Loss; Jacc BCE – Binary cross-entropy combined with Jaccard Loss.

		Validation Loss	**Dice Coefficient**	**Jaccard Coefficient**
NO AUGMENTATION	BCE	**0.01** [0.002; 0.018]	**76.7** [75.6; 77.8]	**60.5** [56.6; 64.45]
	Dice BCE	**0.07** [0.04; 0.10]	**79.9** [76.0; 78.0]	**66.5** [63.8; 69.23]
	Jaccard BCE	**-0.77** [-1.06; -0.48]	**83.6** [74.5; 92.7]	**71.8** [58.9; 84.7]
AUGMENTATION	Dice BCE	**0.04** [0.01; 0.07]	**90.1 [85.2; 95.0]**	**82.0 [74.1; 90.0]**
	Jaccard BCE	**-0.84** [-0.87; -0.81]	**88.9 [88.5; 89.3]**	**80.0 [79.5; 80.5]**

The objective was to check if the RGB-D dataset could be learned by such a model and to gain intuition about what parameters would be best for this type of model and data. In general, and without augmentation methods, the model trained with Jaccard BCE outperformed the other methods, yielding an overall Dice and Jaccard performance of 83.6% and 71.8%, respectively. When combining augmentation techniques, it did result in an increased performance of more than 5%, with the augmented Dice BCE model attaining the best performance of 90.1% and 82.0% on the test set.

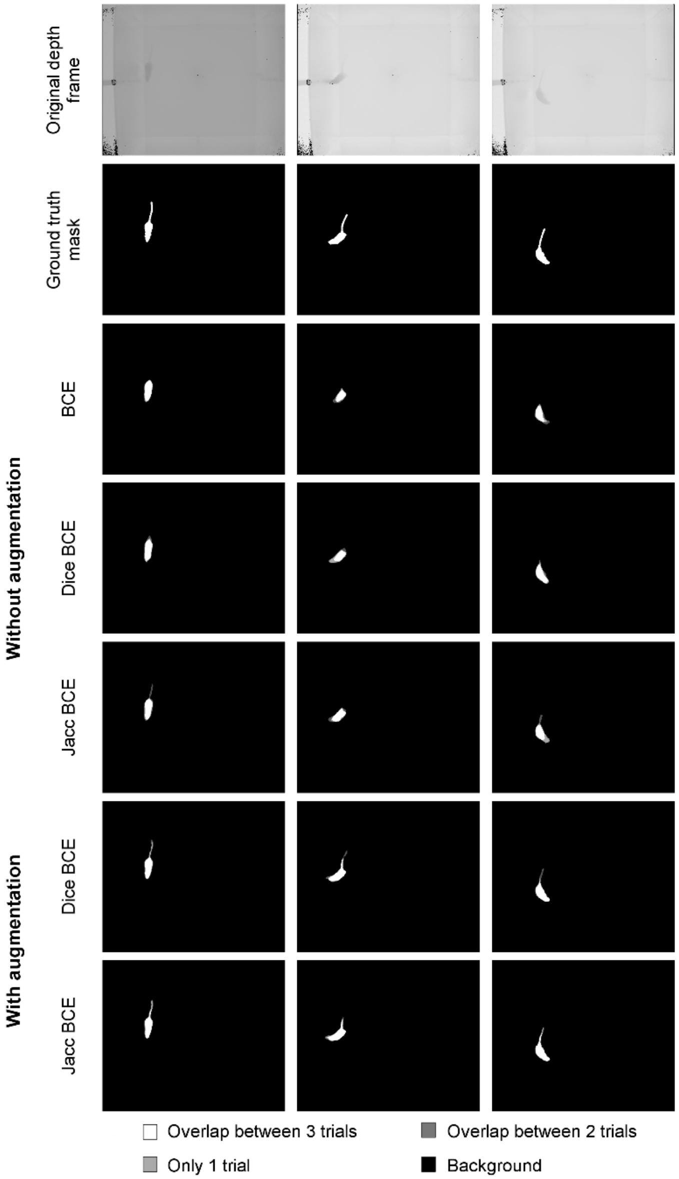


Figure A 4.4 Semantic segmentation results of the different trained models. Comparison between predicted and ground-truth masks of 3 different depth frames of the testing set, for 3 different loss functions. Predicted masks from the 3 trials are merged to show variability between different training runs. BCE – Binary cross-entropy; Dice BCE – Binary cross-entropy combined with Dice Loss; Jacc BCE – Binary cross-entropy combined with Jaccard Loss.

Although the performance in the training set was lower (in terms of Dice and Jaccard coefficients, and training loss) in augmented models (data not shown), that doesn't happen when comparing performance on the testing set. In this sense, the trained model is not closely reproducing a particular set of the training data and failing to fit new additional data, but rather learning how to generalize and predict future observations. Qualitatively, when analyzing the probability maps and binary masks predicted by the trained model with no augmentation (Figure A 4.4), the animal's body was poorly segmented. In particular, animal's tail was usually eliminated or cropped in the final masks.

Also, the BCE method was quantitatively worse than the remaining ones, with more animal's body pixels eroded, absence of tail, less detail in the paws and head, and irregular body shape. With augmentation, either method keeps the animal's body better defined. In particular, the Dice BCE method performed better at distinguishing the totality of animal's tail, and the final masks were cleaner and more detailed (no pre-processing required) (Figure A 4.4).

Indeed, these results confirm that with augmented data the model holds better its generalization capability when tested with new unseen data (testing set), being the best-trained model for learning and segmenting the recorded depth frames. From here on, augmentation techniques were always applied to the dataset.

4.2 Depth representation for Convolutional Neural Networks

CNNs were designed to mimic the biological behavior of the visual cortex and, therefore, initially applied to analyze RGB 2D visual imagery, as natural camera images (Krizhevsky, Sutskever, & Hinton, 2012). In this sense, although RGB images can be directly used as inputs for the CNNs, the same may not happen with depth images. Depth data encodes distance from the sensor to the scene and so, the information of each pixel is of a different nature, when compared to RGB images. Nonetheless, when looking at a rendered version of a depth image, it is possible to recognize some visual features, such as corners, edges, or shaded regions, that are also present in RGB images. Thereby, the question that arises is how should a depth image be encoded for use in CNNs so that it can learn more meaningful features. Does the absence of color and texture information of depth images weaken the discriminative representation and learning power of CNN models? Or, given this intrinsic nature of depth data and geometric similarity to RGB frames, will CNNs learn as effectively when using raw depth images without any encoding?

Different depth encoding methods for CNN feature learning have been proposed in the literature (Figure A 4.5), and depending on the modality adopted for recognition, the architectures are broadly categorized into three groups: single stream depth-based, single-stream 4-channel RGB+D-based, or combining RGB and depth networks in a fusion-based architecture. Following the idea that no additional depth pre-processing is required for CNN input, different methods use raw depth frames for recognition problems (mainly, human action recognition or object recognition), whether in a single stream (Shah, Bennamoun, Boussaid, & While, 2017), 4-channel (Couprie, Farabet, Najman, & LeCun, 2013) or fusion-based architecture (Che & Peng, 2018; Z. Liu, Zhang, & Tian, 2016; Rahman, Tan, Xue, Shao, & Lu, 2019; Socher, Huval, Bath, Manning, & Ng, 2012). In general, as required by network input, the depth data is rendered to grayscale (Figure A 4.5A and B), normalized or standardized, and, if necessary, the grayscale values are replicated to the three channels of the input.

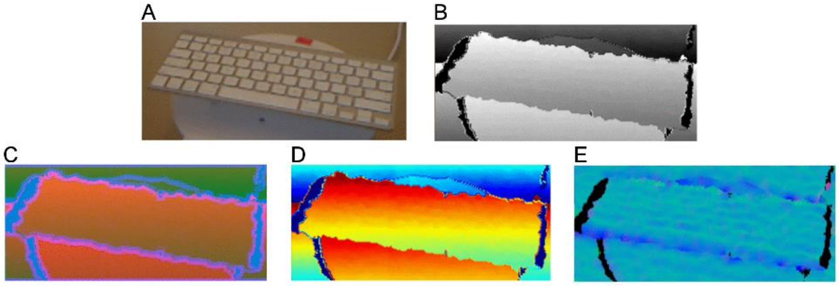


Figure A 4.5 Different approaches for color encoding of depth images. A. RGB, **B.** depth-gray, **C.** HHA, **D.** depth-jet coloring, and **E.** surface normal encodings. Adapted with permission from Zia, Yuksel, Yuret, and Yemez (2017).

Another commonly used method for depth encoding is called HHA encoding. Initially proposed by Gupta, Girshick, Arbeláez, and Malik (2014), this method encodes the depth image in three channels at each pixel with horizontal disparity, height above ground, and the angle between pixel's local surface normal and the inferred gravity direction (Figure A 4.5C). The authors trained the network initially proposed by Krizhevsky et al. (2012) with input data transformed using the HHA encoding, following the hypothesis that there is enough common structure between the proposed HHA geocentric images and corresponding RGB images. In this sense, a network that was initially designed and trained for RGB images can also learn a suitable representation for HHA images. Other authors followed this hypothesis, using HHA depth-based features to learn for RGB-D object and scene recognition (Eitel et al., 2015; Song, Herranz, & Jiang, 2017; Thermos, Papadopoulos, Daras, & Potamianos, 2020), person re-identification (L. Ren, Lu, Feng, & Zhou, 2017), and image segmentation (Tresch, 2016).

The depth jet-encoding, initially used by Eitel et al. (2015), is an alternative and also prevalent method for depth encoding. Here, a jet colormap is applied to the given depth image, transforming the one-channel depth map to a three-channel color image. The depth information is distributed over the three RGB channels, being converted to color values ranging from red (near scene pixels) over green to blue (far scene pixels) (Figure A 4.5D). As in the previous method, the authors believe that as the chosen network is designed for RGB images, this colorization method will provide enough similar structure between the RGB and the corresponding depth image, improving feature representation learning. Other studies explored depth-jet coloring potential in encoding depth frames for automatic object recognition, in a two-stream CNN architecture (Kumar, Shrivatsav, Subrahmanyam, & Mishra, 2016; Thermos et al., 2020).

Another interesting approach to explore depth information, typically in action recognition, is by using the concept of motion representation. The motion information is encoded through a sequence of images/video in a single 2D visual representation, generated by accumulating the motion energy through the entire sequence. If this representation is composed of a sequence of depth frames, it is called Depth Motion Map (DMM) and it was originally introduced by Yang, Zhang, and Tian (2012). DMMs are obtained by projecting the depth frames onto three orthogonal Cartesian planes, and a set of frames are combined to obtain the spatial energy distribution map that represents the movements. In order to preserve subtle motion information, C. Chen et al. (C. Chen, Liu, & Kehtarnavaz, 2016; C. Chen et al., 2017), first, and later other authors (Dawar, Ostadabbas, & Kehtarnavaz, 2018; Elboushaki et al., 2020; Singh et al., 2020) extended and improved the initial concept of DMM to calculate the motion energy by accumulating the absolute difference between consecutive frames (Figure A 4.6A):

$$DMM_{(f,s,t)} = \sum_{t=0}^{N-2} \left| D\,(x,y,t+1)_{\,(f,s,t)} - D(x,y,t)_{\,(f,s,t)} \right| \tag{6.37}$$

where $D(x,y,t)_{\,(f,s,t)}$ corresponds to a depth video sequence with N number of frames and for the three projection planes: f – front view; s – side view; t – top view.

These representations can then be used to describe the shape and motion cues of a particular depth action sequence and can be fed as inputs to CNN-based networks to perform action/gesture recognition in RGB-D image sequences.

Finally, less explored encoding methods have also been described in the literature. One strategy uses surface normals to represent form and surface structure (Madai-Tahy et al., 2016). The gradients for each pixel are calculated and each dimension of the calculated surface normal vector corresponds to one channel in the resulting image ($x \rightarrow R, y \rightarrow G, z \rightarrow B$) (Figure A 4.5E). The final values are transformed to contain natural numbers between 0 and 255 using the following operations:

$$\left(R_{final}, G_{final}, B_{final}\right) = \left(\frac{x+1}{2} \times 255, \quad \frac{y+1}{2} \times 255, \quad z \times 255\right) \qquad (6.38)$$

The voxel grid concept is also used to encode depth information (Zia et al., 2017). The RGB pixel is placed at the same position (coordinates) in the voxel, but at its corresponding depth layer/channel (Figure A 4.6B). In this way, it is possible to combine color and depth information in a single metric, maintaining both the texture and the spatial distribution of the pixels. The number of final channels is equal to the maximum depth value of the images, which in itself can be computationally intensive if fed to large networks.

A different idea introduced by W. Wang and Neumann (2018) is to incorporate depth information into the internal elements of a conventional CNN (convolution and max-pooling layers), instead of pre-processing the input depth data. The so-called Depth-aware CNN, applied in semantic segmentation problems, uses information from the depth images to construct similarity kernels, based on points with similar depth values, and this information is incorporated in depth-aware convolutions and average pooling elements (Figure A 4.6C). The fact that pixels with the same semantic label and similar depth should have more impact on each other and may have similar segmentation labels motivated this depth-aware technique, as well as the fact that standard CNNs are limited to learning geometric transformations due to the fixed convolutional kernels' structure.

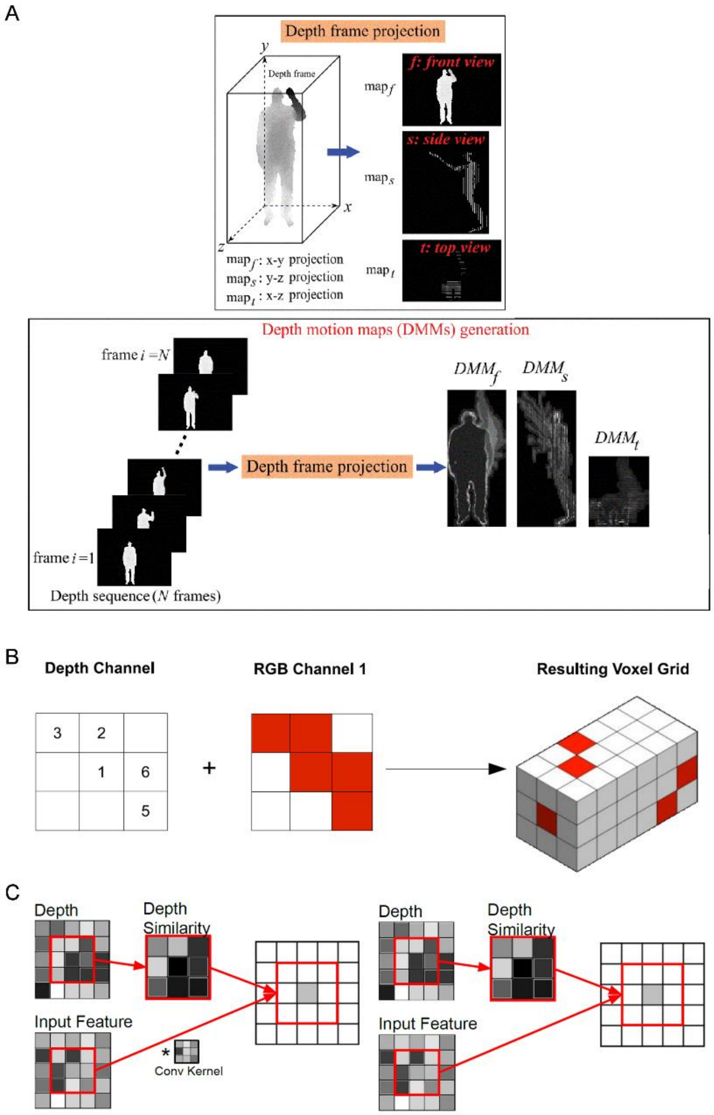


Figure A 4.6 Different depth-encoding approaches. A. Depth Motion Maps (DMMs) generation for a depth sequence. **B.** Illustration of how RGB-D images are converted to voxel representations for a 3×3 input image (fragment), where depth values are quantized into 6 intervals. **C.** Illustration of information propagation in Depth-aware CNN. Without loss of generality, we only show one filter window with kernel size 3×3. In depth similarity shown in the figure, darker color indicates higher similarity while the lighter color represents that two pixels are less similar in depth. Left: the output activation of depth-aware convolution is the multiplication of depth similarity window and the convolved window on input feature map. Right: the output of depth-aware average pooling is the average value of the input window weighted by the depth similarity. Images adapted with permission from: A. C. Chen et al. (2017), B. Zia et al. (2017), and C. W. Wang and Neumann (2018).

4.2.1 Experiments

In order to understand which depth input would allow better and faster learning of the features' representation by the U-Net model, four different encoding strategies for rendering depth information were compared.

Implementation Details

In addition to the raw depth frames, coloring and surface normals encoding methods were applied to the original dataset prepared for semantic segmentation (of 500 depth frames; 16-bit depth; 512x424 pixels). Besides standard jet colormap, previously described, the Turbo colormap (look-up table available at: Mikhailov (2019)) was also applied to the given depth images, encoding depth information into three color channels. This new colormap was originally reported as an improved colormap for visualization that outperforms the jet one in reducing false detail, color banding, and color blindness ambiguity. In this sense, it is argued as being more appropriate to distinguish fine details and perform visual semantic segmentation in depth images.

Acknowledging the fact that depth frames are usually affected by acquisition errors/noise, the raw frames were also filtered, calculating the pixel-by-pixel median between 3 and 5 consecutive frames, to eliminate those one-off fluctuations. Given the nature of our depth frames and the setup in which they were acquired (open-field 1 m x 1 m; minimum height of 1.20 m; Wistar rats), the dynamic range of depth values is very high. The maximum and minimum valid values of depth (excluding outliers/noise) are approximately equal to 2000 (2 meters) and 0, respectively. To improve the contrast of depth images and to understand how better CNN feature learning would be in this context, the dynamic range was reduced to 255, centered on the pixels belonging to the animal and truncated on the others (the manually annotated masks allowed this animal-centered processing). Before all training experiments, depth input images were normalized in the interval [0, 255], all frames were converted to 8-bits, re-scaled to the input size of the U-Net (256x256), and normalized again ([0, 1]).

All experiments were performed using the already described U-Net model, trained with SGD implementation of *TensorFlow* and ADAM optimizer, and Dice BCE loss function. The training consisted of a maximum of 300 iterations with a batch size of 10, an initial learning rate 0.0001 (which dropped by a factor of 0.75 after 5 iterations, and training is stopped after

10 iterations if validation loss doesn’t decrease), a momentum of 0.9, dropout in all layers equal to 0.5. Augmentation techniques were applied, with the same previously described parameters (Appendix - Semantic Segmentation with U-Net – segmentation of animal’s whole-body - Experiments).

Although only 3 independent tests were performed for each encoding method, the obtained results allowed initial filtering of the different techniques tested and choosing a subset to be further explored. In this way, for a more in-depth analysis of the performance of the selected models, the network was trained and tested 40 times for each model, keeping the same partition of the training and validation sets in a single run. Statistical analysis was performed using *GraphPad Prism* (*GraphPad* Software Inc., version 7.0, CA, USA). The method of *D'Agostino & Pearson* was used as a normality test, and parametric or non-parametric (paired) tests were chosen as appropriate. Statistical significance was considered for $p < 0.05$. Parametric data are expressed as mean ± standard deviation, and non-parametric data are expressed as median and 95% confidence intervals.

4.2.2 Results

Different encoding strategies were tested to understand if the U-Net model could learn meaningful features for the semantic segmentation task. Since the U-Net model proved successful when applied to the depth frames dataset, it is now interesting to analyze which depth encodings would work best in learning those features (Figure A 4.7).

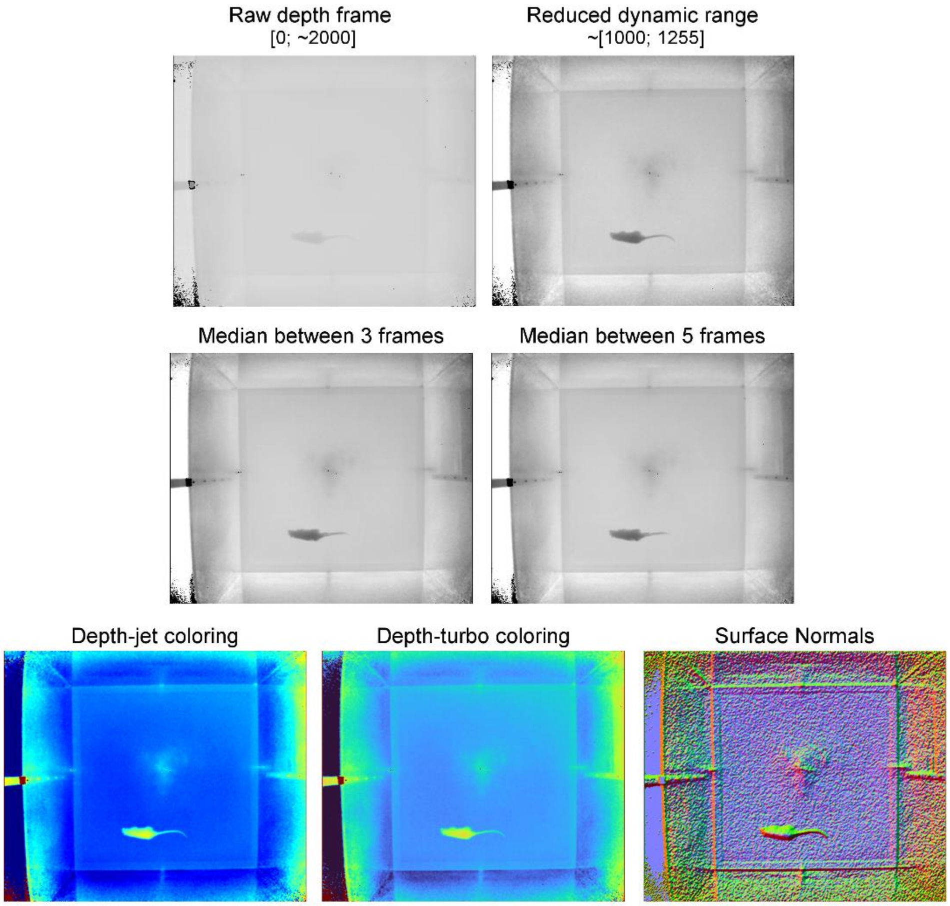


Figure A 4.7 Representative frames of the different types of depth encoding. Median, depth-jet, depth-turbo, and surface normal encodings' frames have reduced dynamic range.

Qualitatively, and as expected, when the dynamic range is reduced, the details of the image are more easily distinguished, such as the animal's tail and paws. When the median is performed between several consecutive frames, some noise is eliminated, mainly in the image corners, without affecting the overall structure and intensity distribution. Also, in the depth-jet and -turbo encodings, the fine details are even better distinguished, mainly in the depth-jet encoding, where the contrast between the background and the animal is clearer. Regarding the surface normals' frames, it is possible to visually identify the same local structure between pixels, even with irregularities.

Model's performances were quantified using Dice and Jaccard coefficients, after testing the U-Net model for the different depth encoding strategies (Table A 4.2). It is possible to see that, in general, decreasing the dynamic range improves the overall performance of the U-Net model, as expected. The model learns using increasingly optimized kernels, that detect spatially local input patterns. Thus, the higher the contrast, the easier it is to detect intensity transitions that can represent these feature patterns, so the higher the model's capability to learn how to recognize them. Also, when performing the median of some consecutive frames, the overall performance has not improved, when compared to the raw depth frames. In fact, in all independent tests (N = 3), the median of 5 frames slightly outperforms the median of 3 frames input network, but it was never superior when using raw depth frames. When the probability maps were analyzed (data not shown), both median methods showed more irregular maps and less clean masks. Taking into account that the computational power of this pre-processing is much higher (non-linear operations), these methods were no longer considered.

Table A 4.2 Segmentation performance results for different depth encoding methods. Summary of the performance on the testing set for different encoding methods and with the original and reduced dynamic depth range ([0, 255]). Reported values are expressed as median values [95% confidence interval] (N = 3 consecutive runs; optimal threshold; after post-processing).

	Dynamic Range	**Validation Loss**	**Dice Coefficient**	**Jaccard Coefficient**
Raw depth frames	Original	**0.04** [0.01; 0.07]	**90.1** [85.2; 95.0]	**82.0** [74.1; 90.0]
	Reduced	**0.03** [0.01; 0.04]	**93.5 [88.4; 98.6]**	**87.8 [78.8; 96.8]**
Median 3 frames	Original	**0.07** [0.02; 0.13]	**83.4** [80.4; 86.4]	**71.0** [66.4; 75.6]
	Reduced	**0.04** [0.01; 0.07]	**82.9** [73.7; 92.1]	**70.8** [57.0; 84.6]
Median 5 frames	Reduced	**0.05** [0.01; 0.08]	**83.5** [74.5; 92.5]	**71.7** [57.9; 85.5]
Depth-jet encoding frames	Original	**0.04** [0.03; 0.06]	**87.1** [84.6; 89.6]	**77.1** [73.2; 81.0]
	Reduced	**0.02** [0.02; 0.03]	**91.2** [85.5; 96.9]	**83.8** [74.3; 93.3]
Depth-turbo encoding frames	Original	**0.03** [0.02; 0.03]	**87.3** [78.3; 96.3]	**77.5** [63.7; 91.3]
	Reduced	**0.02** [0.02; 0.02]	**91.6** [87.7; 95.5]	**84.5** [77.8; 91.2]
Surface Normals frames	Original	**0.04** [0.02; 0.05]	**93.1** [71.3; -]	**81.3** [57.9; -]
	Reduced	**0.04** [0.02; 0.06]	**91.2** [84.3; 98.1]	**83.8** [72.6; 95.0]

Furthermore, even with some type of depth encoding, the method that uses raw depth frames as inputs of the network achieves a Dice and Jaccard coefficients value of 93.5% and 87.8% on the test set, respectively, which are the best results achieved on this classification task. The binary masks outputted by depth-jet and -turbo coloring methods (data not shown) show a more irregular animal body and often the presence of shadows, mainly during *rearing* events, which is less often using raw depth frames. On the other hand, although the normal surface method's masks have segmentation holes inside the animal and contain noise and sparse blobs, the shadows completely disappear, which is intrinsic to the geometric/surface structure information encoded by the surface normals' data. Overall, these experiments show that the CNN network can be trained for recognition of depth data using different depth encoding methods, but interestingly, the model without any pre-processing of the raw frames slightly outperforms all other depth encoding methods.

A recurring error common to all depth encoding methods occurred in frames containing *rearing* events, where the presence of shadows disturbs model learning (Figure A 4.8). To try to reduce the impact that this type of frames has on segmentation performance, 100 new frames containing *rearing* events were manually annotated and added to the training set. Although the performance metrics had no significant improvements quantitatively, the final masks were cleaner and contained fewer shadows' effects (data not shown). This new training set was maintained for future experiences.

Besides, the model's ability to learn using depth frames was further tested using frames without any foreground (in the absence of an animal) and in the presence of two animals (with overlapping or completely separated), examples of which are in Figure A 4.9. Empty frames were constructed by reflecting half of an original frame (in the absence of an animal). The model was then able to detect the absence of animals in empty frames (no foreground pixels on the predicted masks), but also the presence of two animals, even when overlapping (average Dice and Jaccard coefficients equal to 89.7% and average and 81.4%, respectively). These results prove model's ability to generalize and recognize scenarios different from those seen in the training set.

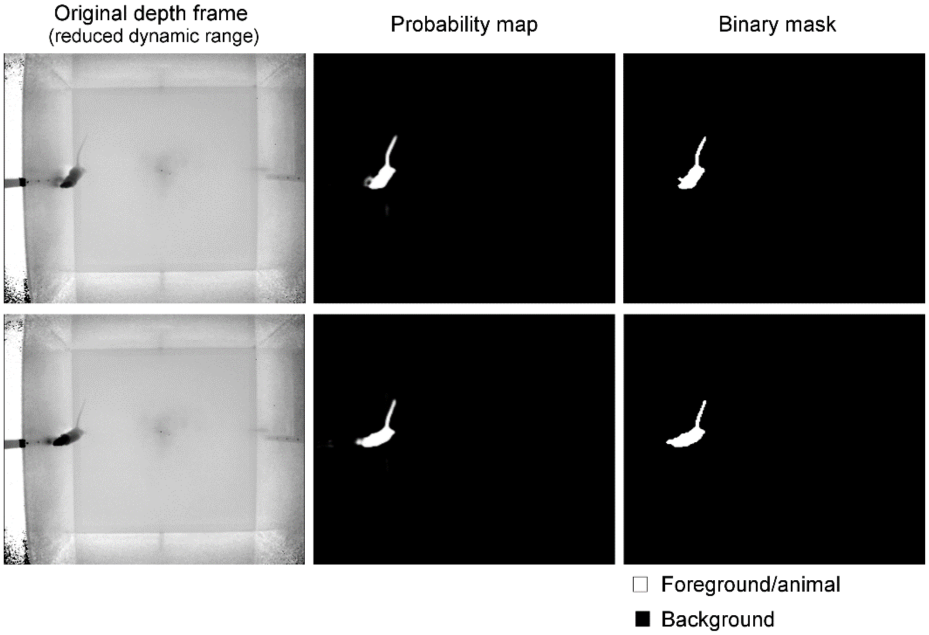


Figure A 4.8 Representative frames of recurrent errors during rearing events.

Given the obtained results, the 3 best methods were chosen for a detailed statistical analysis of performance: raw depth, depth-jet coloring, and surface normal frames' models. Depth-jet coloring method was chosen instead of depth-turbo coloring one since the performance differences were residual and the first is most commonly used in the literature. The performance of these 3 methods (Figure A 4.10) was statistically compared using the Dice and Jaccard coefficients evaluated in the test set, and the time until the model convergence was compared in terms of the number of epochs until the training stopped. This extensive analysis confirmed that the model trained with raw frames significantly improves the performance of the U-Net model, when compared to the two encoding methods ($p < 0.01$ and $p < 0.0001$ for depth-jet and surface normal encoding strategies, respectively), with median Dice and Jaccard coefficients of 93.7% ([92.0; 95.2]) and 88.2% ([85.2; 89.7]), respectively. Furthermore, the model trained with surface normal encoding does not appear to be superior to the one trained with depth-jet coloring (p = 0.28) and presents a greater variance of Dice and Jaccard metrics' values.

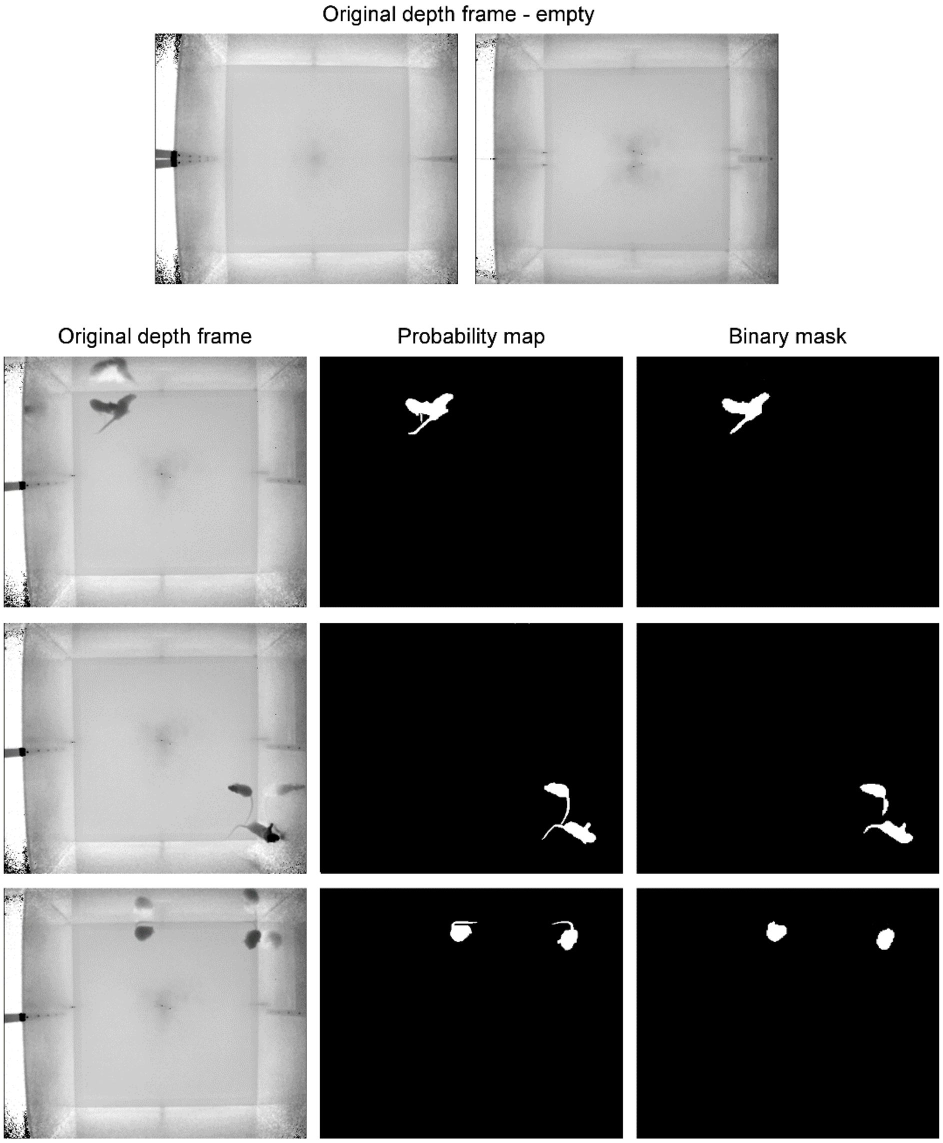


Figure A 4.9 Representative frames of model's ability to detect empty frames and frames with multiple animals. Above: Empty frames constructed by reflecting half of an original frame. Below: Frames containing two overlapping or completely separated animals and corresponding probability maps and binary masks.

Although outperforming the other methods, learning using raw depth frames is significantly slower, taking longer to converge ($p < 0.0001$), for the same stopping criteria. Nevertheless, and given that the computational power between encoding preprocessing can be leveled with the convergence training time, the raw depth frame-based method was selected as the best at learning how to semantically segment depth frames.

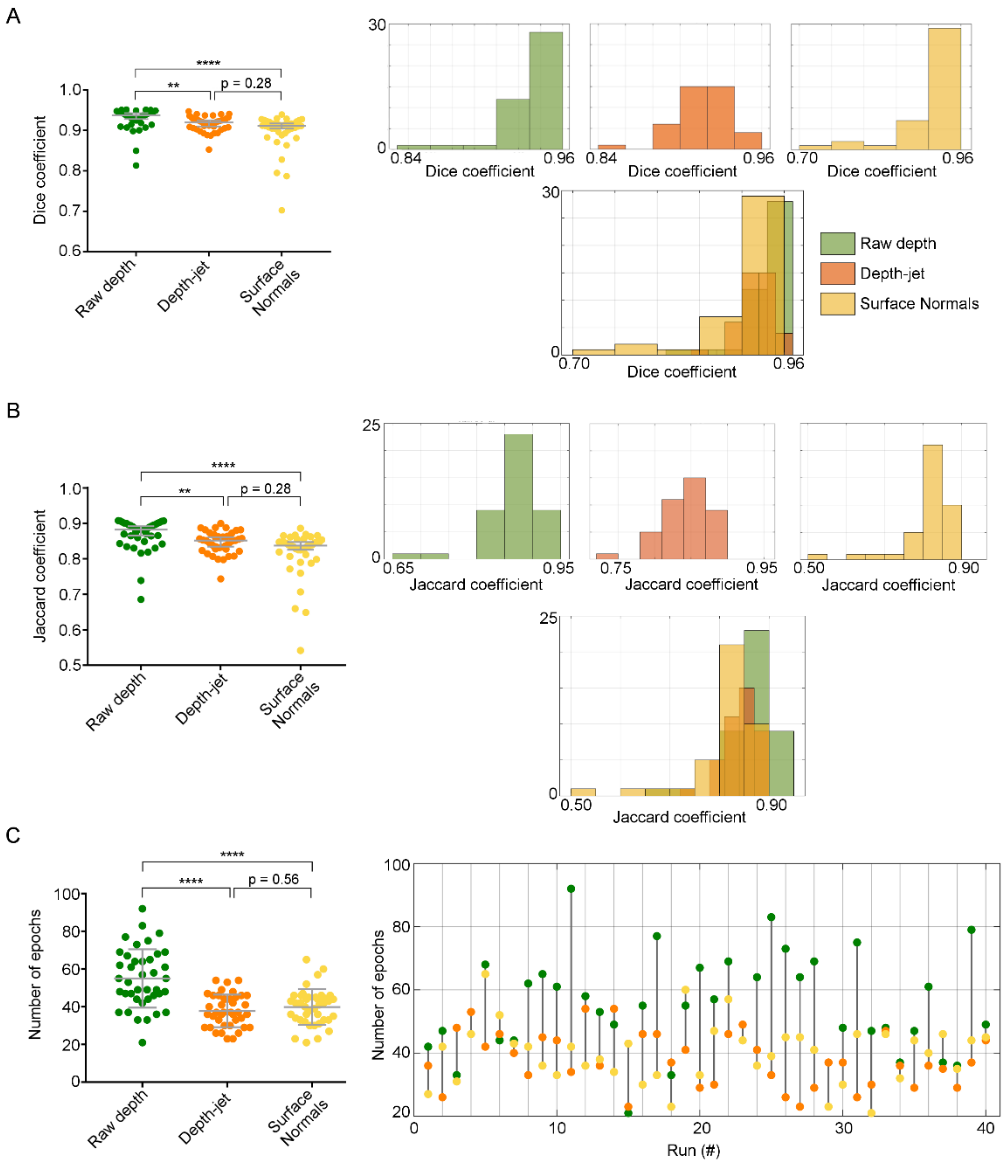


Figure A 4.10 Comparison between different depth encoding models' performance, regarding: **A.** Dice coefficient, **B.** Jaccard coefficient, and **C.** Number of epochs until convergence. **A.** and **B.** Dice and Jaccard coefficients' performance results, respectively, when trained with raw depth (green), depth-jet coloring (orange), and surface normals (yellow) frames (reduced dynamic range; results in the test set). Right: Point-wise distributions for the 40 paired trials. Left: Single histogram distributions (top) and overlap distributions (bottom) for the 3 models. Data represented as median ± 95% confidence interval. **C.** Right: Point-wise distributions of the number of epochs until convergence for the 40 paired trials. Left: Extended analysis of the number of epochs until convergence per event run. Data represented as mean ± standard deviation. ** $p < 0.01$; **** $p < 0.0001$.

4.3 Sequence Models for Extracting Spatiotemporal Features of Depth Sequences

When analyzing a video sequence, for different computer vision tasks, not only the spatial information within each frame is important but also its motion content across frames. In fact, in contrast with still image recognition tasks, the temporal information of video data can provide additional clues hidden in temporally neighboring frames for the recognition of actions/behaviors or segmentation of frames (Elboushaki et al., 2020; Simonyan & Zisserman, 2014). The advances in image recognition methods, mainly after the breakthrough of deep learning in still-image recognition, originated by the introduction of the *AlexNet* model, boosted video understanding research. After showing that CNNs are an effective class of models for understanding image content and for learning powerful and interpretable image features (Krizhevsky et al., 2012; Sermanet et al., 2013), those methods were adapted and extended to deal with video data.

The simplest approach that first emerged was to process each frame of the video sequence separately and apply CNNs to recognize actions or segment at the individual frame level. However, by using this approach, temporal information encoded in neighboring frames is not considered. In particular, for the semantic segmentation task of video sequences, as the temporal dependencies are ignored, the results may present temporal inconsistencies, caused by changes in environment illumination, fluctuations in pixel values (intrinsic to the acquisition device), or occlusions (Ji, Xu, Yang, & Yu, 2012; Rebol & Knöbelreiter, 2020). Another disadvantage when using individual frame level approach for semantic segmentation is that the semantic label of each voxel depends on the entire spatiotemporal context of the video, and labels differ from each other in terms of voxels; on the contrary, in classification tasks, labels depend mostly on the global video representation.

In this way, different research lines have been proposed to extend the connectivity of CNNs in the time domain and incorporate spatiotemporal features in video understanding methods. The first obvious approach applies 2D CNNs to extract spatial features from individual frames and later fuse the temporal information (Karpathy et al., 2014; Koller, Ney, & Bowden, 2016). In particular, Karpathy et al. (2014) explored four different fusion techniques, capable of combining temporal information in CNNs from different contiguous frames. The proposed technique that attained the best performance is called *Slow Fusion*, where higher layers get access to progressively more global information, spatially and temporally, by computing activations through temporal convolutions in addition to spatial convolutions (Figure A 4.11A). Although these architectures are easy to be fine-tuned on pre-trained models given the large availability of image annotated datasets, the temporal encoding is not considered during the feature learning stage (only spatial information) and the temporal order of the sequences tends to be neglected.

Another technique to learn spatiotemporal features is to process spatial and motion information in separated branches, using conventional 2D CNN, followed by a learned fusion of both information. These two-stream CNNs were first proposed by Simonyan and Zisserman (2014), where RGB and optical flow frames are used as spatial/appearance and motion information, respectively, and processed independently (Figure A 4.11B.). This technique has been explored in recent studies (Elboushaki et al., 2020; Feichtenhofer, Pinz, & Wildes, 2017; L. Wang et al., 2016; Ye, Cheng, Yang, & Xu, 2019), combining different spatial and motion (depth, optical flow, etc.) encodings, albeit with some drawbacks (optical/scene flow methods can be computationally expensive, and different branches are largely processed independently, preventing a more effective temporal feature learning).

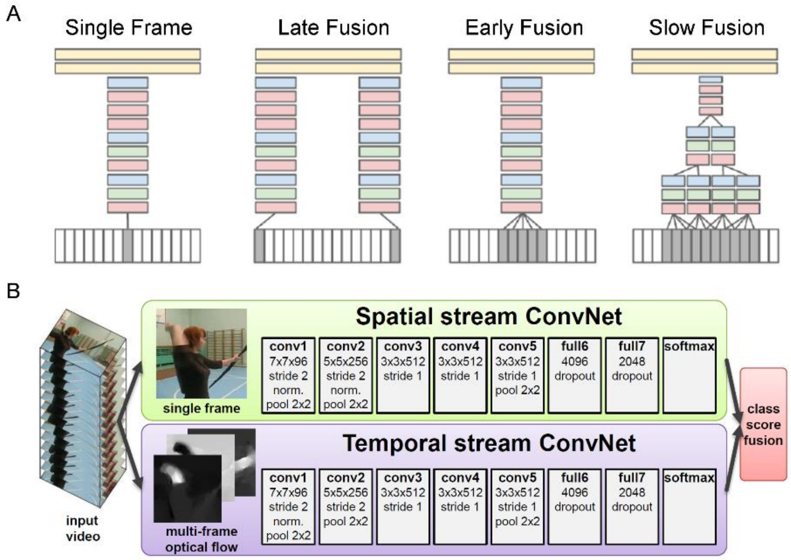


Figure A 4.11 Different techniques for spatiotemporal learning. A. Explored approaches for fusing information over temporal dimension through the network. Red, green and blue boxes indicate convolutional, normalization and polling layers, respectively. In the Slow Fusion model, the depicted columns share parameters. **B.** Two-stream architecture for video classification. Images adapted with permission from: A. Karpathy et al. (2014), and B. Simonyan and Zisserman (2014).

Temporal sequence modeling techniques, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs), are one of the most used for the temporal analysis of sequential data. Traditional RNNs (Figure A 4.12A.) are able to learn complex temporal dynamics by using recurrent connections (Equation (6.39) and (6.40)) in the hidden layers, between the input, hidden states, and output sequences:

$$h_t = g(W_{x\,h}x_t + W_{hh}h_{t-1} + b_h) \tag{6.39}$$

$$\hat{y}_t = g(W_{yh}\,h_t + b_y) \tag{6.40}$$

where g is an element-wise non-linear activation function, x_t is the input sequence, $h_t \in \mathbb{R}^N$ is the hidden state with N hidden units, and $\hat{y}_t$ is the output at time t.

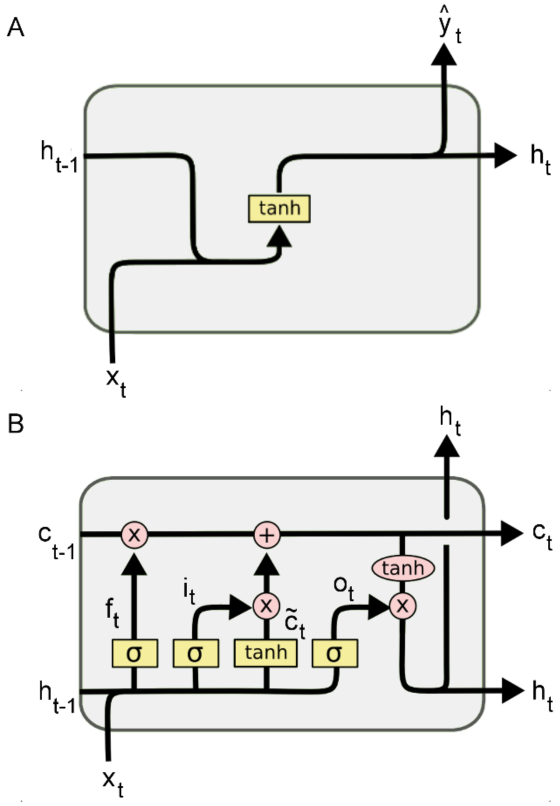


Figure A 4.12 Temporal sequence modeling techniques. Diagram of a basic: **A.** RNN cell, and **B.** LSTM memory cell.

A disadvantage when using RNNs units is the difficulty in training them to learn long-term dynamics. In fact, RNNs networks are more prone to suffer from vanishing or exploding gradients problems, that can result from difficulties in propagating the gradients back through the many layers of the recurrent network, each corresponding to a particular timestep. This is why this short-term memory is not enough to handle some types of sequential data, such as video sequences. LSTMs were proposed by Hochreiter and Schmidhuber Hochreiter and Schmidhuber (1997) as a solution for long time dependencies (Figure A 4.12B). This specific recurrent architecture incorporates memory cells to store foregone information (learning when to forget) and gates to control the updating of the memory (learning when to update hidden states with new information), controlled as follows (Hochreiter & Schmidhuber, 1997):

$$i_t = \sigma(W_{x\,i} x_t + W_{hi} h_{t-1} + b_i) \tag{6.41}$$

$$f_t = \sigma(W_{x\,f} x_t + W_{hf}\, h_{t-1} + b_f) \tag{6.42}$$

$$o_t = \sigma(W_{x\,o}\, x_t + W_{ho}\, h_{t-1} + b_o) \tag{6.43}$$

$$\tilde{c}_t = \phi(W_{x\,c} x_t + W_{hc}\, h_{t-1} + b_c) \tag{6.44}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{6.45}$$

$$h_t = o_t \odot \phi(c_t) \tag{6.46}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid activation function, $\phi(x) = 2\,\sigma(2x) - 1$ is the hyperbolic tangent activation function, $i_t \in \mathbb{R}^N$ is the input gate, $f_t \in \mathbb{R}^N$ is the forget gate, $o_t \in \mathbb{R}^N$ is the output gate, $\tilde{c}_t \in \mathbb{R}^N$ is the input modulation gate, $c_t \in \mathbb{R}^N$ is the memory cell, and '⊙' is the element-wise product of vectors.

These additional cells enable LSTMs to learn very complex and long-term temporal dynamics, overcoming the problem of vanishing/exploding gradients of the classical RNNs. However, traditional full-connected LSTMs, by taking the vectorized features as inputs, loses the spatial correlation information during recurrence operations. Also, and similarly to MLPs when applied to 3D feature maps, LSTMs are time- and memory-consuming due to the large matrix sizes, and translational variant by feature vectorization. Altogether, these limitations shorten its use in image processing applications. Convolutional LSTMs (ConvLSTMs) were proposed by X. Shi et al. (2015) to naturally handle 3D convolutional features as inputs and preserve spatial information, equivalent to the CNNs in feedforward NN. The ConvLSTM cell replaces matrix multiplications in every gate of the traditional LSTM cell (Equations. (6.47)

– (6.52)) by convolution operations. Formally, the activations of a ConvLSTM cell at time t are formulated as follows (X. Shi et al., 2015):

$$i_t = \sigma(W_{x\,i} * x_t + W_{hi} * h_{t-1} + b_i) \tag{6.47}$$

$$f_t = \sigma(W_{x\,f} * x_t + W_{hf} * h_{t-1} + b_f) \tag{6.48}$$

$$o_t = \sigma(W_{x\,o} * x_t + W_{ho} * h_{t-1} + b_o) \tag{6.49}$$

$$\tilde{c}_t = \phi(W_{x\,c} * x_t + W_{hc} * h_{t-1} + b_c) \tag{6.50}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{6.51}$$

$$h_t = o_t \odot \phi(c_t) \tag{6.52}$$

where '$*$' and '$\odot$' denote the convolution and *Hadamard* product operations, respectively, and $W_{x\sim}$ and $W_{h\sim}$ are 2D convolution kernels. The convolutions, together with recurrence operations, can take full use of the spatiotemporal correlation information, empowering traditional LSTMs to learn long-term spatiotemporal features.

When compared to these RNN-based networks in modeling the video sequences' dynamics, traditional feedforward NN are not able to accomplish the task since they do not share feature information across different positions of the network; they simply assume that all inputs and outputs are independent of each other. Therefore, RNN-based models have proven successful on tasks such as image and video description (Donahue et al., 2015), machine translation (Cho et al., 2014), and activity/behavior recognition (Donahue et al., 2015; Kramida et al., 2016; Majd & Safabakhsh, 2020; Murari, 2019). In the particular case of segmentation, RNNs can be used for two main purposes (Figure A 4.13): capture temporal information of a video sequence (Pfeuffer et al., 2019; Qiu, Yao, & Mei, 2017; Rebol & Knöbelreiter, 2020; Salvador et al., 2017; Zou et al., 2019), or learn the global context of an image (in which the image is divided into small regions or superpixels, and each of these is sequentially fed into an RNN-based network to learn the spatial relationship between them) (Byeon, Breuel, Raue, & Liwicki, 2015; Liang, Shen, Feng, Lin, & Yan, 2016). In addition to applying RNN-based models to predict frame-to-frame semantic segmentation, it is also possible to improve the consistency of the results by modifying, for example, the loss function. Rebol and Knöbelreiter (2020) proposed to extend the cross-entropy loss function with a novel inconsistency error term. This inconsistency loss penalizes pixels with different predictions in consecutive frames, which were already predicted correctly in at least one frame of the consecutive set.

Finally, a natural extension of 2D CNNs to video is CNNs with 3D spatiotemporal convolutions and pooling layers, to extract features from both spatial and temporal dimensions. 3D convolutions are achieved by convolving 3D kernels to the cube formed by stacking multiple contiguous frames together, as an extension of the spatial dimension along with the time domain. As a result, multiple contiguous frames are included in the obtained feature maps, which will, in turn, be connected to feature maps of the previous layers to capture motion information across frames. Therefore, unlike 2D convolutions that only learn spatial information, whether they are applied to an individual frame (Figure A 4.14A) or multiple frames sequentially (Figure A 4.14B), 3D convolutions allow preserving temporal information, returning a 3D output feature map (Figure A 4.14C).

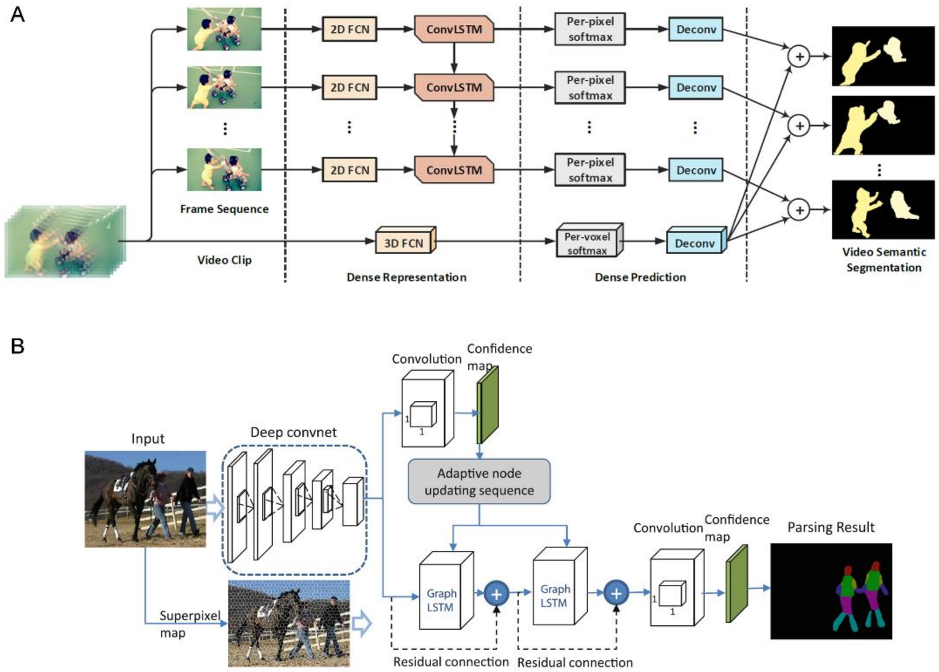


Figure A 4.13 Deep architectures for semantic segmentation tasks. A. DST-FCN architecture for semantic video segmentation. This framework could be divided into two streams, treating the input video clip as sequential (individual) frames (2D FCN + ConvLSTM for long-term temporal relationships) and a whole clip separately (3D FCN for dense representation for the entire clip). **B.** Graph LSTM layers combined with FCN for semantic object parsing. Images adapted with permission from: A. Qiu et al. (2017), and B. Liang et al. (2016).

3D convolutional networks (3D-CNNs) were initially proposed by Baccouche, Mamalet, Wolf, Garcia, and Baskurt (2011) and Ji et al. (2012) for human action recognition, and extended by Tran, Bourdev, Fergus, Torresani, and Paluri (2015) to include 3D pooling layers in a convolutional 3D network (C3D), or by Varol, Laptev, and Schmid (2017) by feeding 3D CNN with longer continuous RGB frames sequences in Long-Term Temporal Convolutions.

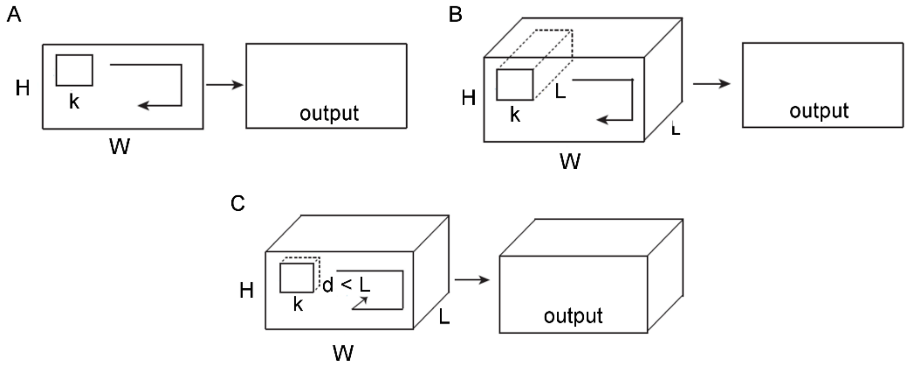


Figure A 4.14 2D and 3D convolution operations. A. Applying 2D convolution on an image results in an image. **B.** Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. **C.** Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal. Adapted with permission from Tran et al. (2015).

Several methods that take advantage of 3D convolutions have been proposed for learning spatiotemporal features, and generally outperform 2D counterparts (Hara, Kataoka, & Satoh, 2018; K. Liu, Liu, Gan, Tan, & Ma, 2018; Mahadevan et al., 2020). However, this comes at the cost of estimating a larger number of parameters for 3D kernels, which increases the risk of overfitting. Also, because of the number of parameters to learn, training 3D networks is more challenging, especially when dealing with long-duration sequences. Consequently, new methods have emerged that propose a combination of 3D CNNs and RNN-based networks (Figure A 4.15), arguing that the latter are more suitable to encode long-term temporal information, especially from various-length videos, and in turn, 3D CNNs are superior in learning short-term temporal features between adjacent video frames (Molchanov et al., 2016; X. Wang, Miao, Zhang, & Hao, 2019; L. Zhang et al., 2017).

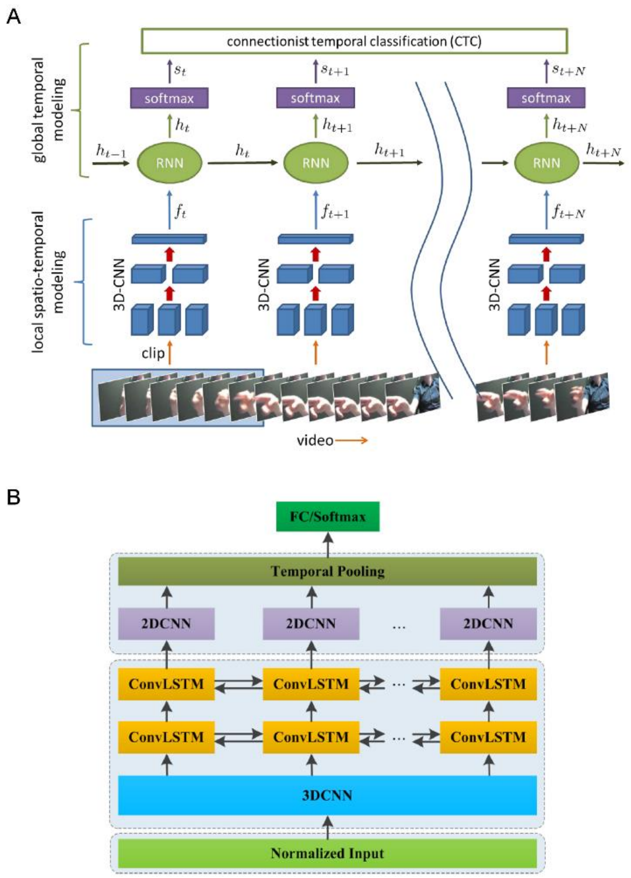


Figure A 4.15 Combination of 3D convolutional layers and RNN/LSTM cells in CCN-based architectures. A. *R3DCNN* architecture for the classification of dynamic gestures. A gesture video is presented in the form of short clips to a 3D-CNN for extracting local spatial-temporal features. These features are input to a recurrent network, which aggregates transitions across several clips. **B.** *3DCNN* and bidirectional ConvLSTM are utilized to learn the short-term and long-term spatiotemporal features, successively, and then *2DCNN* is used to learn higher-level spatiotemporal features based on the learnt 2D long-term spatiotemporal feature maps for the final gesture recognition. Images adapted with permission from: A. Molchanov et al. (2016), and B. L. Zhang et al. (2017).

4.3.1 Experiments

To take advantage of the temporal information present in the videos of the original RGB-D dataset, and to confirm that this information can improve the performance of the previously proposed segmentation approach, the U-Net architecture was extended by using different methods that consider image information from previous or contiguous frames. In fact, some segmentation errors occurring only at single frames, such as flickering pixels, borders, and animal shadows, could be avoided using additional information from previous or consecutive frames, instead of processing each image independently. Hence, temporal information is used in these experiments to improve segmentation results.

Dataset

The annotated dataset, composed of a total of 600 frames, was split into 400 images for training, 100 images for validation, and 100 images for testing, being all consecutive frames within each independent set. Initially, these independent sets were again subdivided into sequences of 16 consecutive frames (clips), following Tran et al. (2015) approach. The time difference τ between two frames of the sequence contains approximately 67 ms.

Implementation Details

Different approaches for the video segmentation task were implemented, using the U-Net model architecture, previously described (Figure A 4.3), as the backbone architecture. Here, all max-pooling operations were replaced by strided convolutions (stride equal to 2) (Figure A 4.16A). The motivation for this modification is that by replacing by a strided convolution, the pooling operation can also be learned, which may increase model's expressiveness ability and improve the overall accuracy of a model with the same depth and width (Springenberg, Dosovitskiy, Brox, & Riedmiller, 2014). Also, in feature upsampling at the decoder, transpose convolutions with stride equal to 2 were applied.

One technique to take into account last frames' information for feature learning is through RNN, such as LSTMs that can be easily trained and integrated. Thereby, the traditional U-Net architecture was extended by placing a ConvLSTM layer at different positions in the network, in order to find which position is most suitable for learning depth images in segmentation tasks. The extended U-Net is called hereinafter U-Net-ConvLSTM. Following Pfeuffer et al. (2019) methodology, 3 different versions of the U-Net-ConvLSTM architecture were tested, where the temporal integration is performed differently (Figure A 4.16B). In version 1, a ConvLSTM layer is placed between the encoder and the decoder. Since the encoder determines global image information, which should not vary much between neighboring frames, memorizing and modifying these image features might avoid flickering of features and improve segmentation results. In turn, version 2 consists of placing a ConvLSTM layer at the end of the network. Here, each frame is processed independently to avoid error propagation through the network. In the end, the results from neighboring frames are combined using the recurrent structure, resulting in temporal filtering of the final segmentation map. A combination of these two versions was also considered to take advantage of both global and filtering advantages (version 3). The kernel size of the ConvLSTM layers in all versions was set to 3×3 and the number of output channels is equal to the one of the previous layer.

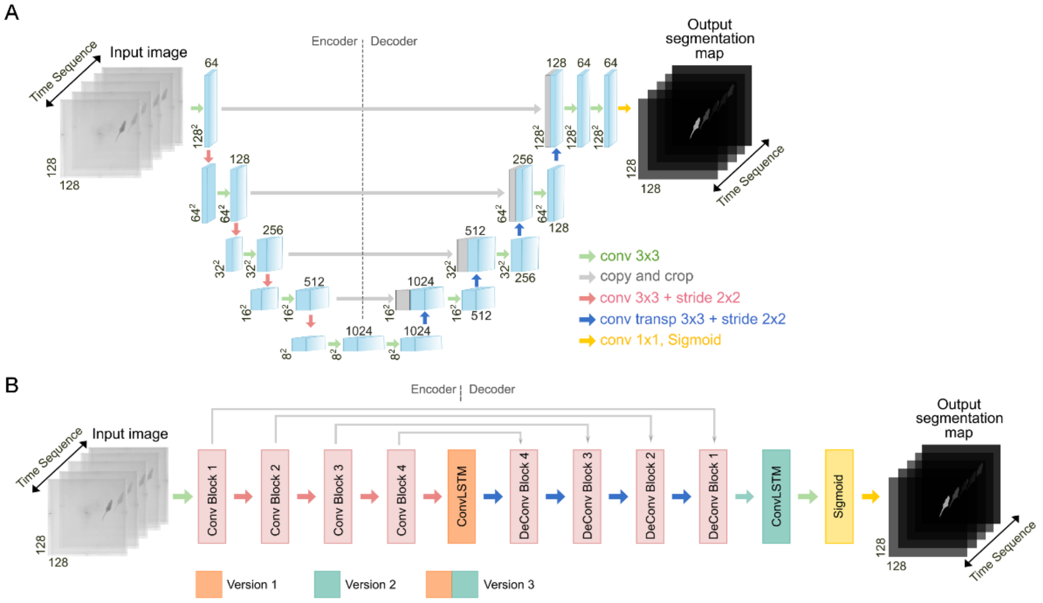


Figure A 4.16 U-Net architecture for a depth video sequence of the original RGB-D dataset (input size of 128x128x16). A. U-Net backbone architecture for an input video sequence. Each blue box corresponds to a multi-channel feature map, in which the number of channels is denoted on top of the box and the corresponding x-y-size at the lower-left edge of the box. The network is divided into two convolutional parts: encoder, on the left, and the decoder, on the right. Grey boxes in the decoder section represent copied feature maps. The arrows denote the different operations, subtitled in the image itself. Inspired by Ronneberger et al. (2015). **B.** U-Net-ConvLSTM network architecture: light pink boxes illustrate original U-Net layers, while the different positions of the ConvLSTM layers are colored, according to the version. Adapted with permission from Pfeuffer et al. (2019).

3D convolutions were also tested as a natural extension of 2D convolutions for spatiotemporal feature learning. In this way, the convolutional and max-pooling layers of the original U-Net architecture were extended in 3D, to comprise 3D input dimensions (U-Net 3D). Following (Tran et al., 2015), 3×3×3 space-time filters were used for all convolutional layers, instead of the original 3×3 kernels. All of these convolutional layers were applied with appropriate padding (both spatial and temporal), and max pooling layers with kernel size 2×2×2, except for the first and last layer with kernel size 1×2×2 to not merge the temporal signal too early and avoid collapsing the temporal signal (Tran et al., 2015).

All networks were trained from scratch, with SGD implementation of *TensorFlow* and ADAM optimizer and Dice BCE loss function. The training consisted of a maximum of 100 iterations with a default batch size of 1, an initial learning rate 0.0001, and a momentum of 0.9. The resolution of the input images was reduced to 128×128 due to memory and time reasons. These training conditions were set for all different networks so that only the architectural differences influence the result. To determine which parameters should be used in training such networks, ablation studies were initially carried out, to find the optimal batch size (N = 1 or 4), dropout (equal to 0.5), activation function (ReLU or Leaky ReLU), and architecture size (with 1 convolution per block – small size, or with 2 convolutions per block – big size).

The effect of increased temporal information was also evaluated, by increasing the size of each clip to $\{32, 48, 64, 80\}$ frames. Also, the time difference τ between each frame within one clip was increased to approximately $\{133, 200, 267\}$ ms, corresponding to sampling every two, three, or four frames of the original dataset, respectively. These experiments will provide some insights into whether network learning is improved by changing the temporal extent (number of frames per clip) and/or the granularity of temporal information (time step between two consecutive frames in a clip). The same augmentation operations were applied to all frames in each clip, with the previously described parameters (Appendix - Semantic Segmentation with U-Net – segmentation of animal's whole-body - Experiments). When using 16 frames per clip with a time difference of 67 ms, a total of 625 clips (10000 frames) were produced for training and a total of 6 clips (96 frames) for each validation and testing.

4.3.2 Results

To understand if the integration of recurrent layers can help in the animal's body segmentation, by taking advantage of temporal information between frames, the performance of networks with different architectures was studied. Initially, and using the default parameters, the effect of introducing ConvLSTM layers was explored, and the results showed that whatever the position of these layers, the performance is always improved when compared to the simple U-Net architecture (Supplementary Figure S 5.1A). It's interesting to point out that, during inference (Supplementary Figure S 5.1B), the presence of ConvLSTM layers improves the segmentation mask as the sequence progresses (the first few frames have poor segmentation masks, but the quality of segmentation improves over time). This is expected for recurrent layers since over time there is more background knowledge available to help segment the frames.

Overall, U-Net-ConvLSTM versions 1 and 3 achieved superior results, proving that, for this segmentation task, introducing a ConvLSTM layer between the encoder and decoder has a stronger effect. Here, since global information is determined by the encoder and that this global information does not vary between consecutive frames (static camera, background features similar across time, animal shape consistent between contiguous frames), segmentation is improved over time using stored global features. Using a ConvLSTM layer at the end of the network allows further improvement of the segmentation, and for that reason, U-Net-ConvLSTM version 3 is the best architecture for the segmentation task, with more consistent results across trials.

Networks' architectures were fine-tuned by testing different parameters (Supplementary Figure S 5.1). When the number of convolutional layers per block is increased, there was an overall increase in the performance of the segmentation (cleaner background and more detailed animal shape). A decrease in overfitting was observed, as expected, with the introduction of dropout layers at the end of the encoder, with a slight increase in model's performance. Finally, changing the activation function to the Leaky ReLU one did not bring any improvements. Results showed weaker segmentation maps, lower overall performance in training and validation sets, and the model was slower to converge.

To investigate the behavior of U-Net-based networks for increasing temporal information, U-Net, U-Net-ConvLSTM version 3 and U-Net 3D networks were systematically compared (Figure A 4.17). The results showed that increasing the time difference between sampling frames impacts the performance of all U-Net-based networks with smaller size (1 conv./block), with a slight performance increase for τ equals to 2 and 3 (which corresponds to sampling every two or three frames in the original dataset). For higher τ values, the segmentation performance starts to decrease (Figure A 4.17).

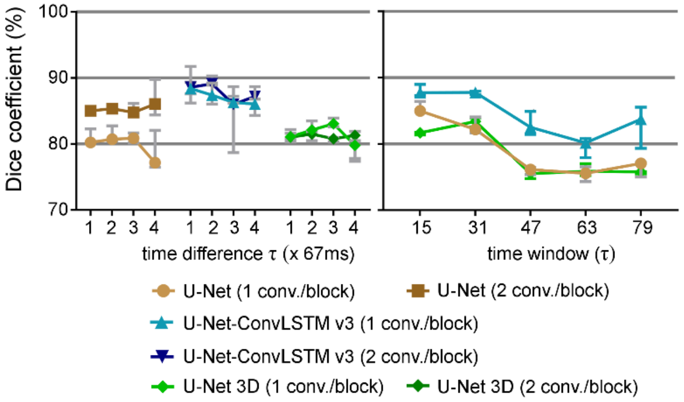


Figure A 4.17 Semantic segmentation results of U-Net-based networks for increased temporal information. Different architectures were compared: U-Net, U-Net-ConvLSTM version 3 (v3) and U-Net 3D, with 1 or 2 convolutional layers in each block (conv./block). Left: Overall Dice coefficient for increasing time differences τ (in miliseconds, ms) between sampling frames in the input sequence. Right: Overall Dice coefficient for increasing temporal extents. Time window in units of τ (τ = 67 ms). Data represented as median ± 95% confidence interval (N = 3 trials).

This can be explained by the granularity of depth information present in contiguous frames: for low time differences, the information contained in consecutive frames is similar and does not suffer from abrupt changes. However, with high time differences, the variations in depth information are greater, with frames more distinct from each other and containing completely different animal movements. For this reason, consecutive frames sampled with a higher time difference may no longer contain relevant information for segmenting the current frame. On the other hand, for time-windows longer than 2 seconds, approximately, the segmentation performance decreases drastically. This is aligned with previous results, where the segmentation learning does not seem to benefit from very distant temporal information, degrading networks' performance. Finally, temporal integration through 3D convolutions doesn't seem to improve segmentation, with similar performances to the U-Net model. In fact, since ConvLSTM layers are designed to process spatial and temporal information separately, they may be more suitable when the temporal component is truly important, avoiding mixing information of different scales.

Overall, these results showed that temporal information appears to be crucial for the learning process and that networks must be carefully designed to allow spatiotemporal integration on a time scale that fits the question under study.